User Manual

Tektronix

TekTMS RunTime Generator
S3FT110

070-8322-01

This document supports software version 2.5

Please check for change information at the rear of this manual.

First Printing: October 1992

Tektronix, Inc., P.O. Box 500, Beaverton, OR 97077

Printed in U.S.A.

SOFTWARE WARRANTY SUMMARY

Tektronix warrants that its software products will conform to the specifications in the documentation provided with the product, when used properly in the specified operating environment, for a period of three (3) months. The warranty period begins on the date of shipment, except that if the program is installed by Tektronix, the warranty period begins on the date of installation or one month after the date of shipment, whichever is earlier. If the software product does not conform as warranted, Tektronix will provide the remedial services as described in the documentation provided with the product.

For products offered without documentation, Tektronix warrants that the media on which the software product is furnished and the encoding of the programs on the media will be free from defects in materials and workmanship for a period of three (3) months from the date of shipment. If any such medium or encoding proves defective during the warranty period, Tektronix will provide a replacement in exchange for the defective medium. Except as to the media on which the software product is furnished, the software product is provided "as is" without warranty of any kind, either express or implied.

Tektronix does not warrant that the functions contained in any software product will meet Customer's requirements or that the operation of the programs will be uninterrupted or error-free.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and, for warranted products, make suitable arrangements for such service in accordance with the instructions received from Tektronix. If Tektronix is unable, within a reasonable time after receipt of such notice, to provide remedial service for warranted products or, for "as is" products, to provide a replacement that is free from defects in materials and workmanship, Customer may terminate the license for the software product and return the software product and any associated materials for credit or refund.

The above warranties shall not apply to any software product that has been modified or altered by Customer. Tektronix shall not be obligated to furnish service under this warranty with respect to any software product a) that is used in an operating environment other than that specified or in a manner inconsistent with the User Manual and documentation; or b) when the software product has been integrated with other software if the result of such integration increases the time or difficulty of analyzing or servicing the software product or the problems ascribed in the software product.

THE ABOVE WARRANTIES ARE GIVEN BY TEKTRONIX WITH RESPECT TO THE LISTED PRODUCTS IN LIEU OF ANY OTHER WARRANTIES, EXPRESS OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO PROVIDE REMEDIAL SERVICE WHEN SPECIFIED, REPLACE DEFECTIVE MEDIA, OR REFUND CUSTOMER'S PAYMENT, AS APPLICABLE, IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO CUSTOMER FOR BREACH OF EITHER WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Preface

About this Manual

This manual describes how to use the Tektronix Test Management System RunTime Generator (TekTMS/RTG). The manual sections are as follows:

- Section 1, General Information. This section describes TekTMS/RTG, its package contents, and the required hardware and software.
- Section 2, Getting Started. This section tells you how to install and run TekTMS/RTG. It also contains exercises to help you become familiar with TekTMS/RTG. For more detailed information about disk files and other start up information see Appendix B. For more information about using the translators and their menus and dialog boxes, see Sections 4 and 5, and Appendix A.
- Section 3, The Translation Process. This section describes the TekTMS/RTG translation process.
- Section *4* Using the Test Procedure Translator *(RTG.EXE)*. This section contains a guide to the various tasks you can perform with RTG.EXE and the instructions for performing those tasks.
- Section 5, Using the Instrument Driver Translator *(IDG.EXE)*. This section contains a guide to the various tasks you can perform with IDG.EXE and the instructions for performing those tasks.
- Section 6, Running a DOS Executable File. This section describes sample displays that occur when running a DOS-executable file. It also describes, and shows how to edit, the TEKRUN.INI file installed by TekTMS/RTG to initialize the DOS execution environment.
- Appendix A, Menus. This appendix describes the Application window and each menu and dialog box for the TekTMS/RTG translators (RTG.EXE and IDG.EXE).
- Appendix *B,* Disk Files and Miscellaneous Operating Information. This appendix contains information about installed files, file extensions, file names, and file name conflicts. It also shows you how to start the TekTMS/RTG translators (RTG.EXE and IDG.EXE) from the DOS prompt.
- Appendix C, Software Performance Report. This appendix provides a form for submitting problems encountered while using TekTMS/RTG.

Typographic and Notational Conventions

This manual uses the following typographic and notational conventions.

CAPITAL LETTERS — Capital letters denote names of files (e.g., TEKTMS.EXE), environmental variables (e.g., SET), programs (e.g., LINK), macros (e.g., MAX)), and names of keyboard keys and key sequences (such as RETURN and CTRL+C). As programs are case insensitive, these names do not need to be capitalized when used in programs or as workstation entries.

CTRL+C denotes a key sequence where you hold down the first key (CTRL), then press the second key (C), then release both keys.

CTRL, C denotes a sequence where you press and release the first key (CTRL), then press and release the second key (C).

Bold Typeface — denotes on-screen items including keywords and menu commands.

***Italic* Typeface** — denotes place holders in syntax descriptions. These place holders must be replaced with specific terms or values by the user. For example, in the statement

```
GoTo name,
```

name must be replaced with a specific user supplied label name. Occasionally, an italic typeface emphasizes a word or phrase.

`Monospace Typeface` — denotes text in program listings, such as in the following listing:

```
For i=start To end Steps p Log
```

Terms and Definitions

.CFG File — A binary file that contains the names of files and the translator, compiler, and linker options selected during a TekTMS/RTG session. Whenever exiting a TekTMS/RTG session, this file must be saved, updated, or abandoned. When saved, TekTMS/RTG assigns a .CFG extension to the file. Once saved, the file can be executed by opening it and selecting a Translate push button. This file also may be called an RTG Configuration File or an IDG Configuration File.

.EXE File — This is a DOS executable file.

Item — A generic term describing various controls in a dialog box (e.g., list boxes, check boxes, push buttons, etc.).

IDG Configuration File — See .CFG file.

.ISD File — This is an Instrument Software Definition file (ISD). This file contains a software driver that provides a graphical interface for configuring and controlling programmable instrumentation. This ASCII text file also may be called an instrument driver file or a script file.

.LIB File — This is a file of predefined, precompiled functions. These functions may be Microsoft® C or TekTMS functions.

.PRG File — A test procedure file that can be read and interpreted by TekTMS/IPG. It contains a collection of test steps to perform a test process (e.g., setup instrument controls, make a measurement, etc.). This file also may be called a Test Procedure File.

Instrument Driver File — See .ISD File.

RTG Configuration File — See .CFG File.

Script File — See .ISD File.

TekTMS/IPG — This is an acronym for Tektronix Test Management System Interactive Procedure Generator.

TekTMS/RTG — This is an acronym for Tektronix Test Management System RunTime Generator.

Test Procedure File — See .PRG File.

Translation Process — This is any process executed by TekTMS/RTG based on its current file and option selections. The process may involve translation only, or translation, compilation, and linking, or any other combination.

TekTMS/RTG Documentation

The documentation provided with TekTMS/RTG is:

- *TekTMS RunTime* Generator User Manual
- TekTMS *RunTime* Generator Programmer Reference Manual

These manuals are contained within a single binder.

Other TekTMS Documentation

Other documentation provided with the TekTMS-family of products and useful with TekTMS/RTG is:

- TekTMS Interactive Program Generator User Manual
- TekTMS Front Panel Developer Manual
- S3FT200 TekTMS VXI Instrument Front Panel Library
- S3FT300 TekTMS GPIB Instrument Front Panel Library

Customer Service

For direct customer service of the TekTMS/RTG package:

- Report Software defects on the Software Performance Report included in Appendix B. Send the report to the address listed on the form.
- For other assistance, contact your local tektronix field office or your sales representative.

Contents

Section 1 General Information

Section 2 Getting Started

Section 3
The Translation Process

Section 4
Using the Test Procedure Translator

Section 5
Using the Instrument Driver Translator

Section 6
Running a DOS Executable File

Appendices

Section 1
General Information

General Information

What is TekTMS/RTG?

TekTMS/RTG is an acronym for the Tektronix Test Management System RunTime Generator. TekTMS/RTG is a Microsoft®Windows application for creating a DOS executable files (.EXE Files) from TekTMS/IPG files. TekTMS/RTG creates the DOS-executable by translating, compiling, and linking the interpreted test programs (.PRG Files), instrument drivers files (.ISD Files), and runtime routines generated by the TekTMS Interactive Procedure Generator (IPG). Figure 1-1 shows the process.

The TekTMS/RTG Package

The TekTMS/RTG package contains the following items:

- S3FT110 TekTMS RunTime Generator software on both 5.25" high density (1.2 MB) and 3.5" high density (1.44 MB) diskettes
- TekTMS *RunTime* Generator User Manual
- TekTMS *RunTime* Generator Programmer Manual

Both TekTMS/RTG manuals are contained in a single binder.

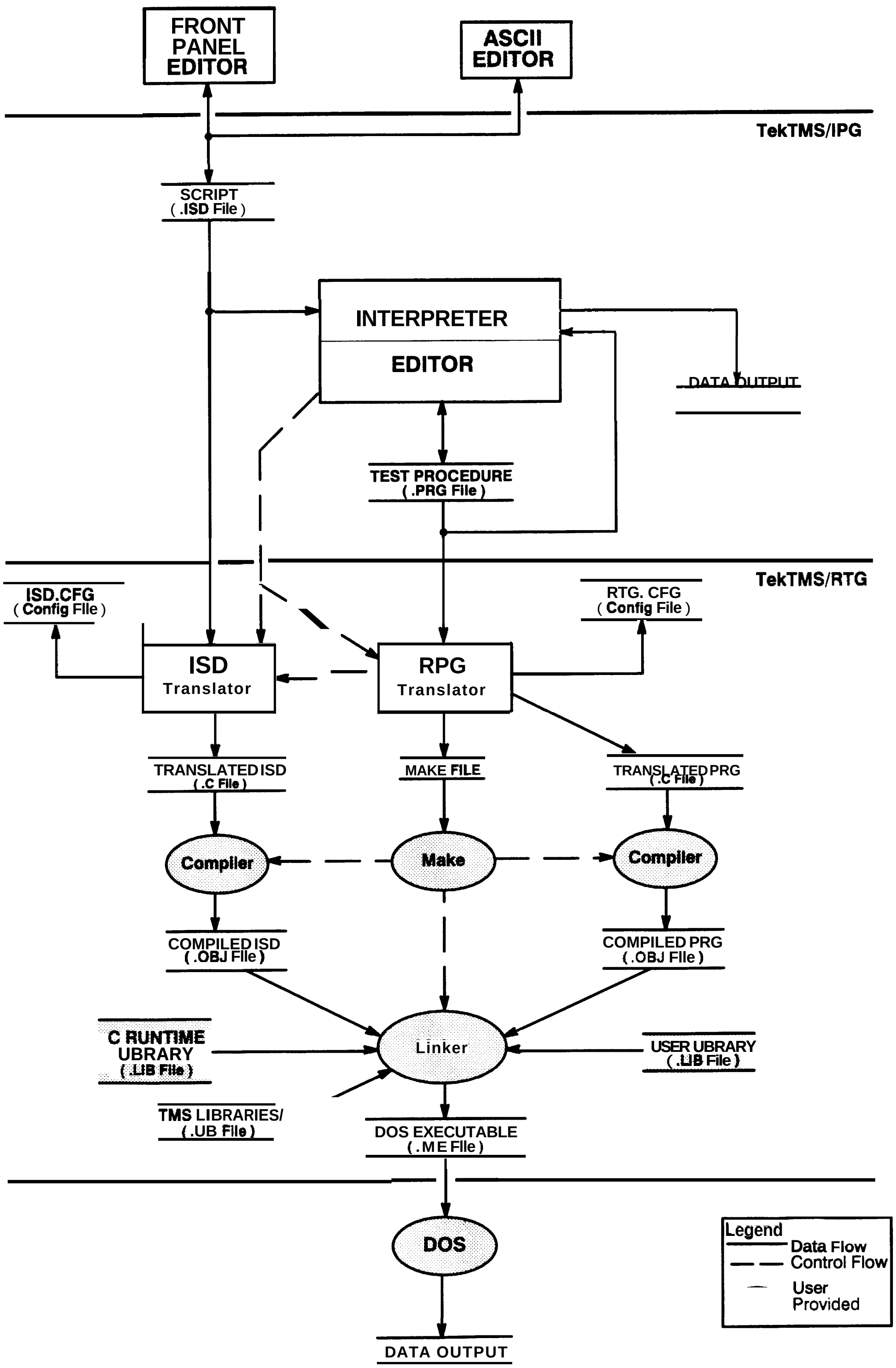


Figure 1-1: TekTMS/RTG Capability

User Supplied System Requirements

Before installing TekTMS/RTG (described in Section 2, Getting Started), check that your system meets the following minimum requirements. The additional recommended system components described here may accelerate the translation process.

Minimum Translate Time Hardware Environment

TekTMS/RTG runs in the same MS-Windows V3.1 or later environment used by TekTMS/IPG. Its requirements are a Personal Computer with:

- At least 640 Kbyte of RAM.
- A 40 Mbyte hard disk.
- A 3.5" high density (1.44 MB) or 5.25" high density (1.2 MB) diskette drive.
- A VGA, EGA, or CGA graphics adapter and display.
- Microsoft® Mouse, or equivalent for MS Windows V3.1 or later.
- Microsoft® Windows V3.1 or later.
- MS- or PC-DOS 3.2 or later as required by MS Windows V3.1 or later.
- Microsoft® C V6.0 or later.
- The National Instruments GPIB.COM instrument driver library for MS-DOS (only required for systems using GPIB compatible instruments).
- VXI Interface software (only required for VXI Systems).

Recommended Translate Time Hardware Environment

We recommend an 80286-, 80386-, or 80486-based computer such as:

8 A Tektronix VXI System Controller for VXI systems.

8 An IBM PC-AT compatible or IBM PC microchannel compatible.

We also recommend adding:

- A printer supported by Microsoft® Windows V3.1 or later.
- A math coprocessor.
- Additional RAM.

Minimum Runtime Hardware Environment

The runtime environment for the DOS-executable file created by TekTMS/RTG has fewer requirements than the translate time environment, but it includes instrument interface drivers. Its minimum requirements are:

- An 8086, 80286, 80386, or 80486 DOS-compatible PC with:
  - at least 512 Kbyte of RAM.
  - a hard disk, diskette, or network connection.
  - a VGA, EGA, or CGA graphics adapter and display.
  - a keyboard.
- Instrument drivers and their associated software for the type of instruments (VXI or GPIB) used in the test scenario.

  Tektronix VXI System Controllers include VXI and GPIB instrument drivers in the basic system. Other standalone computer systems may require one of the following GPIB bus controllers:
- A National Instruments PC-AT, PC-2A, PC-3, or microchannel card.

Recommended Runtime Hardware

We recommend an 80286-, 80386-, or 80486-based computer such as:

- a Tektronix VXI System Controller.
- an IBM PC-AT compatible with at least 640 Kbyte of RAM, one 1.2 Mbyte or 1.44 Mbyte diskette drive, and Hercules, EGA, VGA, or CGA graphics adapter, or equivalent, and compatible display.

We recommend adding:

- a Microsoft® Mouse, or equivalent supported by MS-DOS V3.2 or later.
- a math coprocessor.
- a printer supported by MS- or PC-DOS V3.2 or later.

Selecting and Configuring a System for TekTMS/RTG

Selecting a Computer System for Windows Enhanced Mode

TekTMS/RTG runs best on a computer running Microsoft® Windows in the 386 Enhanced Mode. The Enhanced Mode provides access to the virtual memory capability of the Intel 80386 processor, which allows Windows applications to use more memory than physically available. To run Windows in the 386 Enhanced Mode requires a computer with an Intel 80386 processor or higher, and at least 2 Mbyte of memory.

RAM Drive and Disk Caching

In general, any system hardware or configuration change that increases the performance of Windows also increases the performance of TekTMS/IPG and TekTMS/RTG. Windows provides a RAM Drive utility (RAMDRIVE.SYS) and a Disk Caching program (SMARTDRV.SYS) that speed up system performance. The RAM Drive utility uses part of system memory as a hard disk, which provides faster data access and program execution speed. The Disk Caching program works similarly by storing data in RAM that would normally always be accessed from the hard disk RAM. Any gain in access or execution speed depends on system configuration and RAM speed. For information about using the Windows RAM Drive utility and Disk Caching program, see Chapter 13, Optimizing Your System in the Microsoft Windows User Guide.

NOTE

> You can use disk caching programs other than SMARTDRKSYS, but they may not be totally compatible with Windows. Since SMARTDRVSYS is an integral part of the Windows package, it coordinates system and memory use. Using other disk caching programs may result in less effective memory management, slower performance, or memory management conflicts with Windows.

Improving Performance When Running Windows in Standard Mode

When running Windows in the Standard Mode, the key performance factor is the amount of available memory in the first 640 Kbyte of RAM. Available memory normally consists of base memory (640 Kbyte) minus the amount used by the operating system, device drivers, and any Terminate and Stay Resident (TSR) programs loaded. To determine the amount of available memory, use the DOS CHKDSK command.

NOTE

> Windows Real Mode is not supported in TekTMS RTG version 2.5. Real Mode is also not supported by Windows version 3.1 or later.

To increase the amount of available memory to speed up Windows when using TekTMS/IPG and TekTMS/RTG and Standard Mode, do the following:

CAUTION

Taking action on any of the following steps changes your *AUTOEXEC.BAT* or *CONFIG.SYS* files. Before making changes save a copy of the original files using an extension other than the original (the original extension may be *.BAT,* or some other user assigned extension).

1. Check your AUTOEXEC.BAT and CONFIG.SYS files for a line that loads a mouse driver, such as MOUSE.COM. Remove the line. Since Windows loads its own mouse driver, the extra driver only reduces available memory.
2. Check your AUTOEXEC.BAT file for any lines that load TSR programs, such as Sidekick®, and Sideways®. Remove the lines you don't need for running TekTMS/IPG or TekTMS/RTG.
3. Check your CONFIG.SYS file for a line that loads ANSI.SYS and remove it. If you remove this line, you also must remove any PROMPT statement in the AUTOEXEC.BAT or CONFIG.SYS file.

General Information About Installing Microsoft C and TekTMS/RTG

When the TekTMS/RTG translators compile and link the generated C code to produce an executable (.EXE) or library (.LIB),they use NMAKE.EXE, LINK.EXE, and C Runtime Library that are part of the Microsoft® compiler system.

The TekTMS/RTG translators need to know where to find the C Compiler (CL.EXE), Linker (LINK.EXE), Librarian (LIB.EXE), and Make utility (NMAKE.EXE); thus, the directory where they are installed should be added to the PATH statement in your AUTOEXEC.BAT file.

When you load C *7.0,* specify loading the graphics library (GRAPHICS.LIB) and the presentation graphics library (PGCHART.LIB). You should also select either the math coprocessor emulation or the math coprocessor options. If you have already loaded C, refer to the READ.I10 file on S3FT110 Disk 1 for instructions on how to merge these libraries without reloading C.

Section 2
Getting Started

Getting Started

This section describes how to install the TekTMS/RTG software and how to translate a script.

Installing TekTMS/RTG

TekTMS/RTG is quite easy to install with the supplied INSTALL Windows application. Before installing TekTMS/RTG review, the system requirements on page 1-3 and review the system configuration discussion on page 1-5. As you select from options during the INSTALL program, Help information will appear that describes the option. When prompted, any C include files should be installed into your existing C library, which is the default for INSTALL.

NOTE

For information about installed files, refer to the READ.*110* files on the product *disk(s).* For information about file extensions used by *TekTMS/RTG* and filename conflicts, see Appendix *B.*

To install the software from within the Windows Program Manager, perform the following:

1. lnsert product disk number 1 into an appropriate disk drive.
2. In the **File** menu, select **Run.**
3. In the dialog box, enter the drive designator, a colon (:), and the command INSTALL. If the product disk is in drive A you would enter

   ```
   A:INSTALL
   ```

4. The install program will prompt you for the installation location and give you status as the installation proceeds.

To install the software from the DOS prompt, perform the following:

1. Insert product disk number 1 into an appropriate disk drive.
2. At the DOS prompt, enter the drive designator, a colon (:), and the command INSTALL. If the product disk is in drive A you would enter

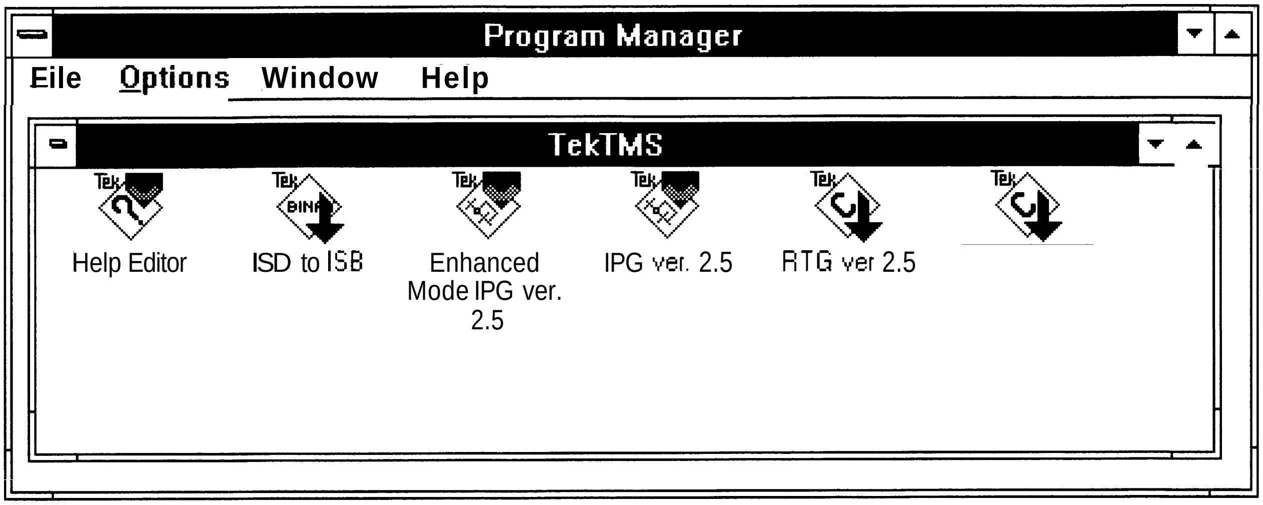
   ```
   A:INSTALL
   ```

   The INSTALL program will launch Windows and then proceed with the installation.

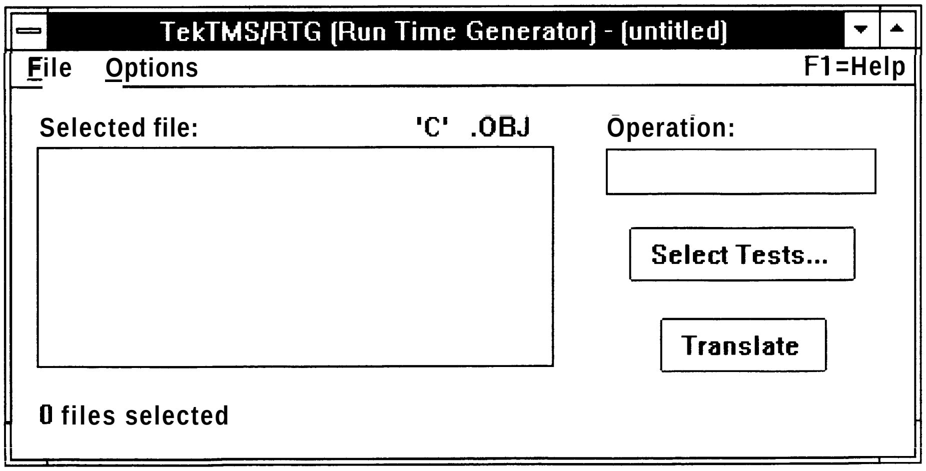
3. The INSTALL program will prompt you for the installation location and give you status as the installation proceeds.

Starting a Translator

If you installed TekTMS/RTG using the INSTALL.EXE program from one of the product disks the following TekTMS-family Application Group window appears when you start Windows. To start one of the TekTMS translators, double-click on the RTG or IDG icon.



When a translator Application window similar to the following appears, the translator is ready for use (the window shown is for the RTG translator).



NOTES

1. Your display may have background windows. These windows won't change the operation of the translator, but you may want to minimize them before continuing. To minimize windows, click on the down arrow in the upper right corner of the window. For more information, see the Windows User's Guide.

2. Because Windows is a multitasking environment, you can have more than one copy of a translator *(RTG.EXE* or *IDG.EXE)* running at the same time. If you should start a translation process on both copies at the same time, any one of the translate, compile, or link processes could fail on one or both copies. There is no advantage to running two copies of the translator at the same time. A disadvantage is the system runs slower because more tasks are being managed and swapped by the processor

Where to Develop Your Test System

When you are developing your test system, we suggest that all files for a particular test system be kept in a single directory. These files include:

*.PRG	The TekTMS/IPG interpretable version of your test program. These files should be developed and tested using the Interactive Procedure Generator before translating them into 'C' code.
*.ISD	The Instrument Software Definition files (instrument drivers) for the instruments used in the test system. You should load only those .ISD files required for your tests. We don't recommend loading the entire .ISD libraries as they are quite large. These files should be tested with the .PRG files using the Interactive Procedure Generator.
*.BMP	These are bit map files containing picture prompts for the test operator. These files must have been created by TekTMS IPG.
*.C	These are the translated C language files produced by the RTG.EXE and ISD.EXE translators. These file have the same names as the .PRG and .ISD files used to create them.
*.H	These are C language include files containing functional prototypes.
*.OBJ	These are object files produced by the Microsoft® C compiler.
*.EXE	This is an executable DOS version of your test system.

Quick Translation of a Main Test Procedure

TekTMS/RTG's makes translation of a main test procedure into a DOS-executable (.EXE) file straight forward. Simply select a main test procedure and click on the **Translate** pushbutton to start the translation process. The translator finishes the task of creating the DOS-executable file using default options.

Thus, to make a quick translation of a main test procedure into a DOS-executable file:

1. From the translator's Application window, click on the **Select Tests...** pushbutton.
2. When the Translate Programs dialog box appears, double-click on the Main test procedure file name in the files list.
3. Click on the **OK** pushbutton.
4. When the Application window appears, click on the **Translate** pushbutton. While the translator is running, file and translation status appear in the **Selected file:** and **Operation:** status boxes. When the compilation and linking process starts, an NMAKE window appears and displays compilation and linking status.
5. When the translation process ends, the screen and TekTMS/RTG window appears as follows:
   a. The **Selected file:** status box lists the name(s) of all .PRG files processed.
   b. Each file listed in the **Selected file:** status box contains an X in both the C and .OBJ columns showing that the files have been translated and compiled.
   c. The **Operation:** status box displays the word 'Complete'
   d. The NMAKE window that displays during compilation and linking is closed and removed from the screen.
   e. The words '.EXE File Current' appears in the lower right corner of the window.

NOTE

For information about the files created by the translation process, see Section 3, The Translation Process.

6. The file and option selections made during the translator session reside in a Configuration file. To save this Configuration file, see the Saving a Configuration File topic later in this section.

NOTE

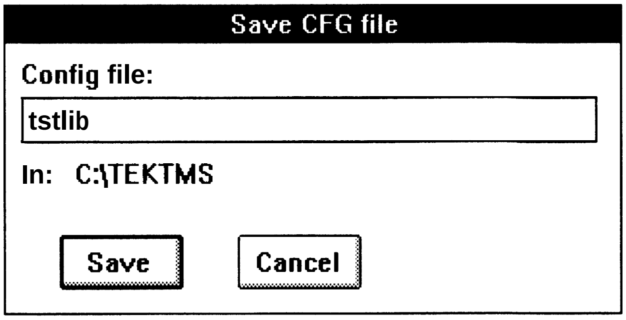
Familiarization Exercises 1 and 2 in the Translator Familiarization topic at the end of this section demonstrate this procedure.

Saving a Configuration File

During any translation session, each TekTMS/RTG translator maintains configuration information about files and options selected. This information can be saved as a Configuration file any time during the session or when exiting the session. The Configuration file is a time saver and convenience to the user when rebuilding or updating the DOS-executable files or libraries.

To save a Configuration file during a translator session:

1. From the translator's application window, select the **File** command.
2. When the File menu appears, select the **Save** command, which initiates one of the following actions:
   a. If the Configuration file is new and there is a main test procedure selected, the translator saves the file using the root name of the Main test procedure and a .CFG extension.
   b. If the Configuration file is new and you are building a library, the translator saves the file with the root name of the library and a .CFG extension.
   c. If the translator session has an opened Configuration file, that file is updated.
   d. If the Configuration file is new and there is no main test procedure or library name selected, the following dialog box appears for specifying a path name and naming the Configuration file. To name the Configuration file, enter a name **(tstlib** in this example) and click on **Save.**



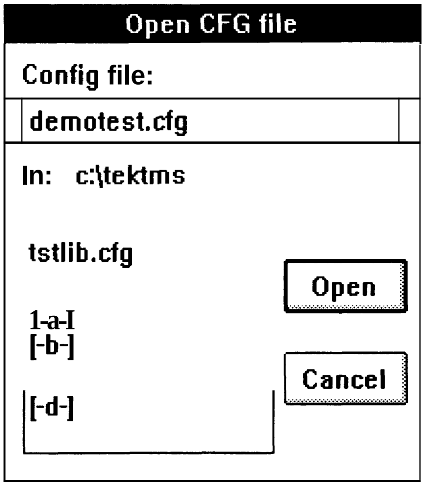
Config file: tstlib

In: C:\TEKTMS

Opening a Configuration File

To open a Configuration file:

1. From the translator's Application window, select the **File** command.
2. When the File menu appears, select the **Open** command.
3. When the following dialog box appears, double-click on the Configuration file name in the files list (**demotest.cfg** in this example) or select a file name and then select the **Open** command.



Open CFG file

Config file: demotest.cfg

In: c:\tektms

tstlib.cfg

1-a-I
[-b-]
[-d-]

Exiting a Translator Session

To exit a translator session:

1. Click on **Exit** from the File menu, or double-click on the Control-Menu box in the upper left corner of the translator's Application window.
2. If the session involves an opened Configuration file that hasn't changed, the session ends.

   If the session created a new or changed Configuration file, a Changed Configuration message appears giving the user a chance to save the Configuration file:

   a. Clicking on **No** exits the session without saving the file.

   b. Clicking on **Cancel** returns you to the Application window.

   c. Clicking on **Yes** invokes the SAVE function.

Translator Familiarization

To help familiarize you with the TekTMS/RTG translators and their menus and dialog boxes, you should complete the following exercises.

- Familiarization Exercise 1 uses a main test procedure named EXAMPLE1.PRG to produce the DOS-executable (.EXE) file. It follows the Quick Translation of a main test procedure method. This exercise doesn't require any test instrumentation or instrument drivers. The translation process uses default options.
- Familiarization Exercise 2 uses a main test procedure named EXAMPLE2.PRG and two GPIB instrument drivers named DM5010.ISD and PS5010G.ISD to produce the DOS-executable file. It follows the Quick Translation of a main test procedure method. This exercise requires instrument drivers during translation and actual instruments during execution. The translation process uses default options.
- Familiarization Exercise 3 uses a main test procedure named VXIDEMO.PRG and two VXI instrument drivers named VX5260.ISD and VX5790.ISD. In this example, the .ISD files produce an instrument driver library, then the .PRG file and the library are used to produce a DOS-executable file. This exercise requires instrument drivers during translation, and actual instruments during execution. The translation process uses default options.

NOTE

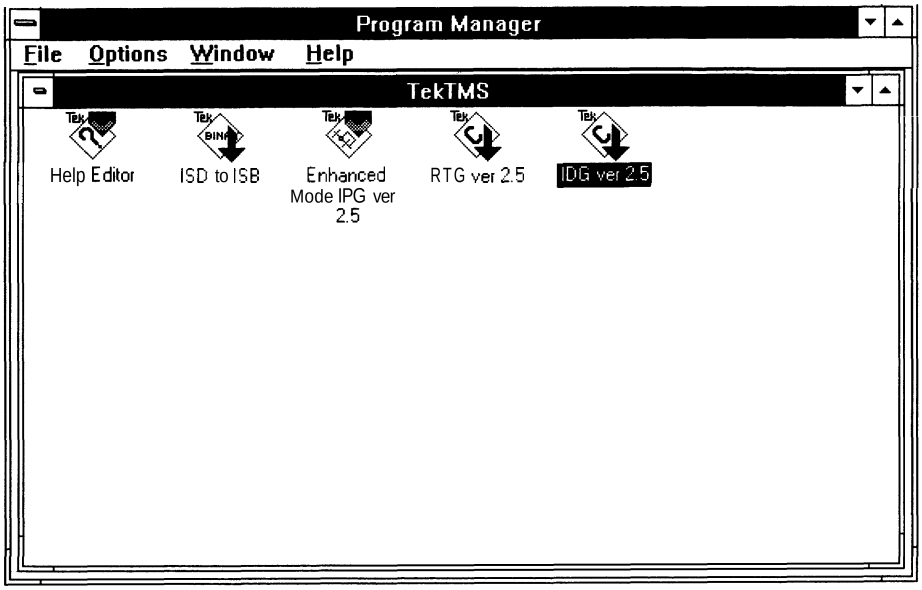
The .PRG test procedure files and the *.ISD* instrument driver files required for the exercises were loaded by the INSTALL program into the appropriate directories.

Familiarization Exercise 1

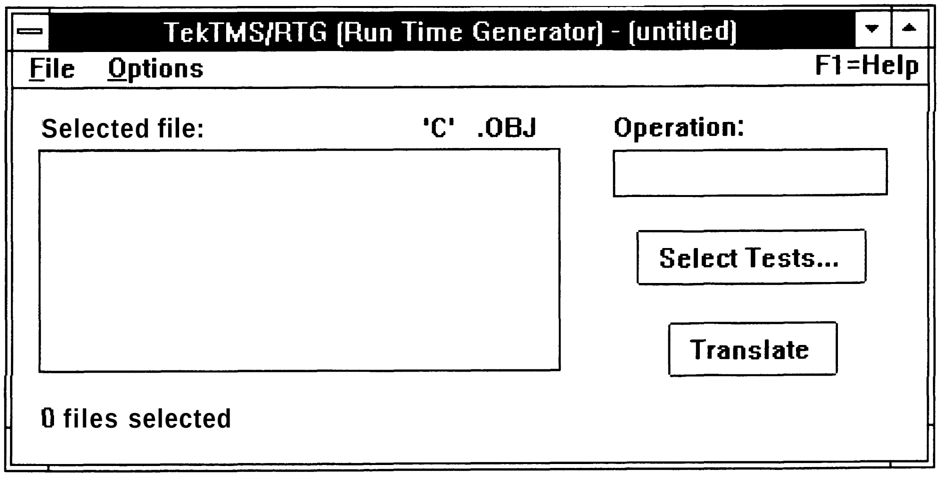
This exercise demonstrates the translation of a main test procedure without instrumentation using default options.

To translate the main test procedure for Exercise 1:

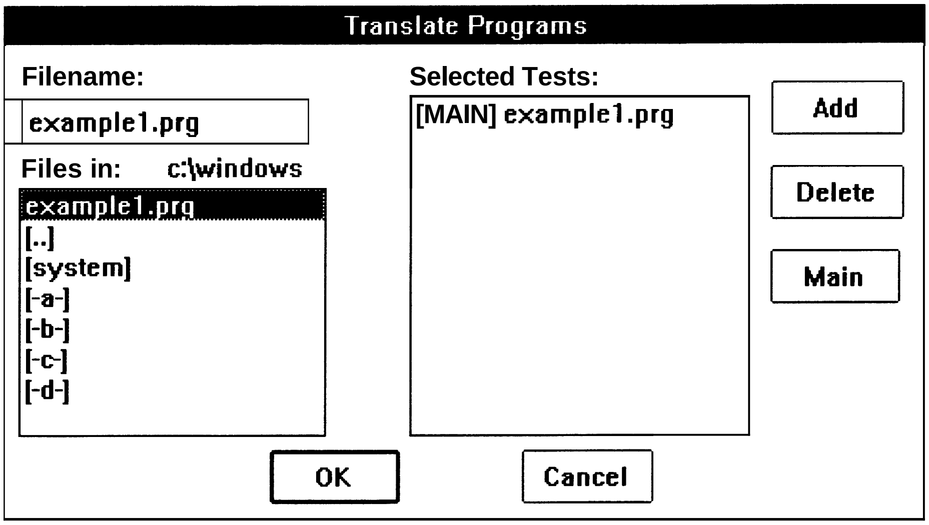
1. From the following TekTMS Application Group window, start the Test Procedure Translator (RTG.EXE) by double-clicking on the RTG icon.



2. When the following RTG translator Application window appears, click on **Select Tests....**



3. When the following Translate Programs dialog box appears, double-click on the EXAMPLE1.PRG file name in the files list, then click on **OK.**



Translate Programs

Filename: example1.prg

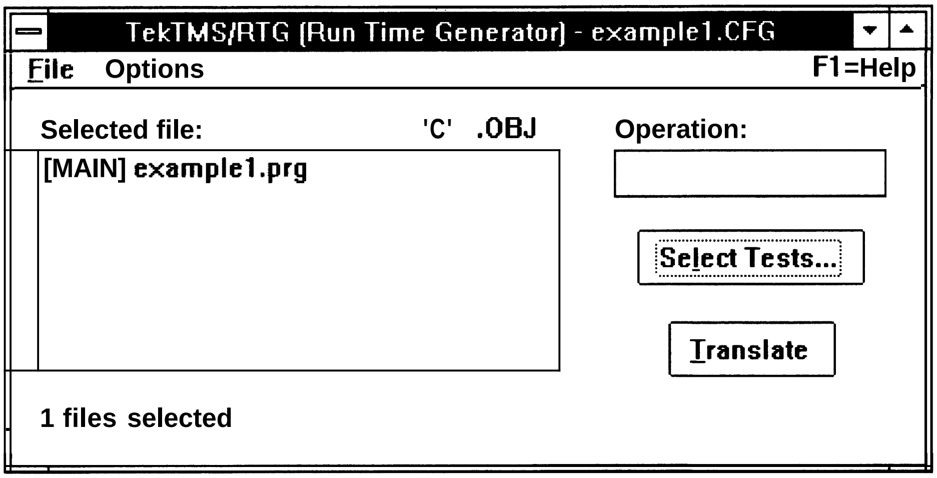
Files in: c:\windows

example1.prg
[..]
[system]
[-a-]
[-b-]
[-c-]
[-d-]

Selected Tests:

[MAIN] example1.prg

4. When the following Application window appears, click on **Translate.**



TekTMS/RTG (Run Time Generator) - example1.CFG

Selected file: 'C' .OBJ

[MAIN] example1.prg

Operation:

1 files selected

5. While the translator is running, file and translation status should appear in the **Selected files:** and **Operation:** status boxes. When the compilation and linking process starts, an NMAKE window should appear and display compilation and linking status.

6. When the translation process ends, the screen and RTG Application window should appear as follows:

    a. The **Selected files:** status box should list the EXAMPLE1.PRG file name with X's in the .C and .OBJ columns.

    b. The **Operation:** status box should display the word 'Complete'

c. The NMAKE window that displayed during compilation and linking should have been closed and removed from the screen.

d. The words '.EXE File Current' should appear in the lower right corner of the window.

7. Close the RTG.EXE translator window by clicking on **File**. When the File menu appears, click on **Exit.** When the Save Configuration File query appears, click on **No.**

Exercise 1 Files Created — The following files should have been created in the drive/directory where you loaded the EXAMPLE1.PRG file:

EXAMPLE1.C
EXAMPLE1.OBJ
EXAMPLE1.MAK
EXAMPLE1.LNK
EXAMPLE1.EXE

Printing Exercise 1 Files — You can print copies of the Example 1 files (except .OBJ and EXE) as follows:

- To print the .PRG file, start IPG from the TekTMS Application Group window by double-clicking on the IPG icon. When the IPG window appears, click on the **File** command. When the File menu appears, click on the Qpen command. When the file list appears, double-click on the EXAMPLE1.PRG file. When the test procedure appears, click on the **File** command. When the File menu appears, click on the Print command. When the print dialog box appears, click on the Full option, then on OK. When finished printing, exit the program.
- To print the .C, .MAK, and .LNK files, use the Print command from the Windows File menu.

NOTE

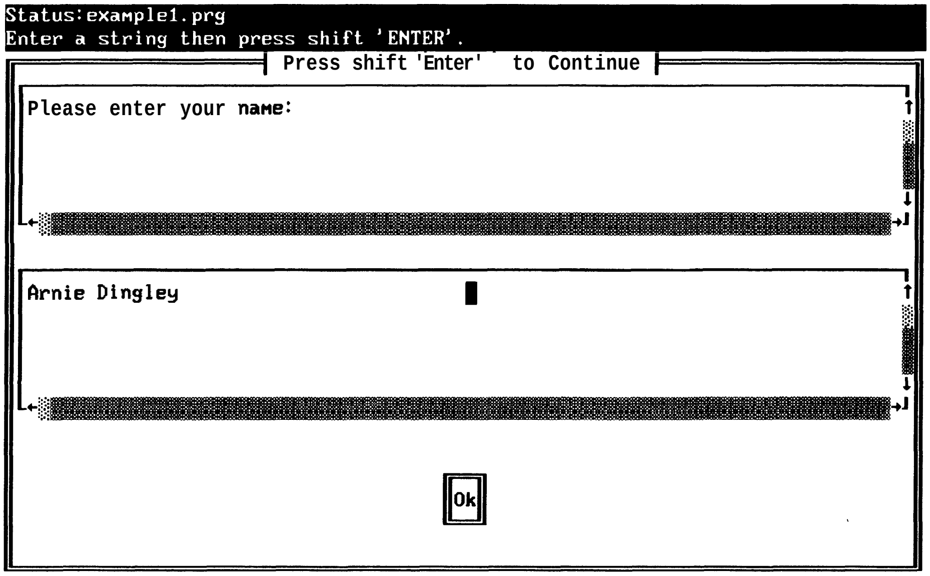
The *.OBJ* and .EXE files cannot be printed.

Running the Exercise 1 Executable (.EXE) File — To run the Exercise 1 executable file from the DOS prompt:

1. Type:

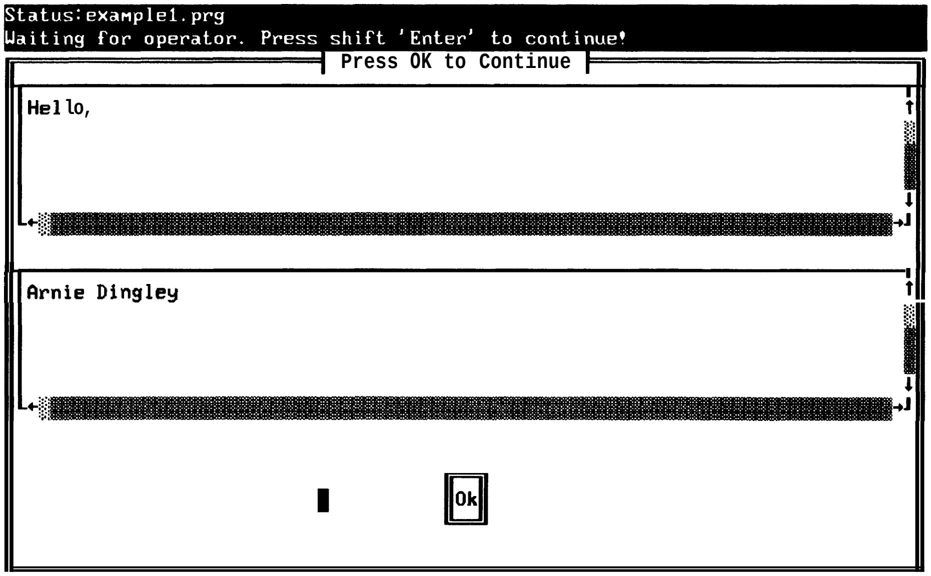
```
example1
```

and press RETURN. The following display should appear (except without the name in the bottom portion of the box).



```
Status:example1.prg
Enter a string then press shift 'ENTER'.
Press shift 'Enter'  to Continue

Please enter your name:

Arnie Dingley
```

2. Type in your name and click on **OK** or press SHIFT+RETURN. The following display should appear (except your name should appear in the lower portion of the box).



```
Status:example1.prg
Waiting for operator. Press shift 'Enter' to continue!
Press OK to Continue

Hello,

Arnie Dingley
```

3. Clicking on **OK** or pressing SHIFT+RETURN returns you to the DOS prompt.

Familiarization Exercise 2

This exercise demonstrates the most common use of TekTMS/RTG: The translation of a main test procedure with instrument drivers using default options.

To translate the main test procedure for Exercise 2:

1. From the TekTMS Application Group window, start the Test Procedure Translator (RTG.EXE) by double-clicking on the RTG icon.
2. When the RTG Translator Application window appears, click on Select Tests....
3. When the Translate Programs dialog box appears, double-click on the EXAMPLE2.PRG file name in the files list, then click on OK.
4. When the RTG Translator Application window appears, click on Translate.
5. While the translator is running, file and translation status should appear in the Selected files: and Operation: status boxes. When the compilation and linking process starts, an NMAKE window should appear and display compilation and linking status.

   The TekTMS IDG application will start automatically to translate the .ISD files used by Example2.PRG. When the translation of the .ISD file finishes, TekTMS IDG will close.
6. When the translation process ends, the screen and RTG Application window should appear as follows:
   a. The Selected files: status box should list the EXAMPLE2.PRG, file name with X's in the .C and .OBJ columns.

   b. The Operation: status box should display the word 'Complete'

   c. The NMAKE window that displayed during compilation and linking should have been closed and removed from the screen.

   d. The words '.EXE File Current' should appear in the lower right corner of the window.
7. Close the RTG.EXE translator window by clicking on **File**. When the File menu appears, click on Exit. When the Save Configuration File query appears, click on No.

Exercise 2 Files Created — The following files should have been created in the drive/directory where you loaded the EXAMPLE2.PRG file:

EXAMPLE2.C
EXAMPLE2.OBJ
EXAMPLE2.MAK
EXAMPLE2.LNK
EXAMPLE2.EXE
DM501O.C
DM5010.H
DM5010.OBJ
PS501OG.C
PS501OG.H
PS5010G.OBJ

Printing Exercise 2 Files — You can print copies of the Example 2 files (except .OBJ and .EXE) as follows:

- To print the .PRG file, start IPG from the TekTMS Application Group window by double-clicking on the IPG icon. When the IPG window appears, click on the **File** command. When the File menu appears, click on the **Qpen** command. When the file list appears, double-click on the EXAMPLE2.PRG file. When the test procedure appears, click on the **File** command. When the File menu appears, click on the **Print** command. When the print dialog box appears, click on the **Full** option, then on **OK.** When finished printing, exit the program.
- To print the .ISD, .C, .H, .MAK, and .LNK files, use the **Print** command from the Windows File menu.

NOTE

The .OBJ and .EXE files can not printed.

Running the Exercise 2 Executable (.EXE) File — To run the DOS-executable file for Exercise 2, you will need a TM500x mainframe containing a Tektronix DM5010 and PS5010G and a GPIB interface to run them from your controller. The HIGH and LOW outputs from the PS5010G must be connected from the POSITIVE and COMMON inputs of the DM5010. The TM500X mainframe must be connected to the controller with a GPIB cable. The DM500 must be set to address 16 and the PS501G must be set to address 22.

To run the Exercise 2 executable file from the DOS prompt:

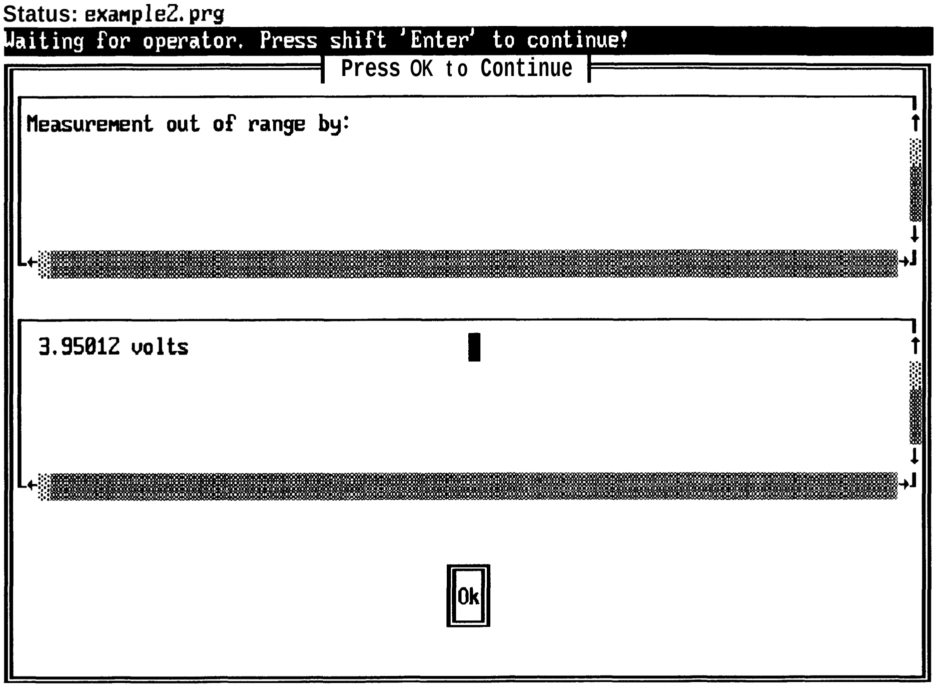
1. Type:

```
example2
```

and press RETURN

If no displays appear and the DOS prompt returns, the example is finished running.

To demonstrate what happens when the PS5010G doesn't set the voltage specified in the Example 2 test procedure, disconnect one of the cable connections and run the example again. Each time the program measures the power supply voltage, the following error display should appear. The example sets the power supply five times, so each time OK is selected the display appears again until the program has sequenced through all five settings. Then the DOS prompt returns.



```
Status: example2.prg
Waiting for operator. Press shift 'Enter' to continue!
Press OK to Continue

Measurement out of range by:

3.95012 volts
```

Familiarization Exercise 3

This exercise demonstrates how to create a library of instrument drivers, then use a .PRG test procedure file and the created library to create a DOS executable file.

To create the instrument driver library:

1. From the TekTMS Application Group window, start the Instrument Driver Translator (IDG.EXE) by double-clicking on the IDG icon.
2. When the IDG Translator Application window appears, click on **Options.**
3. When the Set Options dialog box appears, click on the **Include .OBJ in** lib: check box, type VXIDEMO in its edit box, and click on **OK.**
4. When the IDG Translator Application window appears, click on **Select ISD's....**
5. When the Select ISD's dialog box appears, double-click on the VX5260.ISD and 5790.ISD file names in the files list, then click on **OK.**
6. When the IDG Translator Application window appears, click on **Translate.**
7. While the translator is running, file and translation status should appear in the **Selected files:** and **Operation:** status boxes. When the compilation and linking process starts, an NMAKE window should appear and display compilation and linking status.
8. When the translation process ends, the screen and IDG Application window should appear as follows:
   a. The **Selected files:** status box should list the DX5260F.ISD and DX5790.ISD file names with X's in the .C and .OBJ columns.
   b. The **Operation:** status box should display the word 'Complete'.
   c. The NMAKE window that displayed during compilation and linking should have been closed and removed from the screen.
   d. The words **.LIB File Current** should appear in the lower right corner of the window.
9. Close the IDG.EXE translator window by clicking on **File.** When the File menu appears, click on **Exit.** When the Save Configuration File query appears, click on **No.**

To translate the main test procedure for VXIDEMO.PRG:

1. From the TekTMS Application Group window, start the Test Procedure Translator (RTG.EXE) by double-clicking on the RTG icon.
2. When the RTG Translator Application window appears, click on **Select Tests....**
3. When the Translate Programs dialog box appears, double-click on the VXIDEMO.PRG file name in the files list, then click on **OK.**
4. When the RTG Translator Application window appears, click on **Options.**

5. When the Translate Options dialog box appears, click on Link options.
6. When the Llnk Options dialog box appears, click on the Link **ISD's** from .LIB check box.
7. When a message box appears to indicate that the Translate referenced **ISDs** option has been disabled, click on OK.
8. Click on *.OBJ in the File: edit box to highlight it, then type: *.LIB and press the RETURN key to list the .LIB files in the current directory.
9. Double-click on the VXIDEMO.LIB file name to add it to the Selected files: list (LTEKRUN.LIB should already be in the Selected files: list). Click on OK.
10. When the Translate Options dialog box appears, click on OK.
11. When the RTG Translator Application window appears, click on Translate.
12. While the translator is running, file and translation status should appear in the Selected files: and Operation: status boxes. When the compilation and linking process starts, an NMAKE window should appear and display compilation and linking status.
13. When the translation process ends, the screen and RTG Application window should appear as follows:
    a. The Selected files: status box should list the VXIDEMO.PRG file name with X's in the .C and .OBJ columns.
    b. The Operation: status box should display the word 'Complete'
    c. The NMAKE window that displayed during compilation and linking should have been closed and removed from the screen.
    d. The words '.EXE File Current' should appear in the lower right corner of the window.
14. Close the RTG.EXE translator window by clicking on **File.** When the File menu appears, click on Exit. When the Save Configuration File query appears, click on No.

Exercise 3 Files Created — The following files should have been created in the drive/directory where you loaded the VXIDEMO.PRG, VX5260.ISD, and VX5790.ISD files.

VXIDEMO.EXE
VXIDEMO.C
VXIDEMO.OBJ
VXIDEMO.MAK
VXIDEMO.LNK
VX5260.C
VX5260.H
VX5260.OBJ
VX5790.C
VX5790.H
VX5790.OBJ
VXIDEMO.LRF
VXIDEMO.LIB

Printing Exercise 3 Files — You can print copies of the VXIDEMO files (except .OBJ and .EXE) as follows:

- To print the .PRG file, start IPG from the TekTMS Application Group window by double-clicking on the IPG icon. When the IPG window appears, click on the **File** command. When the File menu appears, click on the Qpen command. When the file list appears, double-click on the VXIDEMO.PRG file. When the test procedure appears, click on the **File** command. When the File menu appears, click on the Print command. When the print dialog box appears, click on the Full option, then on OK. When finished printing, exit the program.
- To print the .ISD, .C, .H, .MAK, .LRF, and .LNK files, use the Print command from the Windows File menu.

NOTE

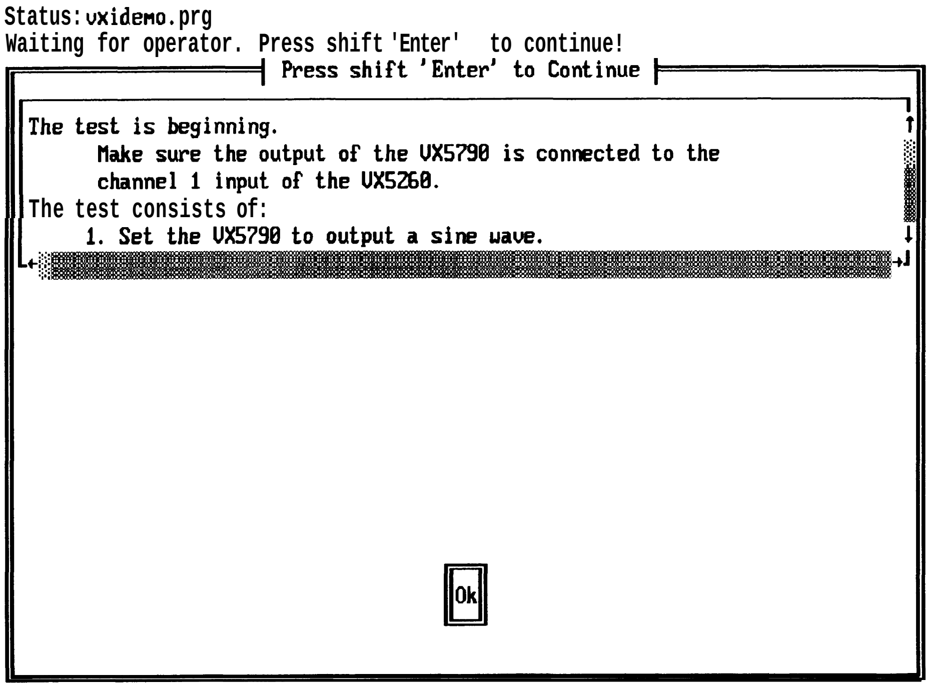
The *.OBJ,* .LIB, and .EXE files cannot printed.

Running the Exercise 3 DOS Executable (.EXE) File — To run the DOS-executable (.EXE) file for Exercise 3, you will need a VXI mainframe containing a Tektronix VXI controller, a VX5260, and a VX5790. Connect the output of the VX5790 to the Channel 1 input of the VX5260. The VX5260 must be named VX5260 and the VX5790 must be named VX5790 by your VXI system SURM (Start Up Resource Manager).

To run the Exercise 3 executable (.EXE) file from the DOS prompt, type:

```
vxidemo
```

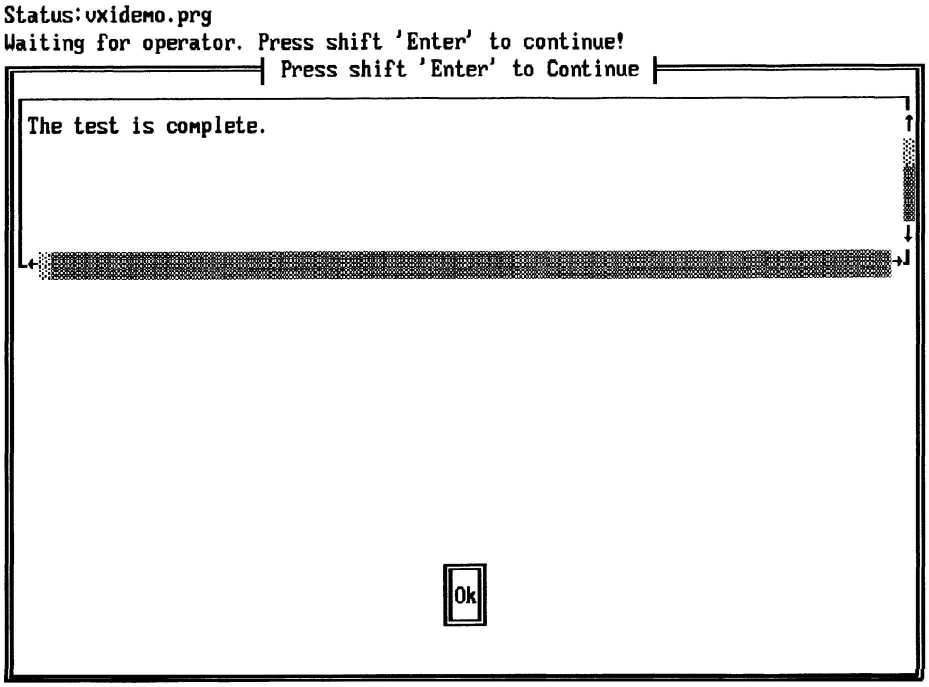
and press RETURN. The following display should appear to let you know the test is starting and what actions you need to take during the test.



```
Status:vxidemo.prg
Waiting for operator. Press shift 'Enter'  to continue!
Press shift 'Enter' to Continue

The test is beginning.
      Make sure the output of the VX5790 is connected to the
      channel 1 input of the VX5260.
The test consists of:
     1. Set the VX5790 to output a sine wave.
```

Use the scroll bar at the right side of the display to read the entire display text. When you are finished with the display, click on **OK.** The next two displays to appear should be a sine wave and a square wave. To continue from those displays, click on **OK.**

Following is the last display that should appear. Click on **OK** to return to the DOS prompt.



```
Status:vxidemo.prg
Waiting for operator. Press shift 'Enter' to continue!
Press shift 'Enter' to Continue

The test is complete.
```

Section 3
The Translation Process

The Translation Process

Figure 3-1 shows a diagrams of the overall TekTMS/RTG translation process with an example. In the process, the MAIN.PRG test procedure calls two other procedures (CALL1.PRG and CALL2.PRG) and uses two Instrument Drivers (REFA.ISD and REFB.ISD). The RTG.EXE Translator (Test Procedure Translator) and IDG.EXE Translator (Instrument Driver Translator) are independent applications, except during creation of the DOS-executable file. During creation of the DOS-executable file, the RTG.EXE translator starts and controls the entire process.

RTG.EXE translates .PRG files, while IDG.EXE translates .ISD files. When operated independently, either translator can save a .CFG file, create a .MAK file, control the compiler and linker, and create libraries. Only the RTG.EXE translator can create a DOS-executable file.

The amount and type of processing depends upon how many and what types of files are involved and the options selected. A translation process may involve only a Main test procedure without instrument drivers, or a Main test procedure with multiple called procedures and many instrument drivers. The translation process may execute completely to create a DOS-executable file using default options. Or the process may stop after translating a single test procedure or instrument driver, thus allowing the user to modify the generated C code.

Configuration Files

Configuration Files provide a time-saving user convenience for repeating the translation process, creating test procedure or instrument driver libraries, and updating DOS-executable files. When functioning independently, each translator maintains configuration information about its current file and option selections. This configuration information can be saved at any time during the translation session.

Both translators prompt the user to save the current Configuration file anytime it changes and the next action resets or loads a new configuration, or exits the session. Users should always save the Configuration file.

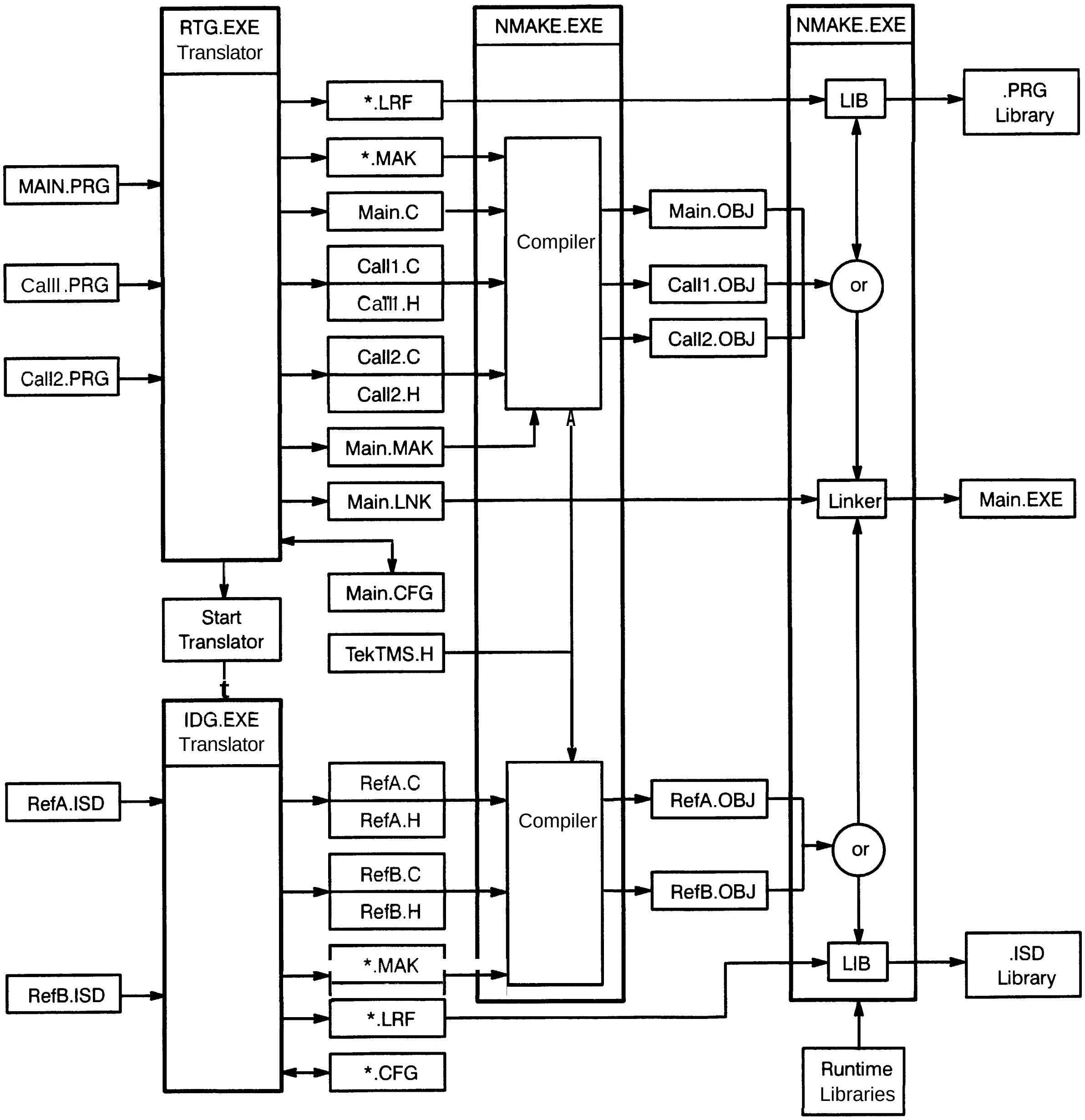


Figure 3-1: The TekTMS/RTG Translation Process

Configuration File Names

Each translator application window title bar displays the word -(untitled) when its configuration information hasn't been saved into a Configuration file. Otherwise, the title bars displays the name of the Configuration file with a .CFG extension. Generally, when saving a Configuration file, the translator assigns the root name of a .EXE or .LIB target file to it; otherwise, it uses the user assigned name. Translators assign Configuration file root names as follows:

1. On a new configuration having a main test procedure and using default options, the RTG.EXE translator assigns the test procedure root name to the .EXE file, which keys the name for the Configuration file.
2. When creating test procedure or instrument driver libraries the user must select an **Include .OBJ in library** compiler option and provide a library name. Both translators key the name for the Configuration file to the .LIB file.
3. When a Configuration file is open and the **Save As** command is invoked, the translator opens a Save CFG File dialog box prompting the user for a new file name. When the user enters and saves the new file name, the translator saves a new Configuration file and changes the title bar to the new file name.
4. When the **Save** command is invoked by either translator for saving '-(untitled)' configuration information for translation actions other than creating .EXE or .LIB files, the translator opens a Save CFG File dialog box prompting the user for a new file name. When the user enters and saves the new file name, the translator saves the new Configuration file and changes the title bar from '-(untitled)' to the new file name.

NMAKE.EXE

The Microsoft® MAKE Utility automates program development by providing a program to update target files automatically whenever changes are made to one of its source files. Microsoft® V6.0 provides both an NMAKE.EXE and NMK.COM version of its MAKE Utility while Microsoft C V7.0 supplies only NMAKE.EXE. During the installation of TekTMS/RTG you were asked which version of C you would be using and the install program saved your selection in the WIN.INI file. So RTG will call NMK.COM for V6.0 or NMAKE.EXE for V7.0. or later version. The TekTMS/RTG translators are aware of this difference and create .MAK files appropriate for each utility. A .MAK file provides the target, source dependency lists, commands, and rules needed during processing.

NOTE

NMK.COM and NMAKE.EXE run automatically only in Windows Enhanced Mode. If you are running Windows in Standard mode, the Make file *(.MAK)* and Link Response file *(.LNK)* are generated, but the make utility is not started automatically. Instead, the application prompts the user to start the make utility

.MAK Files

.MAK files contain the target, source dependence list, command, and rule information needed by NMAKE.EXE to run. When NMAKE.EXE executes the .MAK file, it creates or updates any target file in the translation sequence where changes have been made to its related source(s). Anytime a target's system date/time stamp is older than its source's date/time stamp, NMAKE.EXE executes the commands and rules in the .MAK file to update the target. Source dependency lists are the source files required to create the target file. Commands are DOS commands or programs that execute to create the target file. Rules are instructions to NMAKE.EXE for creating one type of file from another type of file.

When functioning independently, each translator creates its own .MAK file. When a DOS-executable file is built, the RTG.EXE translator's .MAK file includes the Make file information for the IDG.EXE Translator.

The following example from a .MAK file shows how to create a .EXE file from .C files.

```
test1.exe: test1.obj dm 5010.obj ps5010.obj
   link /NOE /STACK:8192 @test1.lnk

.c.obj:
   cl -c -W4 -u -Gs -Ot -AM $*.c
```

TEST1.EXE is the target file. When NMAKE.EXE finds the target file is older than any one, or all, of the TEST1.OBJ, DM010.OBJ, or PS5010.OBJ files, it rebuilds the target. To create or rebuild TEST1.EXE, the NMAKE.EXE issues the command line 'link /NOE /STACK:8192 @test1.lnk’. This command line runs the linker with certain options and reads the list of files from a linker response file named TEST1.LNK.

The line, C.OBJ:, is a rule. This rule states that to create .OBJ files from .C files, execute the command ‘cl -c -W4 -u -Gs -Ot -AM $*.c’. For each file that must be compiled to create the .OBJ file, a root file name replaces $*. New .OBJ files are created when they are older than the .C files. The example above compiles any .C files newer than their .OBJ’s, and then links the .OBJ’s to create a .EXE file. If TEST1.EXE is newer than any of the .OBJ’s, no processing occurs. For an explanation of each option in the command line, refer to the Microsoft C Users Guide.

.MAK File Names

Both translators key their .MAK file names to the name of the target files (.EXE or .LIB and add a .MAK extension (see the Configuration File Names topic).

Runtime Libraries

The runtime libraries block shown in Figure 3-1 represents Microsoft, TekTMS, and User Defined libraries. NMAKE.EXE and the Linker selects only those runtime routines from the libraries that are required to build the DOS-executable files or libraries.

Translation of a Main Test Procedure

Figure 3-2 shows a flow diagram for the initial and subsequent translation of a Main.PRG test procedure using the following default options:

- Translate referenced ISD's
- Produce .EXE Named

Other default options not shown on the flow diagram are (the option selections are shown in bold text):

- Translate **all called tests**
- Memory Model **Large**
- Optimize **for speed**

Figure 3-1 shows the translation process to the point where the Microsoft NMAKE.EXE Make Utility starts the compiler and linker. Figure 3-2 shows the compilation and linking process. The **Selected File** and **Operation** blocks in the flow diagrams represent their respective status boxes in the Application window.

Translate Process

Before the translation process starts, the **Selected File** status box in the Application window is empty. The initial translation process starts by selecting a Main test procedure, then selecting the **Translate** pushbutton. Before translation starts, the RTG.EXE translator displays the names of all selected and called .PRG files in the **Selected File** status box. On an initial translation, the .C and .OBJ columns for each file in the **Selected File** status box are empty. As each .PRG file starts translating, its name in the selected files list is highlighted and the Operation status box displays the word 'Translating'. After each .PRG file is translated into a .C and a .H file, an X appears in the .C column of the **Selected File** status box. Translated files assume the same root name as its source .PRG file.

If any translated .PRG files referenced any .ISD files, the **Translate Referenced ISD's** option starts the IDG.EXE translator. If IDG.EXE is currently running, RTG.EXE sends it a list of .ISD files to translate with the 'C only' flag option and proper memory model size already set. All other options are left at defaults. If IDG.EXE is not currently running, RTG.EXE starts it with default options, then sets the 'C only' flag. Like the RTG.EXE translator, the IDG.EXE lists the .ISD files being translated, highlights their name as they are translated, and displays the word 'Translating' in the **Operation** status box. The .ISD files also translate into .C and .H files, and as each file is translated an X appears in the C column of the **Selected File** status box.

At this point, the **Produce .EXE Named** default option causes the translator to create a .MAK file to control the compiler and a .LNK file to control the linker. Once these two files are created, the process continues to the compilation and linking process shown in Figure 3-2.

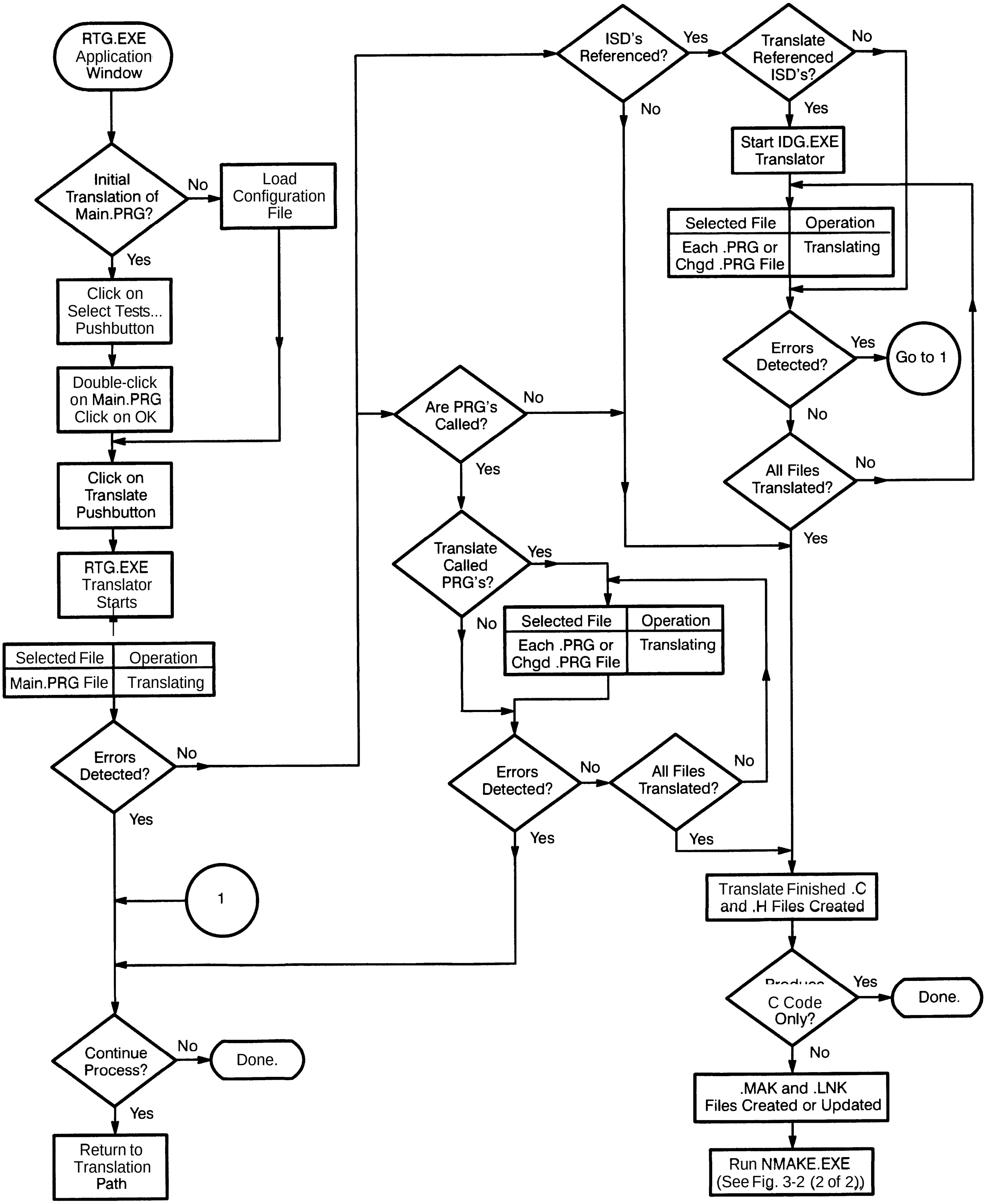


Figure 3-2: Main .PRG Process Using Defaults (1 of 2)

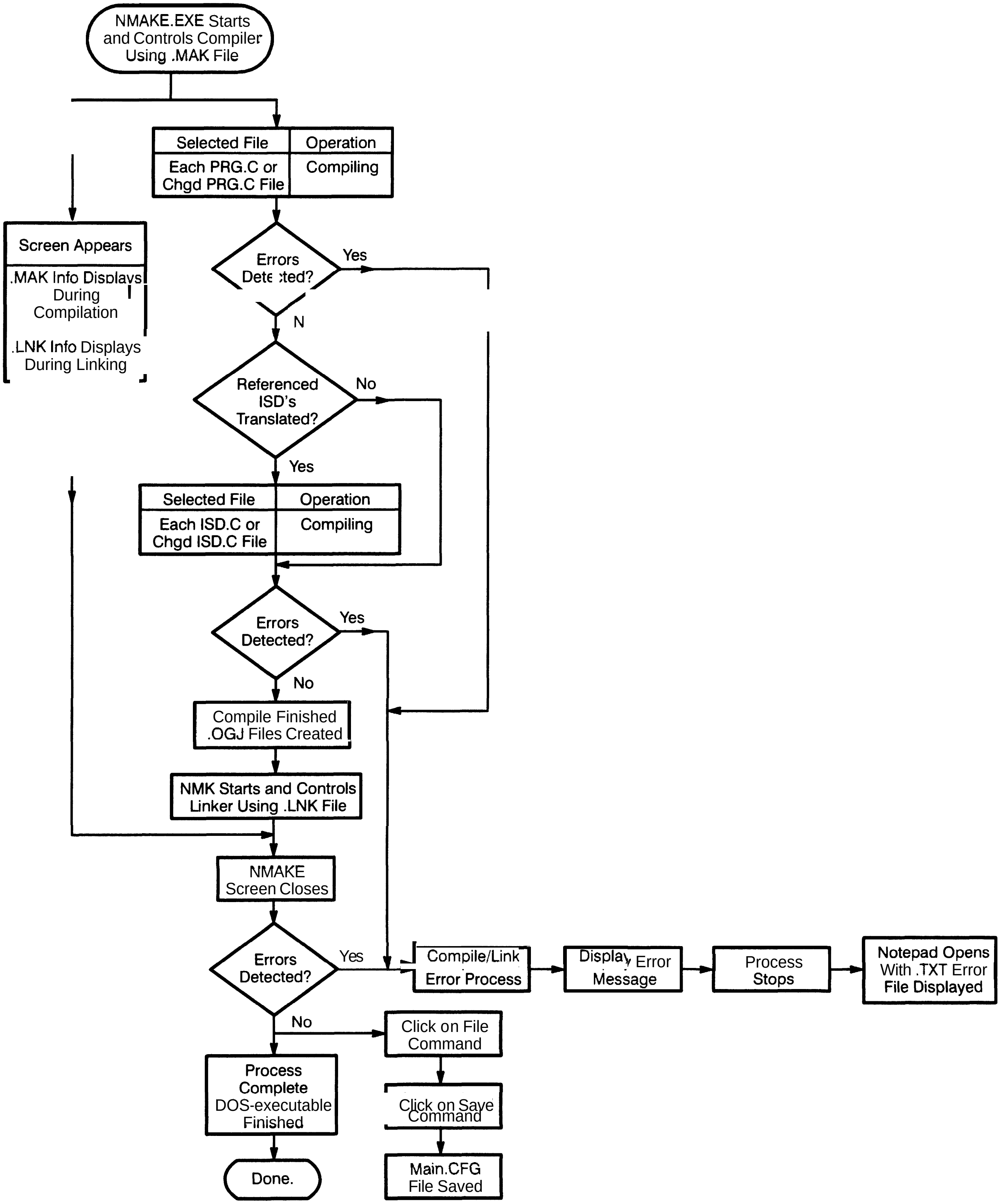

NMAKE.EXE Starts and Controls Compiler Using .MAK File
Selected File
Operation
Each PRG.C or Chgd PRG.C File
Compiling
Screen Appears
.MAK Info Displays During Compilation
.LNK Info Displays During Linking
Errors Detected?
Yes
N
Referenced ISD's Translated?
No
Yes
Selected File
Operation
Each ISD.C or Chgd ISD.C File
Compiling
Errors Detected?
Yes
No
Compile Finished .OGJ Files Created
NMK Starts and Controls Linker Using .LNK File
NMAKE Screen Closes
Errors Detected?
Yes
No
Compile/Link Error Process
Display Error Message
Process Stops
Notepad Opens With .TXT Error File Displayed
Click on File Command
Click on Save Command
Main.CFG File Saved
Process Complete DOS-executable Finished
Done.

Figure 3-2: Main .PRG Process Using Defaults (2 of 2)

Translator Errors

If any errors are detected during the translate process, a message appears asking if you want to stop or continue the process. If you continue, the translation process returns to the point of error and continues; otherwise, the process stops.

Compile and Link Process

When NMAKE.EXE starts, it uses the .MAK file as input.

During file compilation, the **Operation** status box displays the word 'Compiling'. Also during compilation, NMAKE sequentially displays each file description block in the .MAK file. This description block includes the target file name, its source file name(s), and the command line required to create the target file. As NMAKE displays the target and source file descriptions, it checks the modification dates of the files. If the modification dates are out-of-date, NMK displays and executes the .MAK file command line to create the target file, and shows that the translation process is running. When the command line finishes executing, the next file description block appears and the process repeats. This process of displaying file names, checking modification dates, and displaying and executing command lines continues until all target files in the .MAK file are compiled. When compilation finishes, the .OBJ column for all files in the **Selected file** status box are Xed.

Next, NMAKE.EXE starts the linker and uses the link response file (.LNK file) to control the linker to build the DOS-executable file. During the build process, any routines functions required from the C, TekTMS, or User libraries are copied into the executable.

Compiler and Linker Errors

If any errors are detected during compilation or linking, the process stops and the error data is logged to a .TXT file. Next, the translator starts Microsoft NOTEPAD and displays the .TXT error file. The root name of the .TXT file is the same as the target (DOS-executable file or library) or the .CFG file, in that order. If neither file exists, the name of the error file becomes MAKEFILE.TXT.

Saving the Configuration File

The three block action to save the Configuration file just before exiting the session could occur any place in the process after the selection of a Main test procedure. If you are building a new version or modifying the DOS-executable file or a library, you should do a **SaveAs** command action before making any changes.

Subsequent Translations Using Default Options

Subsequent translations of a Main test procedure using default options follow the same flow diagram path as the first translation, except as follows:

1. The process starts by opening a previously saved Configuration file.
2. When the Configuration file opens, the selected files appear in the **Selected File** list box in the Application window. The **Selected File** list box also shows the current modification status of the files with X's in the .C and .OBJ columns of each file. If the .EXE file is current when the Configuration file is opened, the Application window displays '.EXE File Current' in its lower right corner.
3. When the translation process starts, each file is examined by the translator, but only changed files are translated.

Subsequent Translations with Option Changes

The most frequently used options are expected to be:

- the selection of the **Produce C Code Only** option, which turns off the **Produce .EXE Named** option (Both options are in the Translate Options dialog box).
- the selection of a different Memory Model (Compile Options dialog box).
- the selection of the **Link ISD's from .LiB File** option, which turns off the **Translate Referenced ISD's** option.

Produce C Code Only Option (Translate Option)

The **Produce C Code** Only option allows users to separately translate modified .PRG programs or added user C modules for verification or debugging. When you select the **Produce C Code Only** option, the **Produce .EXE Named** option turns off and is grayed. With the **Produce C Code Only** option selected, the translation process stops after all files are translated, or in the case of subsequent translations, are updated. To reactivate the **Produce .EXE Named** option, the **Produce C Code Only** option must be turned off, then the **Produce .EXE Named** option must be reselected.

Memory Model (Compile Option)

When the Memory Model changes, NMAKE takes the following actions:

1. All files (.PRG and .ISD) are examined to verify their modification date(s). Any out-of-date files are translated.
2. A new .MAK file is created.
3. All files are compiled using the current memory model.
4. All files are linked using the current memory model's associated runtime library.

Include .OBJ in Library Option (Compile Option)

This option is selected from the RTG.EXE translator's Compile Options dialog box when building test procedure libraries. When selected, it turns off and grays the **Produce .EXE named** option in the Translate Options dialog box. It also disables and grays the **Main** designation pushbutton in the Translate Programs dialog box. With the **Main** pushbutton disabled, none of the selected test procedures are designated as Main.

This option also is selected from the IDG.EXE translator's Set Options dialog box when building instrument driver libraries.

Link ISD's from .LIB File Option (Link Option)

When you select the **Link ISD's from .LIB File** option and specify a .LIB file name, it turns off the **Translate Referenced ISD's** option. With the **Link ISD's from .LIB File** option selected, TekTMS/RTG looks for all referenced ISD's in the specified .LIB file. To create instrument driver libraries, see Section 5, Using the Instrument Driver Translator.

Section 4
Using the Test Procedure Translator

Using the Test Procedure Translator

Table 4-1 is a guide to the various tasks you can perform using the RTG.EXE Test Procedure Translator.

To use the table, first start the RTG.EXE application. When the application window appears, find the task you want to accomplish in the left column. Then go to the topic listed in the second column and follow the instructions.

Table 4-1: RTG.EXE Translator Task Selector

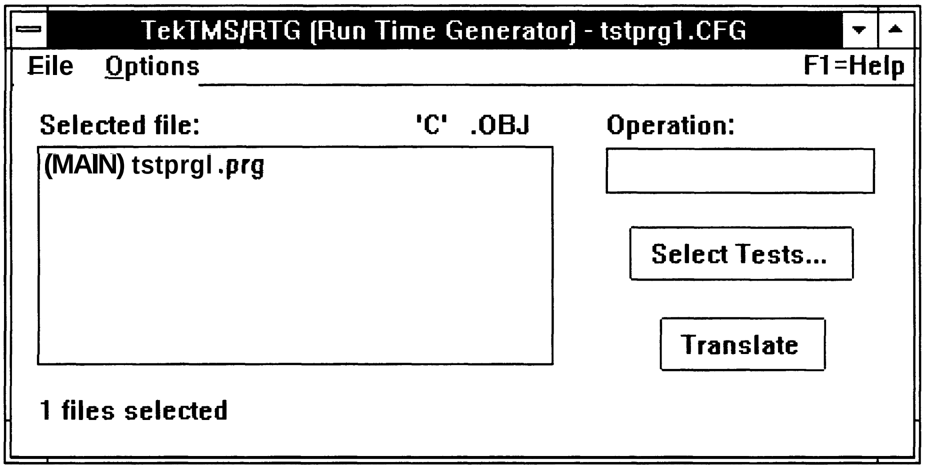
Task	See Topic
Create a DOS executable file from a main test procedure using default option selections	Translation of a Main Test Procedure on page 4-2.
Create or update libraries of test procedures.	Creating Test Procedure Libraries on page 4-3 or Updating Test Procedure Libraries on page 4-5.
Create a DOS executable file using a main test procedure with associated test procedure libraries and instrument driver libraries.	Creating a DOS Executable File Using Libraries on page 4-5.
Update a DOS executable file when an original test procedure, referenced instrument driver, or library file changes.	Updating a DOS Executable File on page 4-8.
Update a DOS executable file by adding or deleting test procedures.	
Update a DOS executable file when a test procedure library or instrument driver library file changes.	

Translating a Main Test Procedure into a DOS Executable File

The most common use of TekTMS/RTG is to create a DOS executable file by selecting a Main test procedure and translate using default options.

To create a DOS executable file with a Main test procedure, which has never been translated, using default options:

1. Starting at the TekTMS/RTG Application window, click on the **Select Tests...** pushbutton. The Translate programs dialog box should appear.
2. Double click on the Main test procedure file name. If the desired file name doesn't appear in the files list, use standard Windows techniques to move to the directory where it resides, then double-click on its name.
3. Click on the OK pushbutton. An Application window similar to the example below should appear.



Selected file:

(MAIN) tstprgl.prg

Operation:

1 files selected

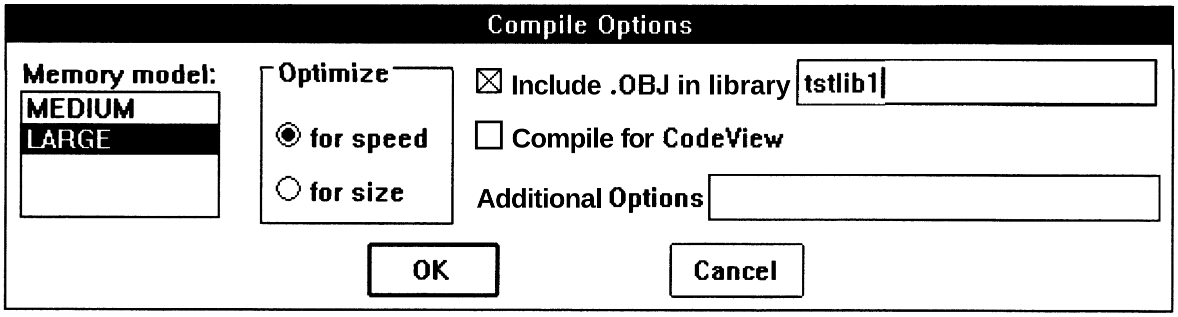
4. Click on the **Translate** pushbutton to start the translation process. For an explanation of the operations that appear during the translation process, refer to the Translation of a Main Test Procedure topic in Section 3, The Translation Process.
5. When the translation process finishes without errors, the Application window displays the words '.EXE is current' in its lower right corner, and the **Operation** status box displays 'Complete'. If an error occurs during translation, an error message appears and you are asked if you want to continue the translation process. If an error occurs during the compile or link process, the error is displayed and logged, then the process stops.
6. Exit the translator.

Creating a Test Procedure Library

A test procedure library can save you translation process time when creating DOS executable files using multiple main test procedures that call various sequences from a select group of test procedures. This select group of test procedures can first be selected, translated and compiled, and combined into a test procedure library. Then when each main test procedure goes through the translation process, only the main test procedure is translated. Any test procedures called by the main test procedure are selected by the linker from the library and added to the program. Thus the called test procedures only need to be translated and compiled once for the libraries rather than each time a DOS executable file is built.

To create a test procedure library:

1. From the Applications window, click on the **Qptions** command. The Translate Options dialog box should appear.
2. Click on the **Compile Options...** pushbutton. A Compile Options dialog box similar to the one below should appear.



Compile Options

Memory model: MEDIUM, LARGE (selected)

Optimize: ● for speed ○ for size

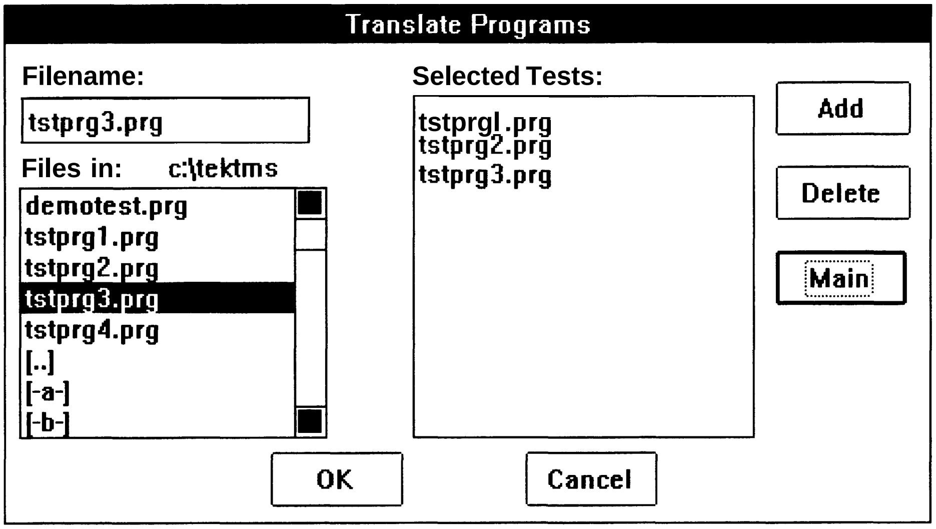
☒ Include .OBJ in library tstlib1

□ Compile for CodeView

Additional Options

3. Click on the **Include .OBJ in library:** check box to enable it (box is Xed). Then type a library name in the edit box. Making this option selection turns off the **Produce .EXE named** option in the Translate Options dialog box. It also prevents any test procedure file selected for the library to be designated as (Main).
4. Select the **OK** pushbutton. The Translate Options dialog box should appear. Note that the **Produce .EXE named** check box and edit box are grayed.
5. Select the **OK** pushbutton. The Application window should appear.

6. Click on **Select Tests... A** Translate Programs dialog box similar to the one below should appear.



Translate Programs

Filename: tstprg3.prg

Files in: c:\tektms

demotest.prg
tstprg1.prg
tstprg2.prg
tstprg3.prg
tstprg4.prg
[..]
[-a-]
[-b-]

Selected Tests:

tstprgl .prg
tstprg2.prg
tstprg3.prg

7. Double-click on each file name in the files list to be included in the library. Each selected file should appear in the **Selected files:** list box.

8. Click on **OK.** The Applications window should appear.

9. Click on the **Translate** pushbutton. The screen displays that appear during the translation process are similar to those described in the Translation of a Main Test Procedure topic in Section 3, The Translation Process, except as follows:

   a. There is no Main .PRG file.

   b. The translator builds a .LRF file (Library Response File) rather than a .LNK file (Link Response File).

   c. The process builds a library with a .LIB extension rather than a .EXE file.

10. When the translation process finishes without errors, the Application window displays the words '.LIB is current' in it lower right corner, and the **Operation** status box displays 'Complete'. If an error occurs during the translate process, an error message appears. Then the translator asks you if you want to continue or stop the process. If you continue, the process finishes the translate process and starts the compile and link process. If an error occurs during the compile or link process, the error is displayed and logged to a .TXT file with the root name of the .LIB file, then the process stops.

11. Exit the translator.

Updating a Test Procedure Library

To update a test procedure library:

1. From the Applications Menu, click on the **File** command. The File menu should appear.
2. Select the **Qpen** command. The Open CFG File dialog box should appear.
3. Double-click on the Configuration file name to be updated. The Applications window should appear.
4. If you are going to replace the old library, go to step 5; otherwise, continue to keep the old library and create a new one. Select the **File** command. The File menu should appear. Select the **Save As...** command. The Save CFG File dialog box should appear with the opened Configuration file name in the **Config file:** edit box. Enter a new name in the **Config file:** edit box and click on **Save.** The Applications window should appear. Click on the **Qptions** command. The Translate Options dialog box should appear. Click on the **Compile options...** pushbutton. The **Compile Options** dialog box should appear. Type a new library name in the **Include** .OBJ in library edit box. Click on the **OK** pushbutton in each option dialog box until the Application window appears.
5. Make needed changes as follows.
   a. If the change only involves updating of the original .PRG files, go to step 6.
   b. If test procedures need to be added or deleted from the library, click on the **Select Tests...** pushbutton. When the Translate Programs dialog box appears, add or delete test procedures as needed, then select **OK.** The Application window should appear.
   c. If options need to be changed, select the appropriate options dialog boxes and make the changes. Click on **OK** in each option dialog box until the Application window appears.
6. Click on **Translate.**

NOTE

At this point, the instructions are the same as steps *10* through *11* of the Creating a Test Procedure Library topic.

Creating a DOS Executable File, Including Libraries

When a DOS executable file is created using test procedure and instrument driver libraries, there must be a main test procedure. This main procedure must include calls to test procedures in the test procedure library. Also, the main procedure must reference all instrument drivers used by the test procedures so the linker can select them from the instrument driver library.

Before starting, create a main test procedure and the libraries for the test procedure instrument driver.

To create a DOS executable file using libraries:

1. From the Applications window, click on the **Select Tests...** pushbutton. The Translate Programs dialog box should appear.
2. Double-click on the main test procedure name in the files list, then click on **OK.** The Application window should appear.
3. Click on the **Options** command. The Translate Options dialog box should appear.
4. Click on the **Translate referenced ISD's** check box to turn it off (the X should disappear).
5. Verify that the **Produce .EXE named** check box is selected and a file name using the root name of the main test procedure appears in its edit box.
6. Click on the **Link Options...** pushbutton. The Link Options dialog box should appear.
7. In the Link Options dialog box:
   a. Click on the **Link** ISD's from .LIB file check box to enable it (check box should be Xed).
   b. Double-click on the test procedure and instrument driver library names in the **Available files:** list box to add them to the **Selected files:** list box.

NOTE

The **Selected files:** list box should include the name of one of the *TekTMS* runtime libraries (such as *xTEKRUN*.LIB where x is a letter denoting the memory model size). **DO NOT DELETE** this file as it may cause the link process to fail.

8. Click on **OK.** The Compile Options dialog box should appear. Click on the **OK** pushbutton. The Application window should appear.

9. Click on **Translate.** The screen displays that appear during the translation process are nearly identical to those described in the Translation of a Main Test Procedure topic in Section 3, The Translation Process, except as follows:

   a. Since the libraries are already translated and compiled, the **Selected file** status box only shows the main test procedure being translated and the NMAKE window only shows it compiled and linked into the .EXE.

   b. The NMAKE window shows the linking of each called test procedure and referenced instrument driver.

10. When translation finishes without errors, the Application window displays the words '.EXE is current' in its lower right corner and the **Operation** status box displays 'Complete'. If an error occurs during the translate process, an error message appears. Then the translator asks you if you want to continue or stop the process. If you continue, the process finishes the translate process and starts the compile and link process. If an error occurs during the compile or link process, the error is displayed and logged to a .TXT file with the root name of the .EXE file, then the process stops.

11. Exit the translator.

Updating a DOS Executable File

Updating a DOS Executable File from Updated Original Files

Before starting, update the source files.

To update a DOS executable file when the source files change:

1. From the Application window, select the **File** command.
2. When the File menu appears, select the **Qpen** command.
3. When the Open CFG File dialog box appears, double-click on the name of the Configuration file used to build the original DOS executable file. The Application window should appear.
4. If you are going to replace the old executable file, go to step 5; otherwise, continue to keep the old file and create a new one. Select the **File** command. The File menu should appear. Select the **Save As...** command. The Save CFG file dialog box should appear with the opened Configuration file name in the **Config file:** edit box. Enter a new name in the Config file: edit box and click on **Save.** The Applications window should appear. Click on the **Options** command. The Translate Options dialog box should appear. Type a new name in the **Produce .EXE named** edit box. Click on **OK.** The Application window should appear.
5. Click on **Translate.** The screen displays that appear during the translation process are nearly identical to those described in the Translation of a Main Test Procedure topic in *Section* 3, *The Translation Process,* except as follows:
   a. Only changed files are translated.
   b. The NMAKE Make Utility examines each file to determine its modification status, but only compiles changed files.
   c. The process builds a new .MAK file.
6. When the translation process finishes without error, the Application window displays the words '.EXE is current' in the lower right corner, and the **Operation** status box displays 'Complete'. If an error occurs during the translate process, an error message appears. Then the translator asks you if you want to continue or stop the process. If you continue, the process finishes the translate process and starts the compile and link process. If an error occurs during the compile or link process, the error is displayed and logged to a .TXT file with the root name of the .EXE file, then the process stops.
7. Exit the translator.

Updating a DOS Executable File by Adding or Deleting Test Procedures

When adding or deleting test procedures, other than the main test procedure, you need to be aware of what actions the translator takes:

- If you add a test procedure to the **Selected files:** list, but don't include a call to it in the main test procedure, the file is translated, compiled, and linked into the DOS executable file, but the main procedure never calls it. Also the Application window shows the .EXE is current even though it isn't. Thus, any added test procedure also must be included in the main test procedure as a called file.
- If you delete a called test procedure from the **Selected file:** list, but don't delete the call in the main test procedure, and the **all called tests** translate option is selected, the translator automatically adds the file to the **Selected file:** list. Thus, any called test procedure that is deleted must have its call step removed from the main test procedure. In this case, there shouldn't be any processing action and the Application window should show the .EXE is current.
- If you delete a called test procedure that isn't called in the main test procedure the process shouldn't do anything because all the files required for the DOS executable file are up-to-date. The Application window should show the .EXE is current. The original .EXE will contain the translated, compiled, and linked called test procedure. If you want to delete the file from the executable, you must delete the .EXE file and run the translator again.

Before starting the following procedure, use the information above to update the main test procedure.

To update a DOS executable file by adding or deleting test procedures:

1. From the Application Menu, select the **File** command. The File menu should appear.
2. Click on the **Qpen** command. The Open CFG File dialog box should appear.
3. Double-click on the name of the Configuration file used to build the original DOS executable file. Click on **OK.** The Application window should appear.
4. If you are going to replace the old executable file, go to step 5; otherwise, continue to keep the old file and create a new one. Select the **File** command. The File menu should appear. Select the **Save As...** command. The Save CFG File dialog box should appear with the opened Configuration file name in the **Config file:** edit box. Enter a new name in the **Config file:** edit box and click on **Save.** The Applications window should appear. Click on the **Qptions** command. The Translate Options dialog box should appear. Type a new name in the **Produce .EXE named** edit box. Click on **OK.** The Application window should appear.

5. Click on the **Select Tests...** pushbutton.

6. When the Translate Programs dialog box appears:

    a. Double-click on file names in the files list to be added to the DOS executable file.

    b. Click on file names in the **Selected Tests:** list box to be deleted from the DOS executable file, then click on the **Delete** pushbutton.

7. Select the **OK** pushbutton. The Application window should appear.

8. Click on the **Translate** pushbutton. The screen displays that appear during the translation process are nearly identical to those described in the Translation of a Main Test Procedure topic in Section 3, The Translation Process, except as follows:

    a. Only the added and changed files are translated.

    b. If the main test procedure includes changes, the translator builds new .MAK and .LNK files. If a called file is added, but the main procedure isn't changed, a new .LNK file is built.

    c. The NMAKE Make Utility examines each file to determines its modification status, but only compiles the new files. The linker executes the new .LNK file to build a new .EXE file.

9. When the translation process finishes without error, the Application window displays the words '.EXE is current' in the lower right corner, and the **Operation** status box displays 'Complete'. If an error occurs during the translate process, an error message appears. Then the translator asks you if you want to continue or stop the process. If you continue, the process finishes the translate process and starts the compile and link process. If an error occurs during the compile or link process, the error is displayed and logged to a .TXT file with the root name of the .EXE file, then the process stops.

10. Exit the translator.

Updating the DOS Executable File When Libraries Change

TekTMS/RTG does not check dependencies of library files when creating a DOS executable file. Thus, if the only changes to a DOS executable file are changes to files in the libraries, you must either delete the .EXE file to force the rebuilding of an updated .EXE, or create a new .EXE file.To update a DOS executable file when libraries change:

Before starting this procedure, you must make the necessary changes to the libraries and translate, compile, and rebuild them. Also you must make any needed changes to the main test procedure.

1. If you are going to create a new .EXE file go to step 2. now; otherwise, if you are going to replace the original .EXE file, delete the original .EXE file now.
2. From the Applications window, click on the **Eile** command. The File menu should appear.
3. Click on the **Open** command. The Open CFG File dialog box should appear.
4. Double-click on the Configuration file name used to create the original DOS executable file.
5. If you are going to replace the original .EXE file, go to step 6 now; otherwise, continue. Select the **File** command. The File menu should appear. Select the **Save As...** command. The Save CFG File dialog box should appear with the opened Configuration file name in the **Config file:** edit box. Enter a new name in the **Config file:** edit box and click on **Save.** The Applications window should appear. Click on the **Qptions** command. The Translate Options dialog box should appear. Type a new name in the **Produce .EXE named** edit box. Click on **OK.** The Application window should appear.
6. Click on the **Translate** pushbutton. The screen displays that appear during the translation process are nearly identical to those described in the Translation of a Main Test Procedure topic in Section 3, The Translation Process, except as follows:
   a. Since the libraries are already translated and compiled, the **Selected file** status box only shows a translate process on the main test procedure when it has changed.
   b. If the main test procedure changes, or the .EXE file has been deleted, the translator builds new .MAK and .LNK files.
   c. The NMAKE window shows any compilation required and the linking of files to update or rebuild the executable.

7. When the translation process finishes without errors, the Application window displays the words '.EXE is current' in its lower right corner and the **Operation** status box displays 'Complete'. If an error occurs during the translation process, an error message appears. Then the translator asks you if you want to continue or stop the process. If you continue, the process finishes the translation process and starts the compile and link process. If an error occurs during the compile or link process, the error is displayed and logged to a .TXT file with the root name of the .EXE file, then the process stops.

8. Exit the translator.

Section 5
Using the Instrument Driver Translator

Using the Instrument Driver Translator

Table 5-1 is a guide for the various tasks a TekTMS/RTG user might perform using the IDG.EXE Instrument Driver Translator.

To use the table, first start the IDG.EXE application. When the Application window appears, go to the topic listed in the second column and follow its instructions.

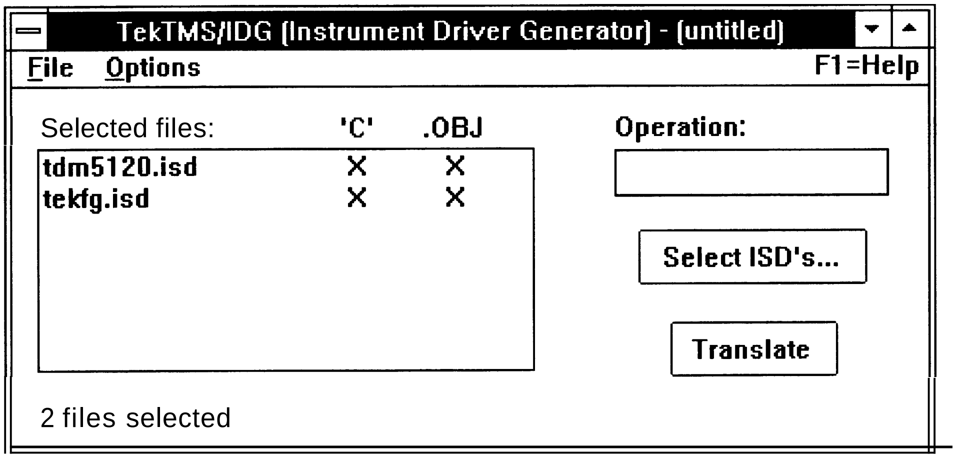
Table 5-1: IDG.EXE Translator Task Selector

Task	See Topic
Translate and compile a new instrument driver into a .OBJ file for creating or updating a DOS-executable file or an instrument driver library.	Translating an Instrument Driver on page 5-2.
Create an instrument driver library by selecting and translating instrument drivers (.ISD files) and saving them in a library file.	Creating an Instrument Driver Library on page 5-3.
Update an instrument driver library when original driver files change.	Updating an Instrument Driver Library on page 5-5.
Update an instrument driver library by adding an instrument driver to it or deleting an instrument driver from it.	

Translating an Instrument Driver

To translate an instrument driver:

1. Starting at the Application window, click on the Select ISDs.... pushbutton. The Translate ISDs dialog box should appear.
2. Double-click on the .ISD file name(s) to be translated.
3. Click on OK. The Application window should appear with the selected files listed in the Selected file status box.
4. Click on Translate. During translation, each file in the Selected file status box is highlighted and the Operation: status box displays the word 'Translating'. After each .ISD file translates, an X appears in the 'C' column of the Selected files status box similar to the example Application window below.



Selected files:	'C'	.OBJ
tdm5120.isd	X	X
tekfg.isd	X	X

2 files selected

After all .ISD files are translated, the IDG.EXE translator generates a .MAK file to control the Microsoft® NMAKE.EXE Make Utility. If the target file for the translated files is a library file, the root name of the .MAK file is the same as the target; otherwise, the root name is 'makefile'. The translator then starts the NMAKE.EXE Make Utility, which opens the NMAKE window and starts the compiler.

During compilation, the NMAKE window displays the .MAK file operations being executed by NMAKE.EXE for each file processed, and the Operation status box displays the word 'Compiling'. When all .ISD files are compiled, an X appears in the .OBJ column of the Selected files status box for all files.

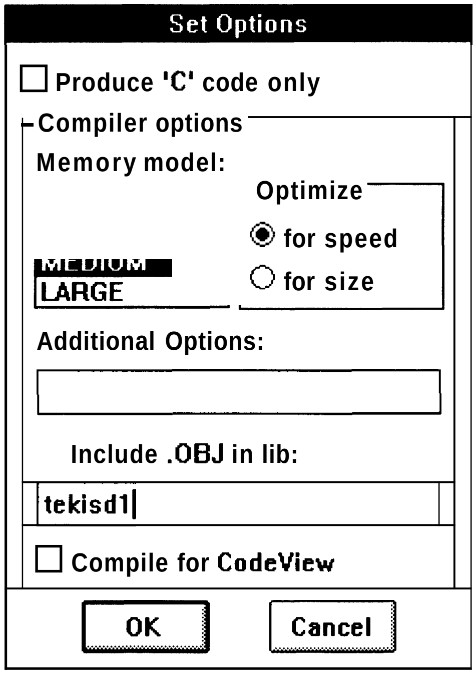
5. When the translation process finishes without errors, the **Operation** status box displays the words 'Complete' and the NMAKE window automatically closes. If an error occurs during the translate process, an error message appears. Then the translator asks you if you want to continue or stop the process. If you continue, the process finishes the translate process and starts the compile and link process. If an error occurs during the compile or link process, the error is displayed and logged, the process stops, and the translator starts Microsoft® NOTEPAD and displays the .TXT error file. The errors are logged to a .TXT file with the name MAKEFILE.TXT or the root name of the target.
6. Exit the translator.

Creating an Instrument Driver Library

An instrument driver library can save you translation process time when creating DOS-executable files. Because the instrument driver files in the library are already translated and compiled, they only need to be linked into the DOS-executable file when it is built. Thus the referenced instrument drivers only need to be translated and compiled once for the libraries rather than each time a DOS-executable file is built.

To create an instrument driver library:

1. From the Application window, click on the **Options** command. The Set Options dialog box should appear.
2. Click on the Include .OBJ in lib: check box to turn it on (box should be Xed) and type a library name in the edit box similar to the example dialog box below.



Set Options

□ Produce 'C' code only

Compiler options

Memory model: MEDIUM LARGE

Optimize

● for speed

○ for size

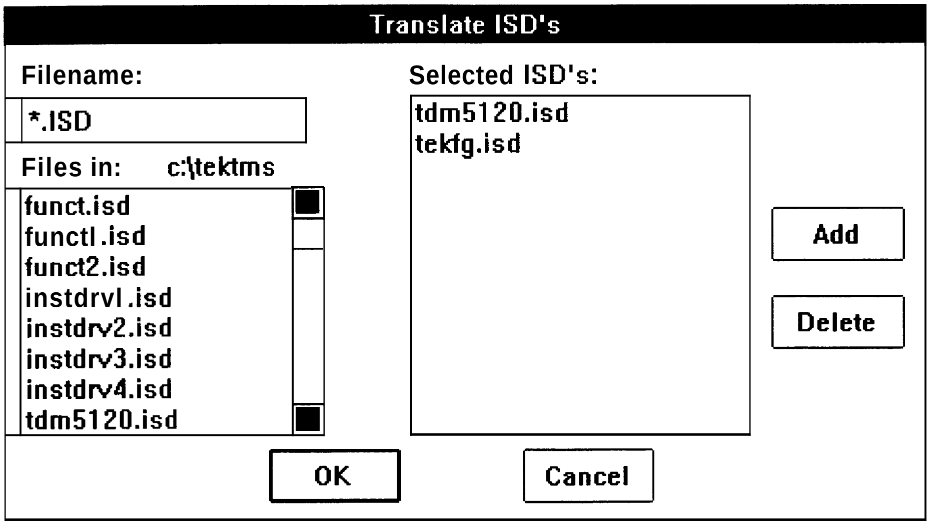
Additional Options:

Include .OBJ in lib:

tekisd1

□ Compile for CodeView

3. Click on OK. The Application window should appear.
4. Click on the **Select ISDs...** pushbutton. The Translate ISDs dialog box should appear.
5. Double-click on each .ISD file name to be included in the library. As each file is selected, its name should appear in the **Selected ISD's**: list box similar to the example shown below.



Filename: *.ISD

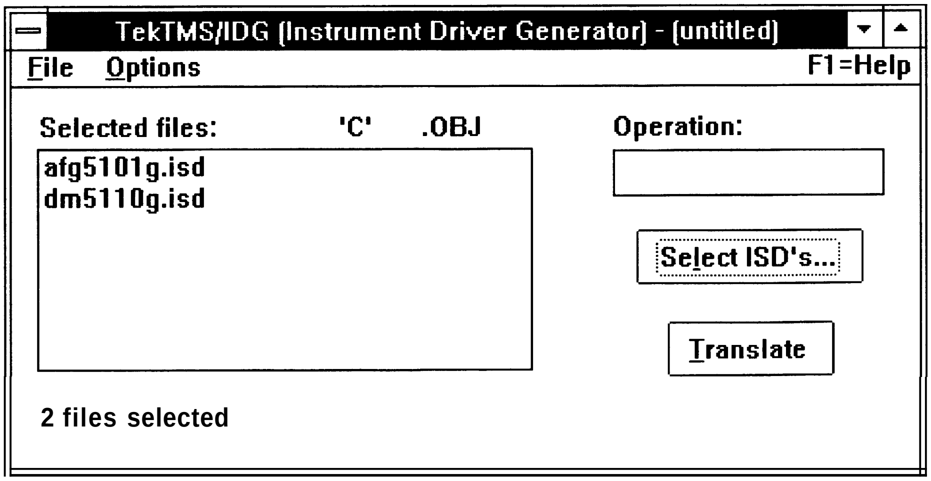
Files in: c:\tektms

funct.isd
functl .isd
funct2.isd
instdrvl .isd
instdrv2.isd
instdrv3.isd
instdrv4.isd
tdm5120.isd

Selected ISD's:

tdm5120.isd
tekfg.isd

6. Select **OK.** An Application window with the files listed in the **Selected files** status box similar to the following should appear.



Selected files:

afg5101g.isd
dm5110g.isd

2 files selected

7. Click on Translate. The screen displays that occur during the translation process are nearly identical to those described in the Translating an Instrument Driver topic earlier, except as follows:

   a. The root name for the .MAK file will be the same as the .LIB file.

   b. The translator creates a .LRF (Library Response File) after translating the files and before starting NMAKE.EXE. The root name of the .LRF file will be the same as the .LIB file.

   c. After all files are compiled, NMAKE.EXE starts the LIB Utility to build the library. The .LRF file provides input to the LIB Utility for building the library.

8. When the translation process finishes without errors, the Operation status box displays the word Complete and the NMAKE window automatically closes. If an error occurs during the translate process, an error message appears. Then the translator asks you if you want to continue or stop the process. If you continue, the process finishes the translate process and starts the compile and link process. If an error occurs during the compile or link process, the error is displayed and logged, the process stops, and the translator starts Microsoft® NOTEPAD and displays the .TXT error file.

9. Exit the translator.

Updating an Instrument Driver Library

Before starting this procedure, create any new .ISD files and update any original .ISD files. Also, if you are going to delete any .ISD files from the library, you need to delete the .LIB file to force the translator to rebuild the library.

To update an instrument driver library:

1. From the Application window, click on the **File** command. The File menu should appear.

2. Click on the **Open** command. The Open CFG File dialog box should appear.

3. Double-click on the Configuration file name used to create the original library. The Application window should appear.

4. If you are going to replace the old library, go to step 5; otherwise, continue to keep the old library and create a new one. Click on the **File** command. The File menu should appear. Click on the Save As**...** command. The Save CFG File dialog box should appear with the opened Configuration file name in the **Config** file: edit box. Enter a new name in the **Config** file: edit box and click on Save. The Applications window should appear. Click on the Qptions command. The Set Options dialog box should appear. Type a new library name in the Include **.OBJ** in lib: edit box. Click on **OK.** The Application window should appear.

5. Make needed changes as follows:

   a. If the change only involves updating of the original .PRG files, go to step 6.

   b. If instrument drivers need to be added or deleted from the library, click on the **Select Tests...** pushbutton. When the Translate ISD's dialog box appears, add or delete test procedures as needed, then click on **OK.** The Application window should appear.

   c. If options need to be changed, click on the Option command. When the Set Options dialog box appears make any changes needed, then click on **OK.** The Application window should appear.

6. Click on the **Translate** pushbutton. During the translation process, the **Operation:** status box and NMAKE window displays are the same as when Creating an Instrument Driver Library, except as follows:

   a. Only the added files and those needing updated are translated and compiled.

   b. The translator builds new .MAK and .LRF files.

   c. A new or updated .LIB file is created.

7 When the translation process finishes without errors, the Operation status box displays the words 'Complete' and the NMAKE window automatically closes. If an error occurs during the translate process, an error message appears. Then the translator asks you if you want to continue or stop the process. If you continue, the process finishes the translate process and starts the compile and link process. If an error occurs during the compile or link process, the error is displayed and logged, the process stops, and the translator starts Microsoft® NOTEPAD and displays the .TXT error file.

8. Exit the translator.

Section 6
Running a DOS Executable File

Running a DOS Executable File

This section describes the general procedures for running a compiled RTG program in DOS. It explains how to run a DOS executable file in a Windows DOS shell and the configuration changes that must be made. It shows some example displays. It lists and describes the TEKRUN.INI file installed by TekTMS/RTG to initialize the DOS execution environment.

Running the Executable File in DOS

The general procedures for running the executable file in DOS are:

1. Setup the test equipment and its associated hardware and the instrument communications bus and its associated software interface needed for the test procedure.
2. From the DOS prompt, type the name of the DOS executable file and press RETURN.
3. As the screen displays appear, perform the prompt actions, then press RETURN.

NOTE

All errors that occur during execution are fatal errors. When an error does occur, the program returns to the *DOS* prompt. In some cases, an error message may appear before returning to the *DOS* prompt.

4. When the last display appears, pressing RETURN ends the program and returns the DOS prompt.

Running the Executable File in a Windows DOS Shell

There are a number of ways to enter the DOS shell from the Windows environment. Two of the more common ways are:

1. From the Program Manager window, double-click on the Main icon. When the Main window appears, double-click on the DOS icon. You should now be in the DOS shell with a full screen display. At the DOS prompt, type the program file name and press the ENTER key. The screen displays that occur while the program is running are the same as when running in DOS. When the program finishes running and the DOS prompt appears, you can exit the DOS shell by typing the word 'exit' and pressing the ENTER key. You should then be returned to the Windows environment at the Main window.
2. From the Program Manager window, double-click on the Main icon. When the Main window appears, double-click on the File Manager icon. When the Directory Tree dialog box appears, double-click on each directory and subdirectory until a file list with the program appears. Then double-click on the program file to start it. Before the program starts, the screen changes to a full screen DOS shell. The screen displays that occur while the program is running are the same as when running in DOS. When the program finishes and the DOS prompt appears, you can exit the DOS shell by typing the word 'exit' and pressing the ENTER key. You should then be returned to the Windows environment at the Directory Tree window.

Controller Configuration When Running in a Windows DOS Shell

When the DOS executable file is executed in a Windows DOS shell to control GPIB or VXI instruments, there are controller configuration considerations that must be observed. All of the configuration considerations assume that the controller is initially set up to talk to instruments as if they were running the TekTMS/IPG version of the test procedure.

Controller GPIB Configuration Considerations — For GPIB communications, the controller must load GPIB.COM.

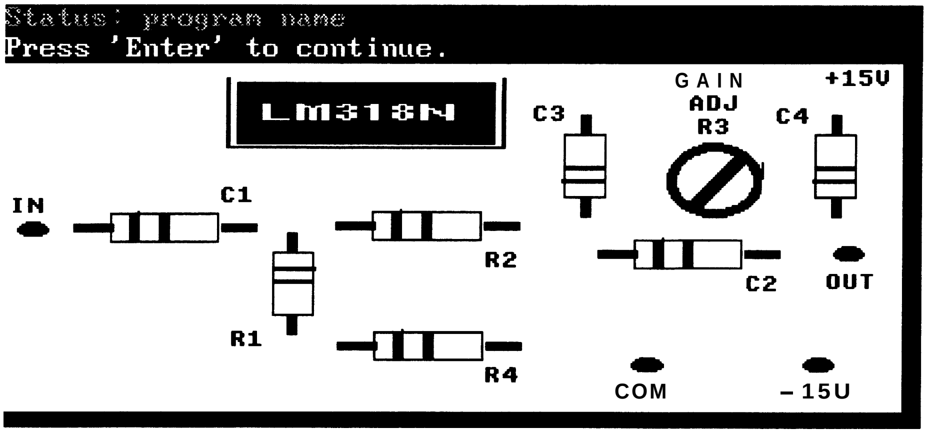
Do not run tests that use GPIB instrumentation in a Windows DOS shell Doing so will produce unpredictable results because GPIB.COM is not designed to run in the Windows DOS environment.

Controller VXI Configuration Considerations — For VXI communications, the Tektronix Imbedded controller may operate in its normal Windows operating mode.

Sample DOS Executable File Displays

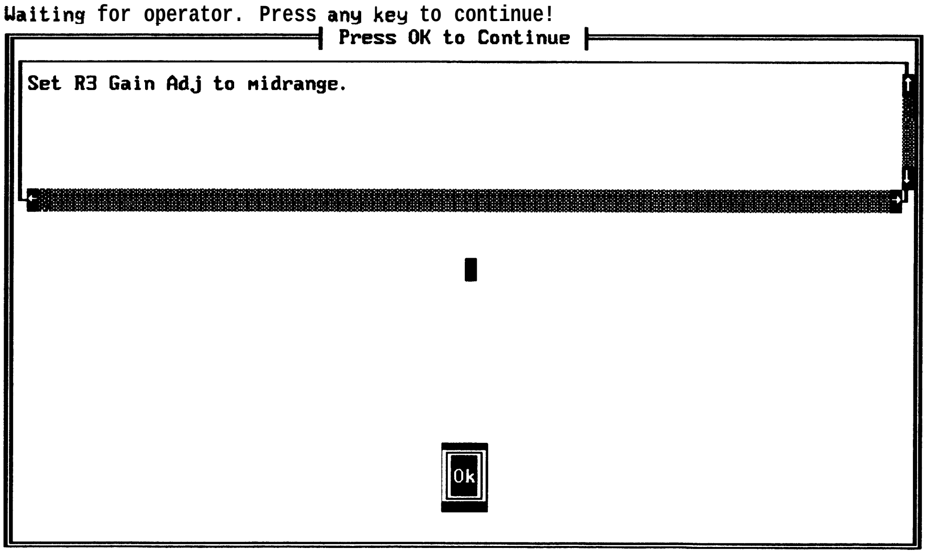
Following are some sample displays to show you the type of displays that appear when running the DOS executable file.

The following display is generated from a Picture Prompt step.



Status: program name
Press 'Enter' to continue.

The following display is generated from an Operator Prompt step.

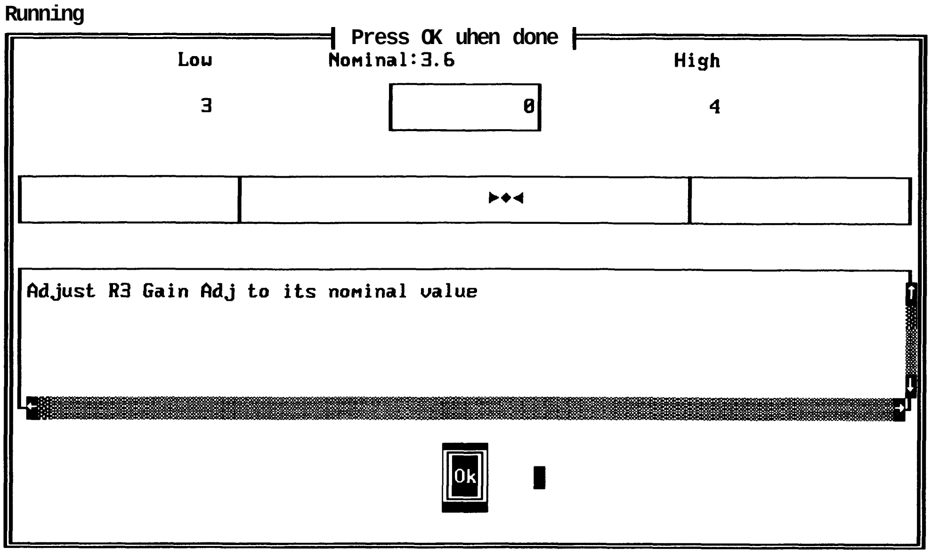


Waiting for operator. Press any key to continue!

Press OK to Continue

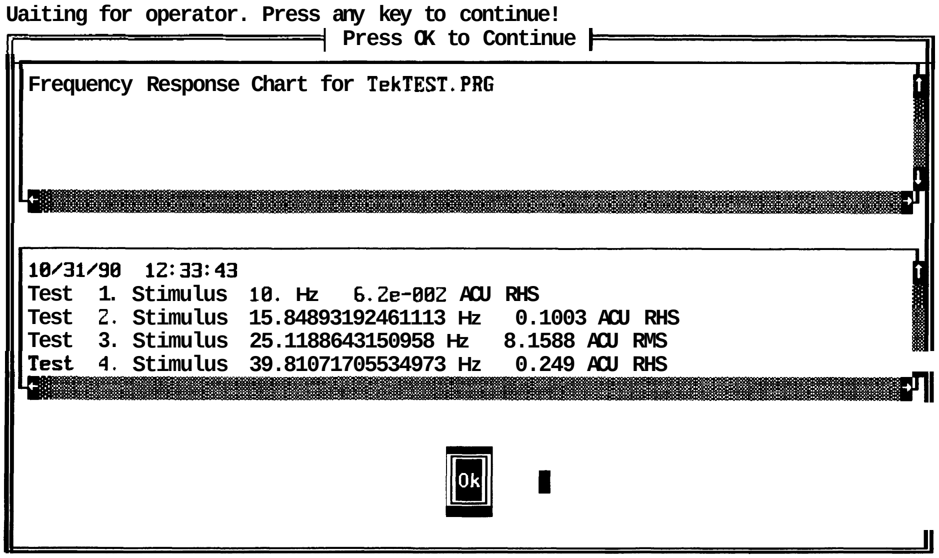
Set R3 Gain Adj to midrange.

The following display is generated from an Adjustment step.



```
Running
                    Press OK uhen done
          Low          Nominal:3.6                  High

            3                     0                   4

                                ▸◆◂

Adjust R3 Gain Adj to its nominal value

                       Ok
```

The following display is generated from a Data Display step.



```
Uaiting for operator. Press any key to continue!
                    Press OK to Continue
Frequency Response Chart for TekTEST.PRG

10/31/90  12:33:43
Test  1. Stimulus  10. Hz   6.2e-002 ACU  RHS
Test  2. Stimulus  15.84893192461113 Hz   0.1003 ACU  RHS
Test  3. Stimulus  25.1188643150958 Hz   8.1588 ACU  RMS
Test  4. Stimulus  39.81071705534973 Hz   0.249 ACU  RHS

                       Ok
```

Differences in Display Colors for Windows and DOS

On a color monitor, the screen colors when running the TekTMS/IPG test procedure in Windows may be different than when running the TekTMS/RTG program test procedure in DOS. In Windows, the screen colors are set though the Windows Color icon on the Control Panel. In the DOS environment, the screen display colors are set through the TEKRUN.INI file described later in this section. For information about setting Windows colors, see the Control Panel chapter in the Microsoft® Windows User's Guide.

TEKRUN.INI File

During the initial installation process, TekTMS/RTG installs a TEKRUN.INI file for initializing the DOS runtime environment. When TekTMS/RTG creates the DOS executable file, it includes a call to this file. TEKRUN.INI specifies the type of graphics adapter and the screen display colors for the user interface displays. Users may edit this file as desired. If for some reason the TEKRUN.INI file gets deleted, or TekTMS/RTG can't find it, the colors are set to the defaults shown in the TEKRUN.INI file below by TekTMS/RTG.

Each line in the TEKRUN.INI file is either a comment or an initialization entry. Comment lines begin with a semicolon (;) in the first column. Blank lines make the file readable and the text program ignores them during initialization.

Following is a sample TEKRUN.INI file —

```
;This is the initialization file for the TekTMS compiled
;code environment. Comments always begin with a semicolon
;in column one. If an entry does not have a value
;specified, TekTMS assigns a default value. The default
;values are those shown in this sample.

ADAPTER EGA

;ADAPTER is a keyword; EGA is the argument. The supported
;graphics adapters are EGA, VGA, or CGA.

STATUS              BLACK, CYAN
STATUS-WAIT         RED, CYAN

WINDOW              WHITE, BLACK
WINDOW-BORDER       WHITE, BLACK
WINDOW-SELECTED     WHITE, LTGRAY

ERROR               RED, LTGRAY
ERROR-BORDER        RED, LTGRAY
ERROR-SELECTED      LTRED, LTGRAY

SCROLLBAR           BLACK, LTGRAY
SCROLLELEVATOR      BLACK, LTGRAY

;All color selections are FOREGROUND, BACKGROUND pairs.
;The supported foreground colors are:

;BLACK      GREEN       LTMAGENTA   WHITE
;BLUE       LTBLUE      LTRED       YELLOW
;BROWN      LTGRAY      MAGENTA
;CYAN       LTGREEN     RED

;The supported background colors are:

;BLACK      BROWN       GREEN       MAGENTA
;BLUE       CYAN        LTGRAY      RED
```

Screen Display Area Definitions

The screen display area definitions are:

STATUS and **STATUS–WAIT** set the colors for the two lines of status text above the window. **STATUS** colors display while the DOS executable file is executing. **STATUS–WAIT** colors display when the DOS executable file stops and waits for an operator action or response.

WINDOW sets the colors for the unselected interior of the main window.

WINDOW–BORDER sets the colors for the single- or double-line border of the main window.

WINDOW–SELECTED sets the colors for the window control with the focus.

ERROR sets the colors for the interior of the error window.

ERROR–BORDER sets the colors for the single- or double-line border of the error window.

ERROR–SELECTED sets the colors for the error window control with the focus.

SCROLLBAR sets the colors for the horizontal and vertical scroll bars.

SCROLLELEVATOR sets the colors for that part of the scroll bar that moves.

Appendices

Appendix A: Menus

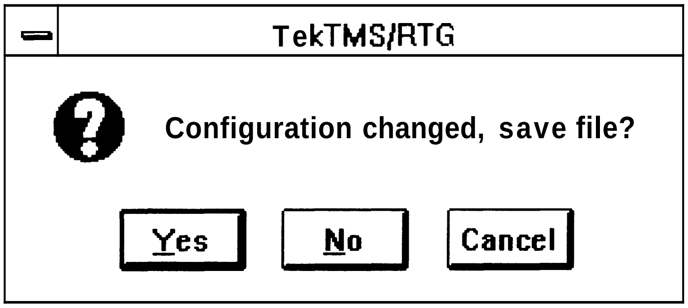
Common Message Dialog Boxes

TekTMS/RTG uses common message dialog boxes for messages to the user during a translation session.

Configuration Changed Message Box

The following message box appears under these conditions:

1. When the user resets the translator with the **New** command from the File menu and the file selection or option configuration has changed.
2. When exiting a translation session where the file selection or option configuration changed.
3. If TekTMS/RTG is closed as an icon.



Configuration changed, save file?

Yes is a pushbutton that saves the current file and option selections into a Configuration file as follows. If the translation session is untitled (shown in the translator's Application window Title Bar), the translator executes the Save command under the File menu. If a Configuration file is open, the translator updates the file. If the session is new and a target (.EXE or .LIB) is named, the translator saves the Configuration file using the root name of the target.

No is a pushbutton that exits the translation session without saving the current file and option selections.

Cancel is a pushbutton that cancels the New command or exiting action and returns to the Application window.

Test Procedure Translator Menus

Applications Window Map

Figure A-1 is a map of the Applications window for the RTG.EXE Test Procedure Translator.

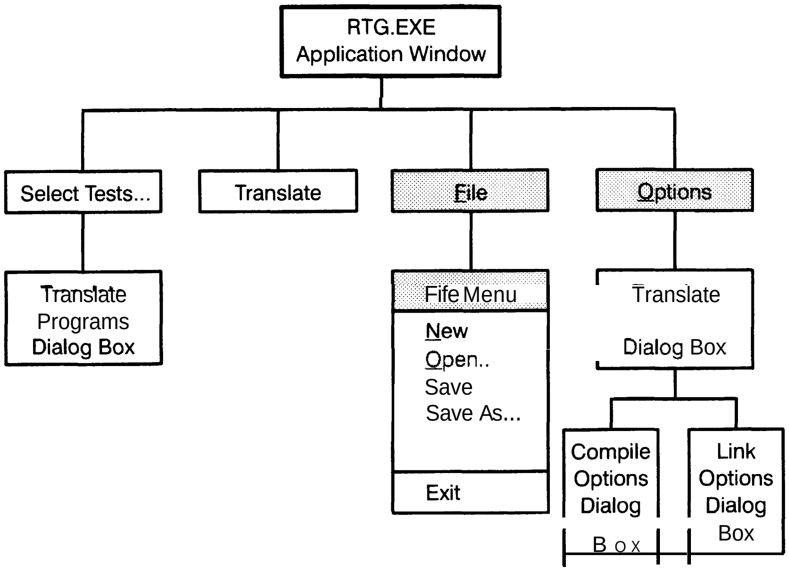
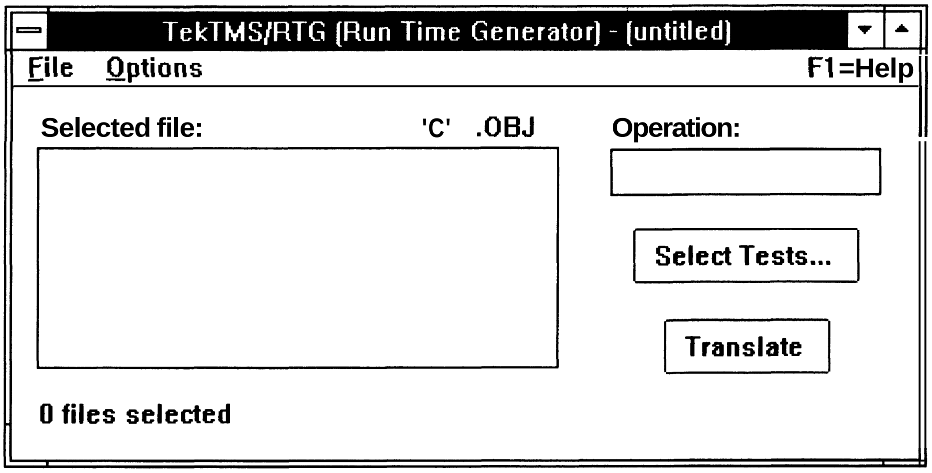


Figure A-1: RTG.EXE Translator Menu Map

The unhighlighted items across the top of the map are pushbuttons in the Application window and the highlighted items are main menu commands. Items in the File menu are commands. Notice that commands have an underlined keyboard shortcut letter. Commands and pushbuttons followed by three dots open a dialog box that must be completed or cancelled before further action occurs.

Application Window

When the Test Program Translator (RTG.EXE) starts, the following Application window appears.



When a target (.EXE or .LIB) is up-to-date, the Application window displays the words '.EXE is current' or '.LIB is current' in its lower right corner. The display in the lower left corner of the Application window indicates how many files are selected. If there are more files than can be displayed in the status box, a vertical scroll bar appears in the status box.

Select Tests... is a pushbutton that opens the Translate Programs Dialog Box to select files for the translation process.

Translate is a pushbutton to start the translation process. It is grayed (disabled) until a test procedure is selected. See Section 3, The Translation Process for actions that occur during the translation process.

Selected file: is an output only status box that lists file names selected for translation and shows the current modification status of the selected files. The C and .OBJ columns show the modification status of each file. An '*X*' in a column indicates the file is up-to-date. When all files are Xed in both columns, the target (.EXE or .LIB) is up-to-date. When a column isn't Xed, the translation process needs to be run. When the Application window title bar shows (untitled) and no files are selected, this status box and its columns are empty. After files are selected, their names appear in the status box, but the C and .OBJ columns are empty. As each selected file finishes translation, an *X* appears in the C column. After all files are compiled, the .OBJ column is Xed. If the target (.EXE or .LIB) is current when a Configuration file opens, both columns contain x's for all files. You can quickly see when files are not up-to-date. When the files are not up-to-date, the listing shows which files need processing.

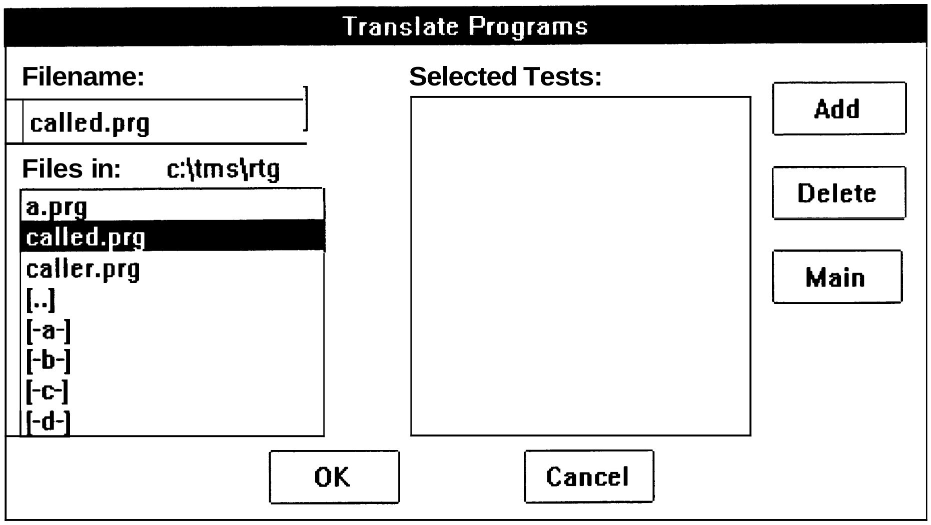
Operation: is an output only status box to display the words 'Translating', 'Compiling', or 'Complete' to show the current status of the process.

File — Pulls down the File menu.

Options — Opens the Translate Options dialog box.

Translate Programs Dialog Box

The **Select Tests...** pushbutton from the Applications window opens the following Translate Programs dialog box.



Filename: is an edit box for entering a file name. To add the file name to the **Selected Tests:** list box, select the Add pushbutton. Multiple files may be entered and added to the **Selected Tests:.**

Files in: is a text string showing the current directory path. Below the text string is a list of available test procedure files. To select files from the list, double-click the file name. Multiple files may be selected.

Selected Tests: is a list box of file names selected for translation. Unless otherwise changed with the **Main** pushbutton, the first name selected assumes a **(Main)** designation.

Add is a pushbutton that adds a name entered in **Filename:** or highlighted in the file list to the **Selected Tests:** list box.

Delete is a pushbutton that deletes a highlighted file name in the **Selected Tests:** list box.

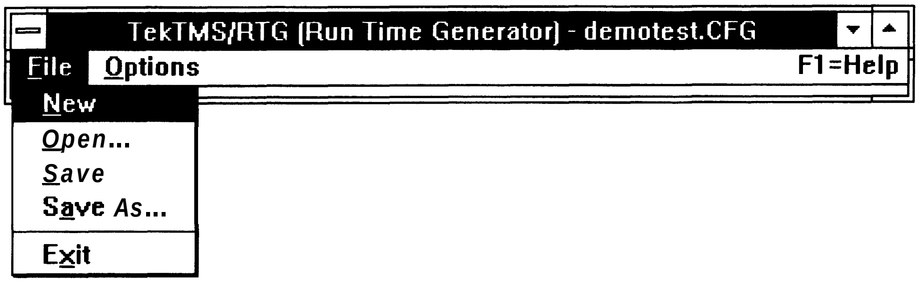
Main is a pushbutton to designate or change the main file in the **Selected Tests:** list box. It also turns off the main designation for building libraries. A Main designation only appears if the target is a .EXE file.

The first file name added to the **Selected Tests:** list box assumes the **(Main)** designation. To change the main designation, highlight any file name in the **Selected Tests:** list box and click the **Main** pushbutton. The new main file name moves to the top of the **Selected Tests:** list box and assumes the **(Main)** designation. To turn off the Main designation, highlight its file name and select the **Main** pushbutton.

OK, Cancel are pushbuttons that implement or cancel the dialog box actions.

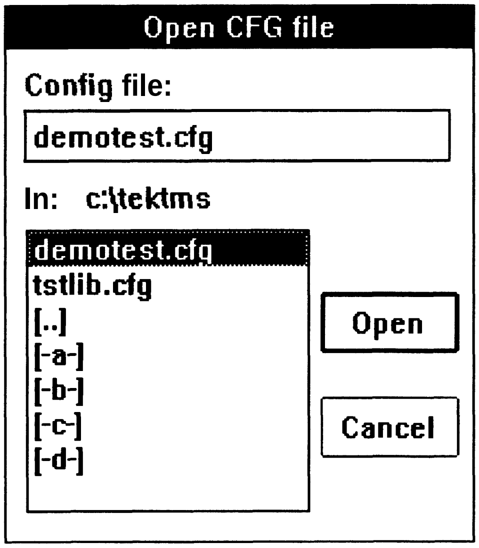
File Menu

The **File** command from the Applications window pulls-down the following File menu.



New — Initializes the Test Procedure Translator (RTG.EXE) by clearing the file selections and setting the default options. If a Configuration file is open, or the file and option selections changed during the session, a Configuration Changed Message Box appears before the initialization process occurs.

Open... — Opens the following dialog box for opening a Configuration file. If a Configuration file is open, or the file and option selections changed during the session, a Configuration Changed Message Box appears for action before the dialog box appears.



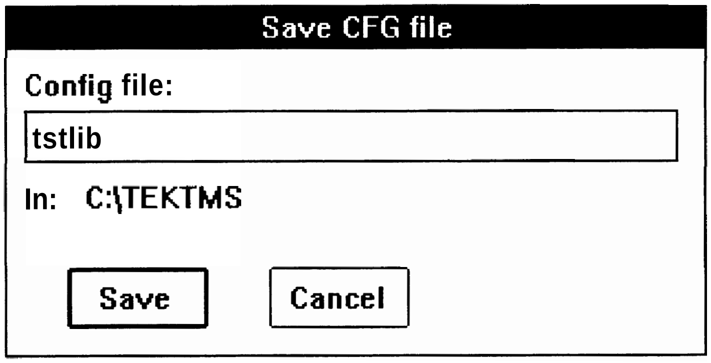
Config file: — is an edit box for entering the name of a Configuration file. To open the file, click the **Open** pushbutton.

In: — is a text string showing the current directory path. Below the text string is a list of Configuration files (.CFG files) for opening. To open a file, double-click its name, or select a name and select the **Open** pushbutton.

Open, Cancel are pushbuttons that implement or cancel the dialog box action.

Save — Saves the current file and option selection into a Configuration file named as follows:

1. If the translation session is new and a target (.EXE or .LIB) is selected, the Configuration file is saved using the root name of the .EXE or .LIB file with a .CFG extension. A .EXE target is named by the selection of a Main .PRG file. A .LIB target is named when a name is entered in **Include .OBJ in library** edit box in the Compile Options dialog box.
2. If a Configuration file is open, clicking on the **Save** command updates the file.
3. If the session is untitled (shown in the translator's Application window Title Bar), the following Save CFG File dialog box appears for naming the file.



Config file: tstlib

In: C:\TEKTMS

Config file: is an edit box for entering the path and file name for the Configuration file. After naming the file, select the **Save** pushbutton.

In: is a text string showing the current directory path where the file in **Config file:** is saved or resides (when a previously saved file is open).

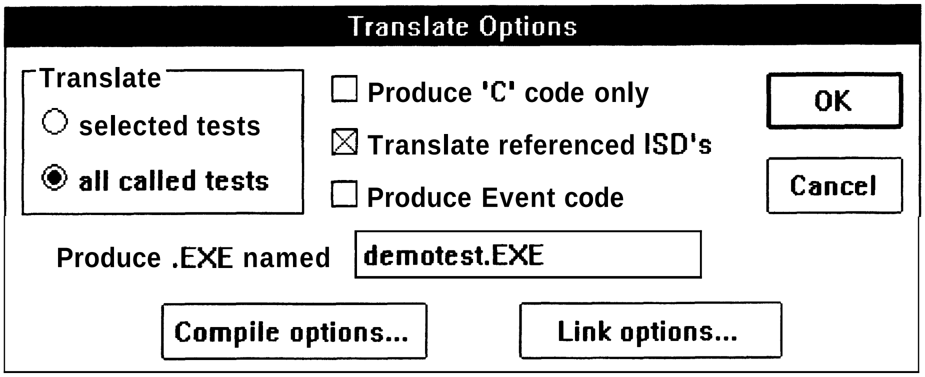
Save, Cancel are pushbuttons that implement or cancel the dialog box action.

Save As... Opens the Save CFG File dialog box above for renaming the Configuration file. If there is an open Configuration file, its name appears in the edit box. When the open Configuration file name appears in the edit box, it must be specifically changed to save a new version.

Exit Ends the current test procedure translation session and closes the Application window. If a Configuration file is open, or the file and option selections changed during the session, a Configuration Changed Message Box appears before exiting the session.

Translate Options Dialog Box

The **Options** command from the Applications window opens the following Translate Options Dialog Box shown with default options.



Translate Options

Translate

○ selected tests

● all called tests

☐ Produce 'C' code only

☒ Translate referenced ISD's

☐ Produce Event code

Produce .EXE named demotest.EXE

Selecting the **New** command from the File menu sets the options to the defaults.

Selecting a file and designating it as **(Main)** in the Translate Program dialog box:

- turns on the **Produce .EXE named** check box,
- places a file name in the **Produce .EXE named** edit box consisting of the **(Main)** file name with a .EXE extension, and
- enables the **Link options...** pushbutton.

Opening a Configuration file sets the options to those saved in the file.

Translate is a group of mutually exclusive radiobuttons that determine which files are translated.

selected tests causes translation of only the selected test procedure files.

all called tests causes translation of all selected test procedure files plus all called test procedures.

Produce .EXE named: is a check box that causes the translator to complete an entire translation process to produce a DOS-executable file. Selecting this check box causes a default DOS-executable file name to appear in the edit box using the name of the Main test procedure file with a .EXE extension. If the selected files list doesn't include a Main test procedure file, selecting this check box causes a warning message to appear.

Produce C code only is a check box that causes the translation process to stop after translating the selected files into C code. Checking this box grays the **Compiler options...** and **Link options...** PushButtons.

Translate referenced ISD's is a check box that causes all referenced instrument Drivers (.ISD files) to be included in the translation process. The checked or unchecked status of this check box determines the contents of the Make file. When checked, all referenced .ISD files are automatically translated, complied, and included in the .OBJ section of the Link command by the translator. When unchecked, users must link the .ISDs from a library, or add each .OBJ file to the **Selected files:** list box in the Link Options dialog box.

Produce Event code is a check box that includes the TekTMS On Event and On Abort handlers in the translation process. When these event handlers are not included, the DOS-executable program may take an undesired default action when it encounters an event step without finding a matching event handler.

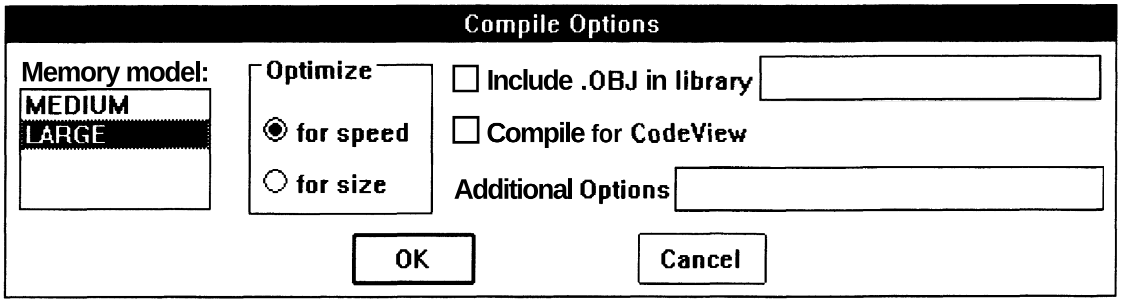
If the test procedure contains any On Event or On Abort steps from the Actions Menu in the *TekTMS/IPG User Manual,* this option will be selected when the first On Abort or On Event Step is encountered. If you want to do Control C or Control Break processing throughout the program, set this option before translation begins. If you do not have Event or Abort handlers and do not wish to do Control C processing, do not select this option because it will make the compiled test larger.

Compile options... is a pushbutton that pulls down the **Compile Options** dialog box.

Link options... is a pushbutton that pulls down the Link Options dialog box.

Compile Options Dialog Box

The **Compile options...** pushbutton from the Translate Options dialog box opens the following Compile Options dialog box.



Compile Options

Memory model: MEDIUM LARGE

Optimize
● for speed
○ for size

□ Include .OBJ in library

□ Compile for CodeView

Additional Options

Memory model: is a list box for selecting a standard compiler memory size. Each size limits the code and data segments of a program to a maximum number of 64 Kbyte segments as shown in the table below. To select a memory size, select the size name. The Medium selection produces faster code but does not provide adequate memory for calculations or large numbers of instruments. Out of memory link errors indicates a need for the Large memory model selection.

Model Size	Allowed Number of 64 K Segments	
(Compiler Option)	Code Size	Data Size
Medium (-AM)	More than one	One
Large (-AL)	More than one	More than one

Optimize for: is a set of mutually exclusive radiobuttons to select C compiler optimization options. **Speed** (C Compiler option -Ot) makes the compiled code larger in size, but faster in execution speed. **Size** (C Compiler option -0s) makes the compiled code smaller in size, but slower in execution speed. For more information, refer to the Microsoft® C manuals.

Include .OBJ in library: is a check box that causes compiled test procedures to be included in a linkable library named in the edit box. The file name can be any legal DOS file name; the translator assigns a .LIB extension. Selecting this option grays the **Produce .EXE named** option in the Translate Options dialog box.

Compile for CodeView is a check box that adds options for the compiler CodeView debugger. It changes the optimize switch to-Od (disable optimization), and adds the -Ai option (put debug information in object file) to the compiler, and the -CO option (incorporate addresses for symbols and source lines into the executable file) to the linker. For more information, refer to your Microsoft compiler manuals.

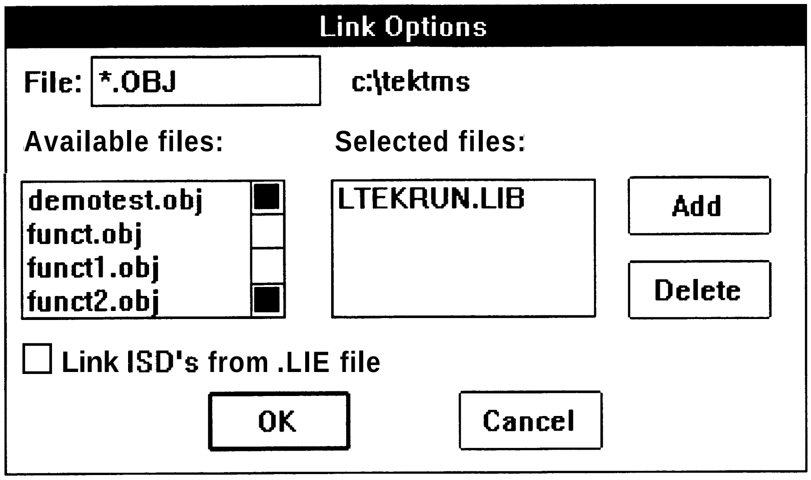
Additional Options: is an edit box for entering other C compiler options. The edit box accepts up to 128 characters including spaces. An entry may look similar to (arguments are separated by one or more spaces):

```
-c -w3 -u ...
```

For more information, refer to the Microsoft compiler manuals.

Link Options Dialog Box

The **Link options...** pushbutton from the Translate Options dialog box opens the following Link Options dialog box.



File: *.OBJ c:\tektms

Available files:
demotest.obj ■
funct.obj □
funct1.obj □
funct2.obj ■

Selected files:
LTEKRUN.LIB

□ Link ISD's from .LIE file

File: is an edit box for entering a .OBJ or .LIB file name for linking. Click on **Add** to add the file name to the **Selected files:** list box. Multiple files may be selected. A string to the right of the **File:** edit box shows the current directory.

Available files: is a list box containing the selected file names.

NOTE

> The translator automatically adds a *TekTMS* Runtime *Library* file such as *xTEKRUN.LIB,* where x is a letter such as *M,* L, etc. denoting a relationship to a memory model size, to the **Selected Tests:** list box.

Link ISD's from .LIB file is a check box to indicate to the translator that compiled Instrument Drivers are included in a library file (.LIB file). User created libraries must be added to the **Selected files:** list box when this option is checked.

Add is a pushbutton for adding the **File:** name to the **Selected files:** list box.

Delete is **a** pushbutton for deleting a file name highlighted in the **Selected files:** list.

OK, Cancel are pushbuttons that must be selected to implement or cancel the dialog box actions.

Instrument Driver Translator Menus

Application Window Map

Figure A-2 is a map of the Applications window for the IDG.EXE Instrument Driver Translator.

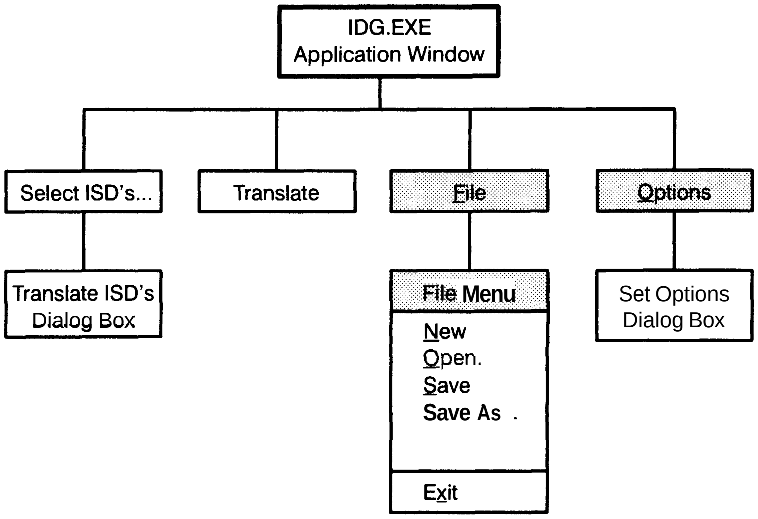
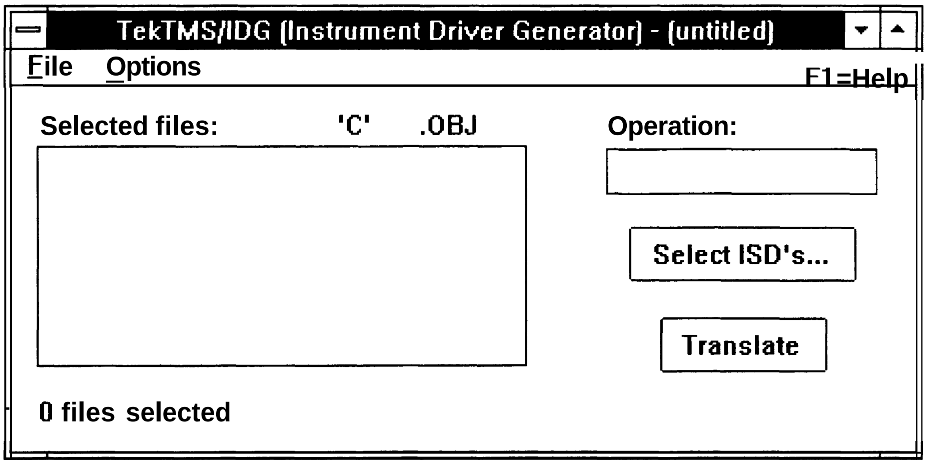


Figure A-2: IDG.EXE Translator Menu Map

The unhighlighted items across the top of the map are pushbuttons in the Application window and the highlighted items are commands. The items in the File menu are commands. Notice that commands have an underlined keyboard shortcut letter. Commands and pushbuttons followed by three dots open a dialog box that must be completed or cancelled before further action occurs.

Application Window

When the Instrument Driver Translator (IDG.EXE) starts, the following Application window appears.



Select ISD’s... is a pushbutton that opens the Translate ISD’s dialog box.

Translate is a pushbutton that starts the Instrument Driver translation process. The actions that occur during the translation process are explained in more detail in Section *4,* Using the Instrument Driver Translator.

Selected files: is an output only status box. It lists the .ISD file names selected for translation and shows the current chronological status (date/ time) of the files.

The C and .OBJ columns show the translation status of target files in the translation process. An X in a column indicates the file is up-to-date. When both columns are Xed for all selected files, the compiled files are all current. When a column doesn't contain an X, the translation process needs to be run.

When the Application window title bar shows (untitled) and no test programs are selected, this status box and its columns are empty. When files are first selected, their names appear in the box, but the C and .OBJ columns are empty. As each selected file finishes translation, an X appears in the C column. The .OBJ column is Xed after all files finish compilation.

If all selected files are current when a Configuration file opens, the C and .OBJ columns are Xed for all files.

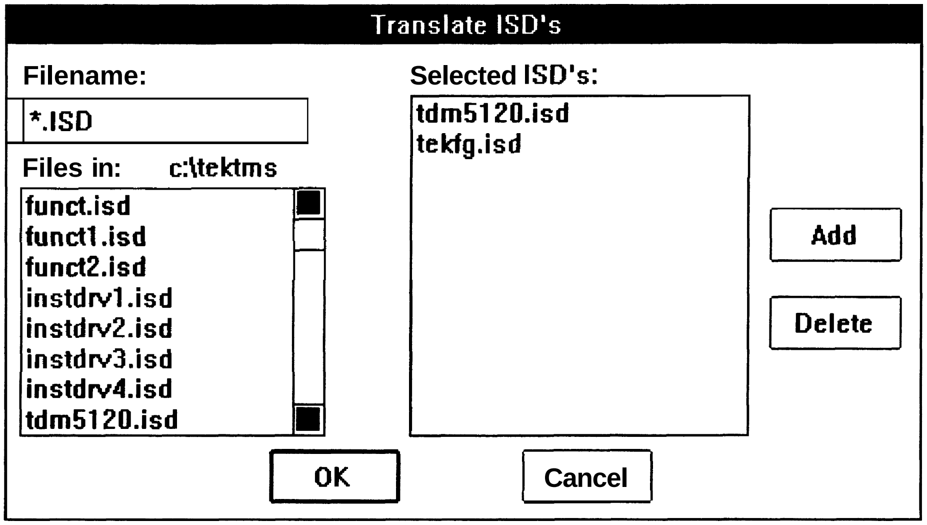
Operation: is an output only status box. It displays the words 'Translating', 'Compiling', or 'Complete' to show the current status of the translation process.

File Pulls-down the File menu.

Options Opens the Set Options dialog box.

Translate ISD's Dialog Box

The **Select ISD's...** pushbutton from the Applications window opens the following Translate ISD's dialog box.


Translate ISD's

Filename:

*.ISD

Files in: c:\tektms

funct.isd
funct1.isd
funct2.isd
instdrv1.isd
instdrv2.isd
instdrv3.isd
instdrv4.isd
tdm5120.isd

Selected ISD's:

tdm5120.isd
tekfg.isd

Filename: is an edit box for entering an instrument driver file name. To add the file name to the **Selected ISD's:** list, select the **Add** pushbutton. Multiple files may be entered and added to the **Selected ISD's:** list.

Files in: is a text string showing the current directory. Below the text string is a list of available instrument driver files list to select. To select files from the list, double-click the file name. When selected, the name appears simultaneously in **Filename:** and **Selected ISD's:.**

Selected ISD's: is a list box of instrument driver file names selected for translation.

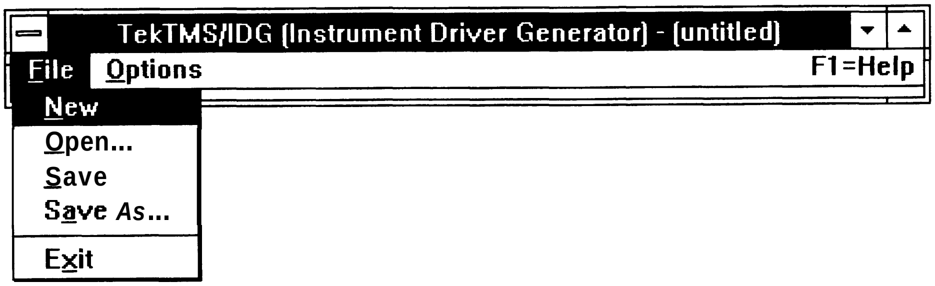
Add is a pushbutton that moves the file in **Filename:** to the **Selected ISD's:** file list.

Delete is a pushbutton to delete a highlighted file name in the **Selected ISD's:** list.

OK, Cancel are pushbuttons that implement or cancel the dialog box actions.

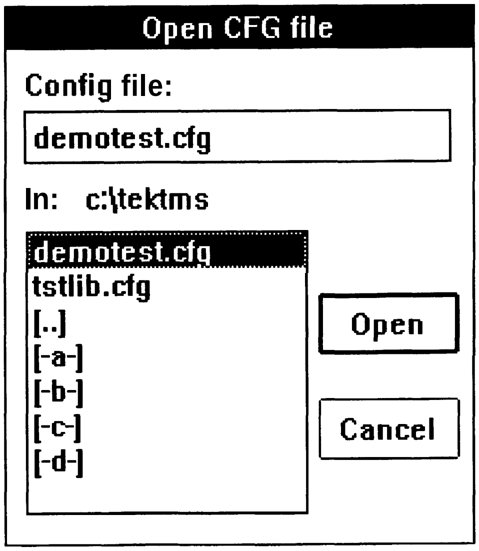
File Menu

The **File** command from the Application window pulls-down the following File menu.



New Initializes the Instrument Driver Translator (IDG.EXE) and sets default. If a Configuration file is open, or the default configuration changed, a Changed Configuration Message Box appears to give the user an opportunity to save the configuration.

Open... Opens the following dialog box for opening a Configuration file. Double-click on a file name to open it. If a Configuration file is open, or the default configuration changed, a Changed Configuration Message Box appears to give the user an opportunity to save the configuration.



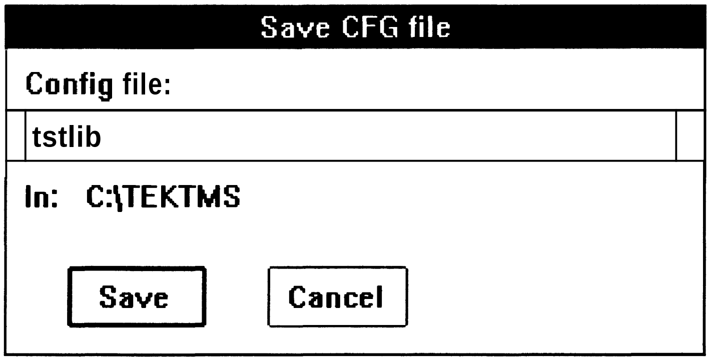
Config file: is an edit box for entering a Configuration File name. To open the file, select **Open.**

In: is **a** text string indicating the current directory. Below the text string is a file list. To open a file, double-click on its name in the file list.

Open, Cancel are pushbuttons that implement or cancel the dialog box actions.

Save Saves the current file and option selection into a Configuration file named as follows:

1. If the translation session is new and a target (.LIB) is selected, the Configuration File is saved using the root name of the .LIB file with a .CFG extension. A .LIB target is named when a name is entered in **Include .OBJ in library** edit box in the Compile Options dialog box.
2. If a Configuration file is open, clicking on the **Save** command updates the file.
3. If the session is untitled (as shown in the translator's Application window Title Bar), the following Save CFG File dialog box appears for naming the file.



Config file:
tstlib

In: C:\TEKTMS

Config file: is an edit box for entering the path and file name for the Configuration file. After naming the file, select the **Save** pushbutton.

In: is a text string showing the current directory path where the file in **Config file:** is saved or resides (when a previously saved file is open).

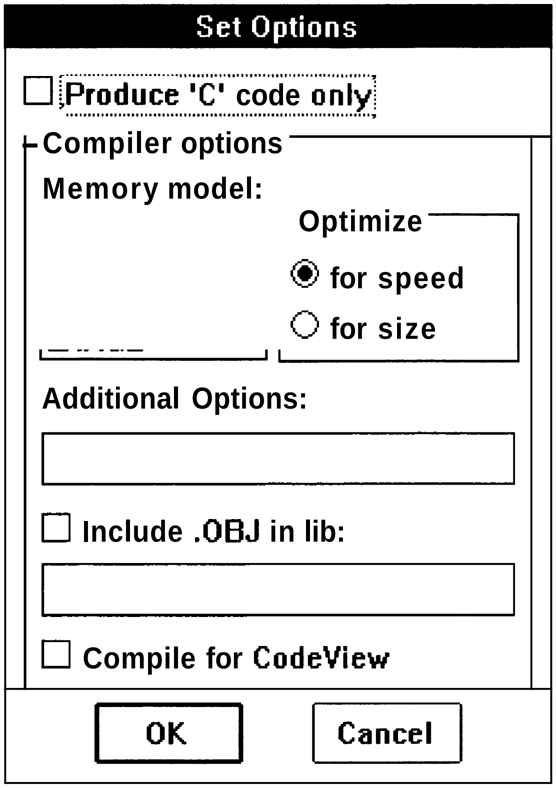
Save, Cancel are pushbuttons that implement or cancel the dialog box action.

Save As... Opens the Save CFG File dialog box above for renaming the Configuration file. If there is an open Configuration file, its name appears in the edit box. When the open Configuration file name appears in the edit box, it must be specifically changed to save a new version.

Exit Ends the current IDG.EXE translator session and closes the Application window. If a Configuration file is open, or the file and option selections changed during the session, a Configuration Changed Message Box appears before exiting the session.

Set Options Dialog Box

The **Options** command from the Applications window opens the following Set Options dialog box.



Set Options

□ Produce 'C' code only

Compiler options

Memory model:

Optimize

● for speed

○ for size

Additional Options:

□ Include .OBJ in lib:

□ Compile for CodeView

Produce 'C' Code Only is a check box that causes the translation process to stop after translating the selected files into C code. Checking this box grays the Compiler options.

Memory model: is a list box for selecting a standard compiler memory size. Each size limits the code and data segments of a program to a maximum number of 64 Kbyte segments as shown in the table below. To select a memory size, select its size name.

Model Size	**Allowed Number of 64 Kbyte Segments**	
(Compiler Option)	**Code Size**	**Data Size**
Medium (-AM)	More than one	One
Large (-AL)	More than one	More than one

Optimize for: is a set of mutually exclusive radiobuttons to select C compiler optimization options. **Speed** (C Compiler option -Ot) makes the compiled code larger in size, but faster in execution speed. **Size** (C Compiler option -0s) makes the compiled code smaller in size, but slower in execution speed. For additional information, refer to the Microsoft compiler manuals.

Additional Options: is an edit box for users to enter other C Compiler options for the translation process. This edit box accepts up to 128 characters including spaces. An entry may look similar to:

```
-c -W3 -u ...
```

where each entry is separated by one or more spaces. For additional information, refer to the Microsoft compiler manuals.

Include .OBJ in lib: is a check box that causes compiled files to be stored in the linkable library named in the edit box. The file name can be any legal DOS file name. The translator assigns a .LIB extension.

Compile for CodeView is a check box that adds options for using the C compiler's CodeView debugger. It adds the -Od option (disable optimization) and -Zi option (put debug information in object file) to the compiler. For additional information, refer to the Microsoft compiler manuals.

OK, Cancel are pushbuttons that implement or cancel the dialog box actions.

Appendix B: Disk Files and Miscellaneous Operating Information

TekTMS/RTG Program Files

For a listing of the program files refer to the READ.110 file on Disk 1 of the program disk(s).

File Names and File Name Conflicts

There are few rules to observe when assigning file names to TekTMS/RTG Configuration, Library, or Executable Files. File names can be any legal DOS file name and extension.

Because TekTMS/RTG often runs with other software, users must ensure file names do not conflict. For example, file name conflicts may occur when using the National Instruments GPIB.COM interface card software in a Tektronix System Controller or stand alone PC to control GPIB instruments. The GPIB.COM software uses file names, such as: DM5010, PS5010, etc. for GPIB devices. These names may conflict with file names assigned to ISD Configuration or Library files. If a file name conflict occurs between the software and TekTMS/RTG, the DOS executable program (.EXE) may "lock-up" or "hang" the system during runtime.

File Name Extensions

Table A-1 lists the file name extensions used or created by TekTMS/RTG.

Table A-l: TekTMS/RTG File Name Extensions

File Extension	Use
.C	Translated files
.CFG	RTG and IDG configuration files
.DAT	Data files created when running a DOS-executable file
.EXE	DOS-executable files
.H	Include files of function prototypes
.INI	Initialization files
.ISD	Instrument driver files
.LIB	Library files
.LNK	Link response file of LINK commands
.LRF	Library response file of LIB commands
.MAK	Make file of compiler commands for the NMAKE Make Utility
.MAP	Linker listing files
.110	Read Only information files
.OBJ	Compiled files
.PIF	Controls Windows execution of non-Window programs
.PRG	Test procedure files
.TMP	Temporary files created by the TekTMS/RTG translators
.TXT	Listing file of compiler/link/lib actions

.TMP Files

When the TekTMS/RTG translators are running, they may create temporary files (.TMP Files), that are automatically deleted whenever a translator session closes normally. If a translator session closes abnormally, the .TMP files may not be deleted. Thus, you need to periodically find the .TMP files and delete them to save disk space.

The DOS environmental variable TEMP determines the storage location of .TMP files. An undefined TEMP variable causes .TMP files to be stored in the root directory of the boot drive.

The DOS SET command sets the TEMP environmental variable. Its format is:

```
set temp=<pathname>
```

The following example causes .TMP files to be stored in a TEMP directory.

```
set temp=C:\tmp
```

To automatically set the TEMP environmental variable at boot-up, put the SET command line in your *Autoexec.bat* file.

NOTE

If you are using a RAM drive, such as Windows *RAMDRIVE.SYS*, putting the TEMP storage location in the RAM drive could make the *TekTMS/RTG* translation process run faster.

READ.110 File

The READ.110 file on Disk 1 of the program disks contains information about TekTMS/RTG program changes that occurred too late to be put in the manual. When the INSTALL program asks if you want to read the file, you should do so. To read the READ.I10 file at a later date, type:

```
more < a:read.110
```

where a: is the pathname to the file. The file displays one page at a time. To view the next page, press any keyboard key.

Starting the RTG.EXE or IDG.EXE Translator from DOS

To start the RTG.EXE test procedure translator and open a Configuration file from the DOS prompt, type:

```
win rtg fname.cfg
```

where fname is the name of the Configuration file.

To start the IDG.EXE instrument driver translator and open a Configuration file from the DOS prompt, type:

```
win idg fname.cfg
```

where fname is the name of the Configuration file.

Appendix C: Software Performance Report

This Software Performance Report is for your use in reporting any problems you experience when using TekTMS/IPG. It provides us with a way to track problems with a particular system and it ensures that we provide you and other customers with a prompt solution to the problem. Please supply the following information:

Customer (Company Name):______________________________

User (Person making report):______________________________

Address:______________________________

City:________________ State:____________ Zip:________

Country:______________ Telephone:________________

System Description (Product Name, S/W Version, Serial Number):__________

Product Name: __________ PC Model/Make:______________

Version:______________ DOS Version:________________

Serial No:_____________ Windows Version:______________

Other Software (TSR's, ...)

Problem Description:______________________________

____

____

____

____

(If possible, include exact steps to recreate the problem.)

Itemize attached documentation (i.e., listings, diskettes, ...)

Mail the report to:

Tektronix, Inc.
Instrument Controllers and Software Marketing Dept.
MS 47-665
P.O. Box 500
Beaverton, OR 97075-9965

Or FAX to:

(503) 627-2933

Marked for ICS Marketing, 47-665

Index

Index