CHICAGO BOOTH

CRSP®
Center for Research in Security Prices

CRSP UTILITIES & PROGRAM LIBRARIES RELEASE NOTES

Tools for CRSPAccess

105 West Adams, Suite 1700
Chicago, IL 60603
Tel: 312.263.6400
Fax: 312.263.6430
Email: Support@crsp.ChicagoBooth.edu

Updated August 7, 2019

CRSP CONTACT INFORMATION

For further information, please visit our website at www.crsp.chicagobooth.edu or email support@crsp.chicagobooth.edu.

Table of Contents

Chapter 1: CUPL 3.22 access 1

Chapter 2: Installation 4

Chapter 3: Using c with cupl 9

Chapter 4: Using FORTRAN-95 with cupl 16

CHAPTER 1: CUPL 3.22

This release of CRSPAccess Version 3.22 is also known as the CRSP Utilities and Programming Libraries (CUPL). It is intended for subscribers on Linux and Solaris platforms, and Windows subscribers who have specifically requested the command-line CRSPAccess tools.

64-Bit Support

CRSP is pleased to announce that we are now providing support on 64-bit platforms:

- Linux Redhat – 64-bit
- Solaris- Sun Ultra Sparc and on Intel x86
- Windows XP and Windows 7

Executables and files used in the 64-bit software have the same names as the 32-bit executables and files. When installed, folder names holding the for the 64-bit files are different:

32-BIT FOLDER NAMES	64-BIT FOLDER NAMES
Accbin	Accbin64
Acclib	Acclib64
Include	Include64
Sample	Sample64

Support for 32-bit platforms continues.

CRSPAccess Supported Systems

CRSP has tested programs and libraries on these supported operating systems and compilers. More recent versions of these systems and compilers or others may be compatible, but are not fully supported.

OPERATING SYSTEM	CPU	FORTRAN COMPILER*	C COMPILER	BINARY	CRSPACCESS VERSION
Windows XP	Intel x86 32-bit	Intel Fortran 9.1 and higher with Visual Studio 2005 or 2008	MS Visual Studio C++ 2005 or 2008	IEEE Little Endian	3.14 and higher
Windows 7	Intel x86 32 – or 64-bit	Intel VisualFortran 2011/ ParallelStudio XE	MS Visual Studio C++ 2008	IEEE - Little Endian	3.22
Sun Solaris 10	Sun Sparc	Sun Fortran-95 8.2	Sun C 5.8, part of SunStudio 11	IEEE – Big Endian	3.14 and higher
	Intel X86	Sun Fortran-95 8.2	Sun C 5.8, part of SunStudio 11	IEEE - Little Endian	3.22
Red Hat Enterprise Linux 5.0	Linux x86 32-bit	Lahey/Fujitsu Fortran-95 6.20 g95 0.91	gcc3.2.3	IEEE - Little Endian	3.14 and higher
	Linux x86 64-bit	g95 0.91	gcc 4.1.2	IEEE - Little Endian	3.22

*Refer to Chapter 4 for more specifics related to compilers and compatibility.

CRSPAccess Supported Versions

Official Support for CRSPAccess versions 2.97, 3.10 – 3.12 is now discontinued.

CRSPAccess Versions 3.14 and 3.19 will continue to be supported through June 2012.

CRSPACCESS (CUPL) VERSION	FORTRAN-95	C PROGRAMS	CRSP UTILITIES (TS_PRINT, STK_PRINT, CCM_PRINT, IND_PRINT)	CRSP LEGACY UTILITY SUPPORT CST_PRINT
3.14	Supported	Supported	Supported	Supported
3.19	Supported	Supported	Supported	Supported
3.22	Supported	Supported	Supported	Not Supported

Attention SASECRSP Engine Users

Compatibility between SAS versions and CRSPAccess Verion 2.97 and higher follows:

- SAS Version 9.1.3 – Service Pack 3, SAS Version 9.2, or the recently released SAS Version 9.3 is required for the SASECRSP engine to work at its best. Functionality includes access to Indexes data and to the old format Compustat (CPZ) for those who subscribe to those products
- SAS Version 9.3 includes a new SASEXCCM engine that includes support for the new format Compustat (CMZ) Databases. In this release, the SASEXCCM engine is still marked as experimental.

Programmers

Programming libraries have been compiled for support on 64-bit computers for C and FORTRAN.

CRSP continues to provide support for set-based data access through programming libraries, but encourages subscribers to transition to the item-based access that was first introduced in 2008. *CRSP Programming Guides* available on our website provide instructions for both methods.

The CRSP software includes one CRSP library for each supported language: C, F95 and G95 (for Fortran on Linux) Each library includes both set-based and item-based access.

CRSPAccess DLL is included in both 32-bit and 64-bit CRSP-supported Windows platforms.

Command Line Tools

Ts_print sample program, ts_samp8.rqt requests data from both the stock and CRSP/Compustat Merged Databases. Detailed in the June 2011 CCM release notes, beginning with the June data cut of the CCM database, keysets for Banks were changed from 2-digit numbers to 4-digit numbers. Ts_samp8.rqt reflects this keyset change. If using this sample program with a database prior to June 2011, this sample program will need to be edited to replace keyset 2100 with keyset 44.

Known Issues

Large Volumes

There are four dates where the daily trading volume for Citigroup, PERMNO 70519, Ticker C, exceed our database's maximum value (2147483648). Instead of inserting a false value into the database, CRSP has listed the volumes for these dates as -99 (missing). The true trading volume values for those dates:

DATE	VOLUME
20090805	2674463281
20091217	3772638437
20091218	2813697156
20101207	3267829406

We expect a future release of CRSPAccess to be able to handle these large values properly at which time they will replace the missing values.

CRSP Guides

All CRSP User Guides and Manuals are available on our website at: www.crsp.ChicagoBooth.edu/documentation

CHAPTER 2: INSTALLATION

The following installations are now available:

Windows DVD (Single DVD with 2 installation options):

- Setupwin32.exe
  - 32-bit installation will install on either 32- or 64-bit machines.
  - 32-bit CRSPAccess command-line utilities will work on either 32-or 64-bit machine
  - C and FORTRAN programming libraries will work only on 32-bit machine.
- Setupwin64.exe
  - 64-bit installation will install only on 64-bit machine.
  - Error message will return during the initialization phase of the installation when trying to install on 32-bit computer:

Solaris DVDs:

- Setupsolaris.bin
  - Separate DVD for setup for Solaris – Sparc
- Setupsolarisx86.bin

Separate DVD for setup on Solaris Intel x86

Linux DVD (Single DVD with 2 installation options):

- Setuplinux.bin
  - 32-bit installation will work on either 32- or 64-bit machines.
  - 32-bit CRSPAccess command-line utilities will work on either 32- or 64-bit machines
  - C and FORTRAN programming libraries will work only on 32-bit machine.
- Setuplinux64.bin
  - 64-bit installation will work only on 64-bit machine
  - Error message will return during the initialization phase of the installation when trying to install on 32-bit computer:

```
Launching installer...
./setuplinux64.bin: Line 2471: /space/temp/install.dir.4493/Linux/resource/
jre/bin/java: cannot execute binary file
./setuplinux64.bin: line 2471: /space/temp/install.dir.4493/Linux/resource/
jre/bin/java: Success
[root@localhost CUPL1_VER322_SRD]#
```

Note Regarding Data Installs

Current CRSP database installs will install properly on either 64-bit or 32-bit platforms. When uninstalling CRSP data, if using the Uninstaller, on some systems, small log files will be left behind. These are harmless and do not eat significant space.

If you encounter this and are concerned, please contact support@crsp.chicagobooth.edu or call us at 312-263-6500, Option 2.

Preparing for Installation

CRSP utilizes the InstallAnywhere© wizard-driven installation process that is used for both CRSP software and accompanying data. Files are compressed on DVD and are not directly accessible until installed.

A single DVD with CRSPAccess software and Programming Libraries (Volume label CUPL) is available to both Compustat/Merged and Stock and Indices database subscribers.

Please Note:

- *Installation over a previous version of CRSPAccess software:* CRSP strongly recommends executing one of the following two actions before installing CRSPAccess 3.22 directly into a location that contains a prior version of the software. This will insure a clean installation. Either:

  1. Uninstall the older version before installing CRSPAccess 3.22, using either the uninstall command from the CRSPAccess menu, or using Add/Remove programs through the Control Panel, or

  2. First rename the old folder containing the CRSPAccess software then install CRSPAccess 3.22 into a folder with the name you wish to use. For example, if you have CRSPAccess 2.97 on your computer in a folder named CRSP, first rename this folder to something such as CRSP297 or CRSP_old. When installing CRSPAccess 3.22, it may now be installed into a new folder named CRSP.

- *Windows Command Prompt:* InstallAnywhere bypasses the need for users to set path variables. A shortcut labeled CRSP Command Prompt is available in CRSPAccess from the start menu. To run the command line utilities, this shortcut will set the environment variables and open a window. To use the command prompt from Accessories or by running cmd.exe, you will need to manually set your path in the command window with the following:

  ```
  set path=%crsp_bin%;%path%
  ```

- *Uninstall for Windows:* To comply with recommended Windows procedures, shortcuts are no longer provided. The cleanest uninstalls are performed by going through the Control Panel > Add/Remove Programs.

- *Client Environment for Windows:* The client_environment.exe is used to set the environment variables needed to run CRSPAccess for multiple or single users. This can set variables at either the user or system level. A client_environment.exe is included in the 3.22 release of CRSPAccess. Stock or Stock & Index-only subscribers should leave the area for the CRSP\Compustat Merged Database blank. Client_environment.exe is located in the accbin folder of CRSPAccess or can be accessed from the CRSPAceess menu under Start, if installed on the local machine.

Installation steps

The following screen shots and instructions were written from the InstallAnywhere procedures for Windows systems. The installation is very similar for all supported operating systems, so these systems are all served by this one set of instructions. Differences between systems lie primarily in accessing the DVD and how the paths are defined. Where there are differences, they are clearly noted.

Windows:

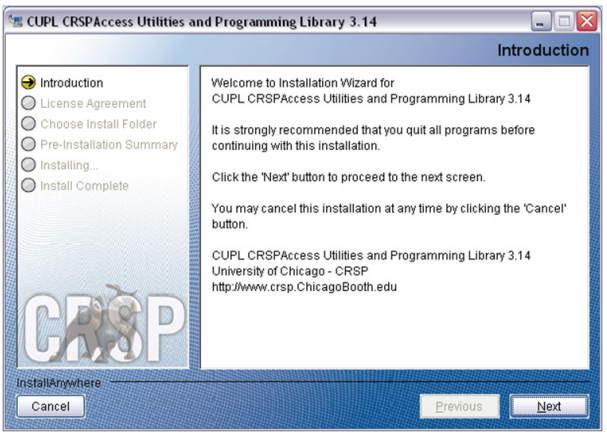
To install the CRSP Access software, insert your DVD and from Start, select **Run**. Browse the DVD and select the appropriate platform, 32- or 64-bit. Click on `setupwin*.exe`. Click **OK**. The Install Splash screen will pop up on your screen and disappear. A few moments may follow before the install process begins. Once the installation initiates, you will be guided by the InstallAnywhere Wizard.

Sun Solaris:

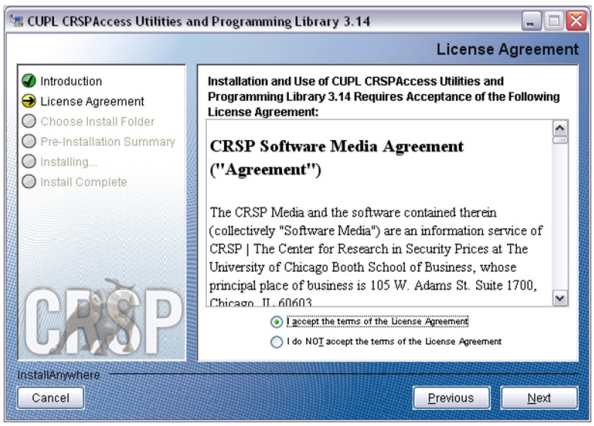
Upon insertion of the DVD, a terminal will open with the file, `setupsolaris.bin` or `setupsolarisx86.bin`. Double click on this setup file to begin the installation process.

Linux:

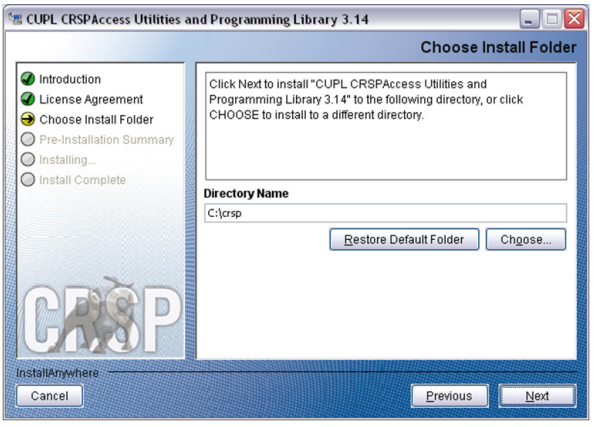
Upon insertion of the DVD, a terminal will open with the file, `setuplinux*.bin`. Double click on the appropriate setup file to begin the installation process.



After you have clicked **Next** on the Welcome screen, scroll through and read the CRSP Software Media Agreement. Click to accept the terms of the license agreement, and assuming you do, click **Next**.



The default directory for the software installation is `c:\crsp`. You may accept or modify it. We recommend creating a folder that reflects the version of the software. Click **Next**.

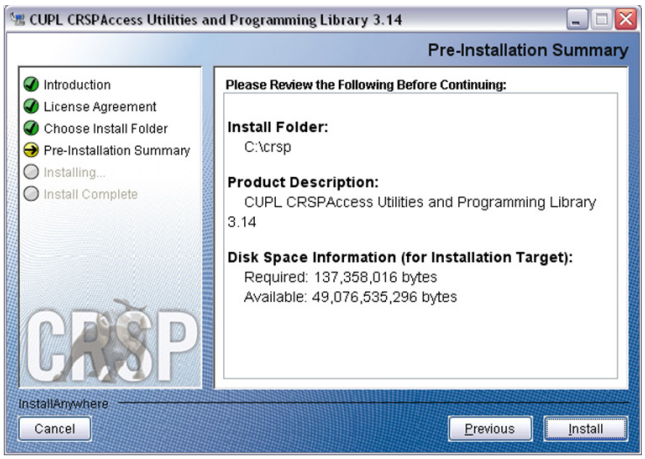


Windows:

Summary information is displayed: Location, software features, and the amount of space that will be used. Click **Install**.

Sun Solaris & Linux:

The root directory that will appear on the screen, based on the previous suggestion would read `/home/username/crsp`. Click on **Install** to proceed.

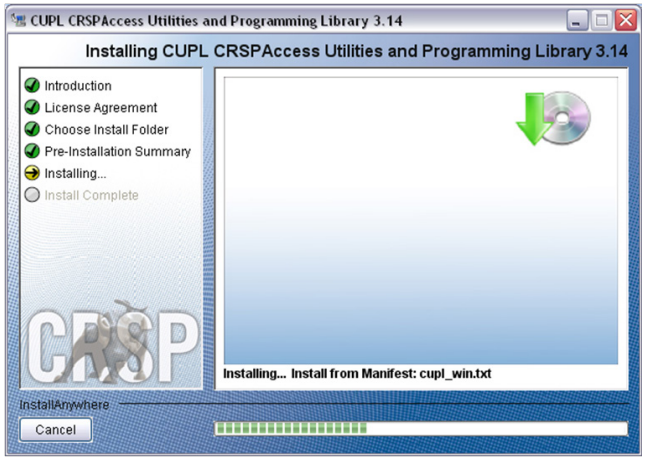


Windows:

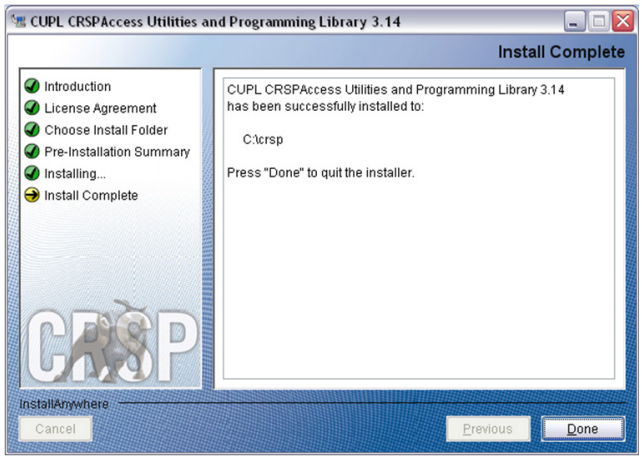
The status of your installation will appear.

Sun Solaris & Linux:

The root folder that will appear will be `/home/username/crsp/accbin`.



A message indicating the success of the installation appears when the process is complete. Click **Done**.



Sun Solaris & Linux:

After clicking **Done**, a file, crsp.kshrc can be run to set software alias names at the prompt, type:

```
>. ./crsp.kshrc <enter>
```

CRSPAccess Environment Variables

ENVIRONMENT VARIABLE	USAGE
CRSP_ROOT	Top level program directory. Most other CRSP environment variables are set based on CRSP_ROOT
CRSP_LOG	Log directory used for user
CRSP_MSTK	CRSP Monthly Database directory
CRSP_DSTK	CRSP Daily Database directory
CRSP_CCM	CRSP/Compustat Merged Database directory (if available) - Xpressfeed CCM version
CRSP_INCLUDE	Programming header files; include subfolder of root
CRSP_SAMPLE	Sample programs; sample subfolder of root
CRSP_LIB	Object libraries; acclib subfolder of root (control files)
CRSP_BIN	Executables and scripts; accbin subfolder of root
CRSP_ENV_ULOG	Usage logs produced by users; =CRSP_LOG
CRSP_ENV_ELOG	Error logs produced by users; =CRSP_LOG
CRSP_ENV_ROOT	Variable must point to the most recent CRSPAccess database installed on your system
CRSP_WORK	Directory used to store user-generated files

Linux and Sun Solaris Environment Variables

Important Change for Sun Solaris & Linux Installations:

Following previous installs, the `crsp.kshrc` file was run upon completion to set both environment variables and software alias values. This new version of InstallAnywhere sets the environment variables directly on the system when the software is installed. The `crsp.kshrc` file must still be run to set the aliases.

A user may wish to revert back to a previous cut of data or a previous version of the software. In order to do so, CRSP provides shell scripts for users to run that will create a custom-named kshrc file that the user may run to set environment variables at the session level. This process also provides a way for a system administrator to create a script that can be put into the system login process so that the environment variables are seen by all users.

To use a shell script for generating an initialization script file, follow these steps:

1. cd to the root directory where program files have been loaded
   `cd accbin`
2. If you are running csh shell, enter: `source crsp_setup.csh`

   If you are running ksh or bash shell, enter: `./ crsp_setup.sh`
3. The script will prompt for data, root, and log directories. Follow the instructions on the prompts in terms of trailing slashes in directory names.

The script will create new scripts, `mycrsp.cshrc` in `csh` or `mycrsp.kshrc` in `ksh`. `mycrsp` is the default that may be changed.

Note: When creating a custom `kshrc` file, be aware that it will overwrite a like-named file if one exists rather than create a new version. The `kshrc` file will overwrite all environment variables, so must be completely filled in.

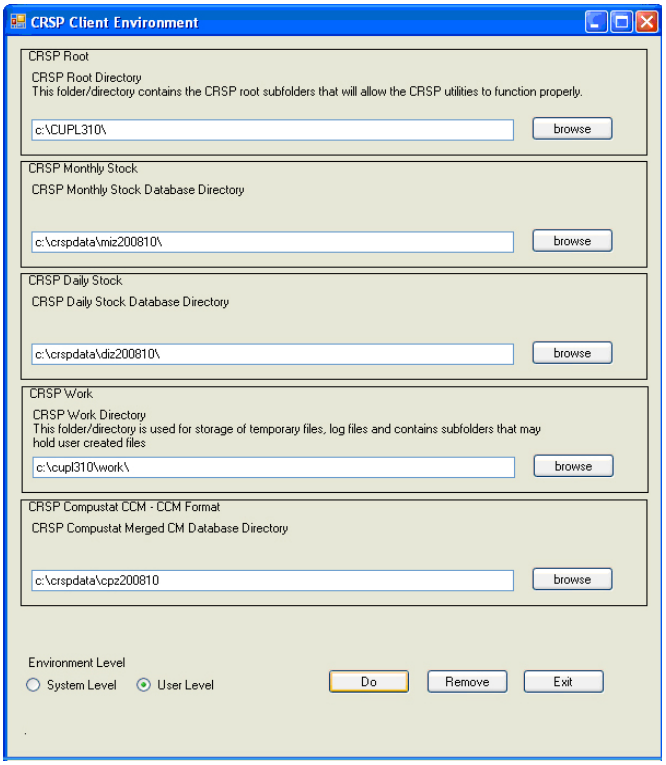
`env | grep CRSP` can be used to check the CRSP environment variables set.

PC Network Installation of CRSPAccess

CRSPAccess can be installed on a Windows network with Windows XP clients. Data, programs and libraries are loaded to a server machine that can be accessed by clients with access to the data. A separate client installation program is provided to configure the clients. Configuring a client involves installing program shortcuts to CRSP programs and setting environment variables on the client workstation. The executable, `client_environment.exe` is located in the crsproot\accbin folder and may be set to run on the user or system level.

- `client_environment.exe` run on the user level sets the environment variables on a computer for the current user.
- `client_environment.exe` on the system level sets the environment variables on a computer for all users of that machine and requires administrator privileges to run.
- The environment variable `CRSP_WORK` is defined. The directory defined by `CRSP_WORK` must have write permission established for the intended users. This directory is used as storage for temporary files, log files and is a recommended location for user created folders and files.

The following steps will configure a client:



1. Run software and data installs first on the server machine. The program and data disks must be accessible and mapped on the client workstation. The client installer must know the path of the CRSP root folder and monthly and/or daily data folders in terms of the client disk mappings.
2. From the client workstation, execute the desired client install. client_environment.exe is located in the crsproot\accbin folder and should be run from this location.
3. Select whether the environment variables are to be set at the user or system level. Identify file locations for the programs and databases.
4. File locations will default to what is currently set on the system. If new databases or software are replacing existing versions and if the same data locations are used, it is not necessary to reset the environment variables.
5. Once locations are defined, click DO. DONE will appear in the lower left hand corner of the screen once the environment variables and shortcuts are set. The client_environment tool will be included with the Start Menu shortcuts in case future changes are needed.
6. The Remove option is useful for moving settings from user to system or vice versa. The Remove option erases all CRSP environment variables and shortcuts, thus should be used with caution. CRSP recommends making note of the variables and locations before running this option.

CHAPTER 3: USING C WITH CUPL

Windows Systems

CRSP software is tested and fully supported on Windows XP. All C libraries and sample programs were compiled and tested using the Microsoft Visual Studio 2005.

CRSP access relies on environment variables set during installation. The environment variables can also be set through the client_environment tool or Control Panel/System/Advanced/Environment menu on Windows XP. Environment variables can be used in command prompt windows with the name enclosed in percent `(%)` characters. The `set` command can be used in a command prompt window to show available environment variables. (e.g. `>set crsp`). See Installation Procedures (Page 9) for information on installing the CRSPAccess data and programs.

Important CRSP files and directories have the following names.

%crsp_bin%	folder containing executable sample programs and batch files. This folder should be in the `PATH` so programs can be run from any folder.
%crsp_lib%	folder containing CRSP object library and internal files
%crsp_lib%\crsp_dll.lib	CRSP dynamic link library
%crsp_lib%\crsp_lib.lib	CRSP object library
%crsp_include%	location of CRSP C Header Files referred to by `INCLUDE` statements
%crsp_sample%	folder containing CRSP sample programs
%crsp_mstk%	folder containing monthly CRSP stock and index databases
%crsp_dstk%	folder containing daily CRSP stock and index databases
%crsp_ccm%	folder containing CCM database
%crsp_work%	folder identified for user containing log, temporary and other user-generated files

Using the crsp_dll.lib

CRSP_dll.lib is included in both 32-bit and 64-bit CRSP-supported Windows platforms.

If you are using the CRSP dynamic link library, crsp_dll.lib, make note of the following:

1. Your program must be modified if it uses the global CRSP err-msg string to report CRSP error messages, or if it uses the crsp_file_fopen function. The stk_samp1.c sample program shows proper use of the crsp_errprintf function to print CRSP error messages and can be used directly with the CRSP DLL.

2. To compile a program with the CRSP DLL, the library file crsp_dll.lib must be used instead of crsp_lib.lib. This can be done by simply switching the file names when adding the library file to your project, in the library definition in an NMAKE file, or on the command line.

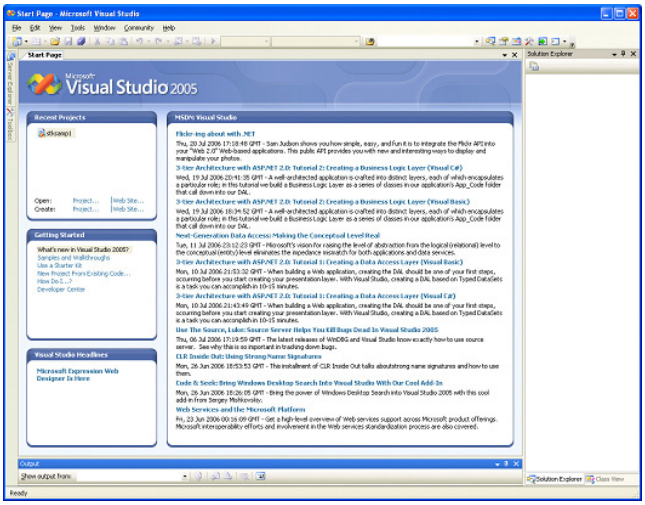
3. The %crsp_bin% folder must be in the PATH at run-time. CRSP installs do not set the PATH automatically unless running in a CRSP Command Prompt window. The user must set PATH directly under Control Panel/System/Advanced/Environment Variables, or with a SET command in the shortcut or Window prior to running the program.

C Compiler Instructions

Following is an example of compiling a sample C program using Microsoft Visual Studio 2005, which CRSP supports for compiling C programs in a Windows environment.

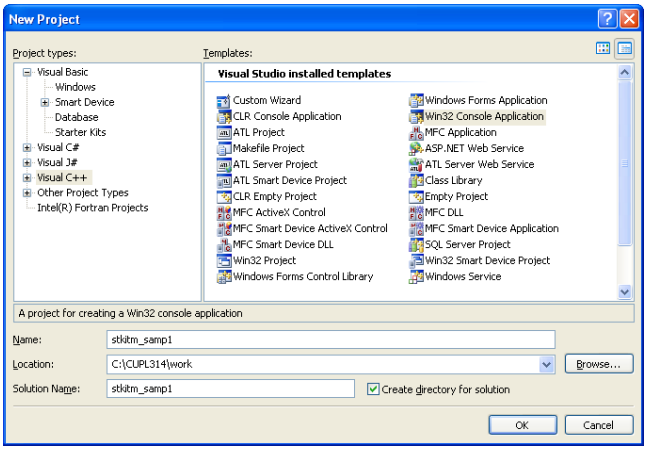
Step 1:

To begin, **Start→Programs→Microsoft Visual Studio 2005**. Click on the **Create: Project** button on the left of your screen, or from the Menu bar select **File→New→Project**.



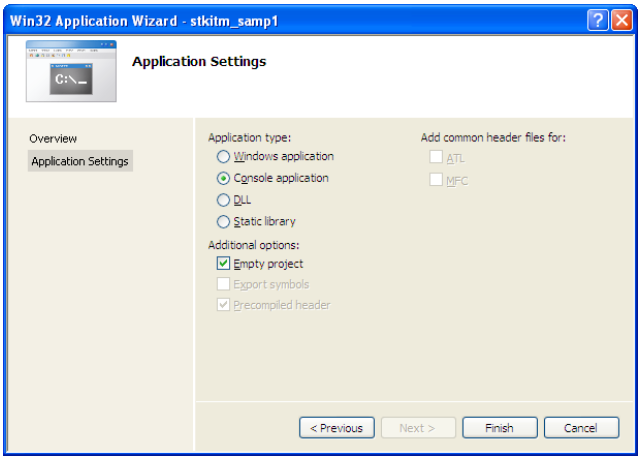
Step 2:

To create a new project, highlight the **Visual C++** folder in the Project Types box on the left and the **Win32 Console Application** in the right Templates box on the right. Enter the name of the project you are creating in the Name box below as "stkitm_samp1". Move the cursor to Location and overwrite as `C:\CUPL322\work` , or the directory in which you wish to work. Click **OK**.



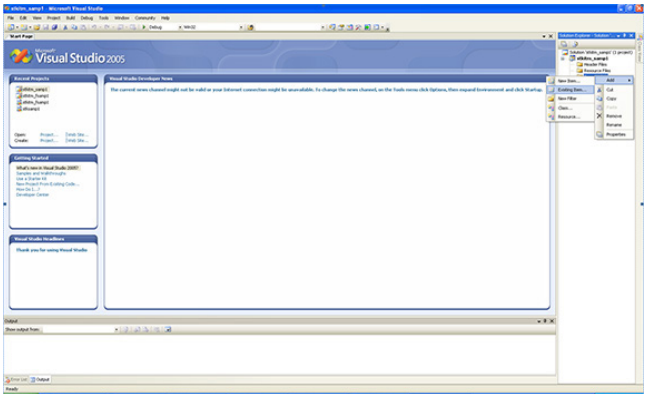
Step 3:

The Win32 Application Wizard will open. Click on **Application Settings**. Within this screen, confirm that Console application is turned on. Check **Empty project** and Click **Finish**.



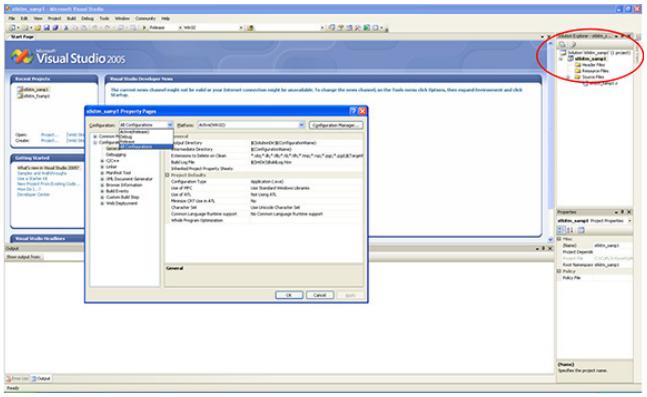
Step4:

You are ready to add items to the **stkitm_samp1** project that you are creating. On the right side of the screen, in the Solution Explorer box, below stkitm_samp1, right click on **Source Files** and follow **Add→Existing Item**. Browse for the sample program, `stkitm_samp1.c` from your `c:\CUPL322\sample\` directory. Double click to add it. Click on the "plus" sign next to **Source Files** for `stkitm_samp1.c` to appear. Double click on `stkitm_samp1.c` to display the program.



Step 5:

Right click on the `stkitm_samp1` project and select Properties at the bottom of the drop-down. The Property Pages screen will pop-up. Click on the drop-down Configuration options and select **All Configurations.**

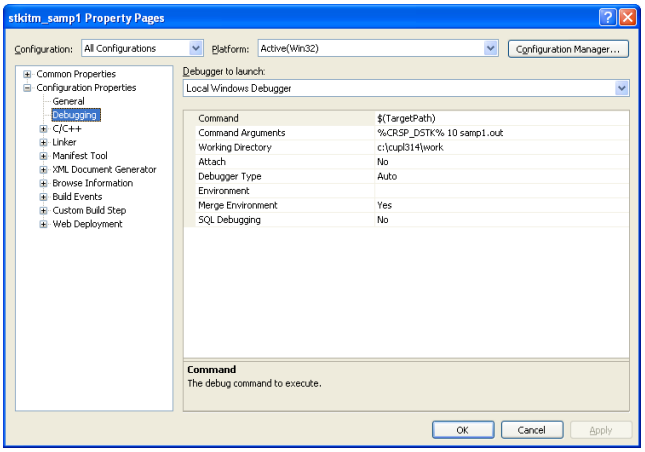


Step 6:

On the **Configurations** panel, select **Debugging** and enter **Command Arguments**:

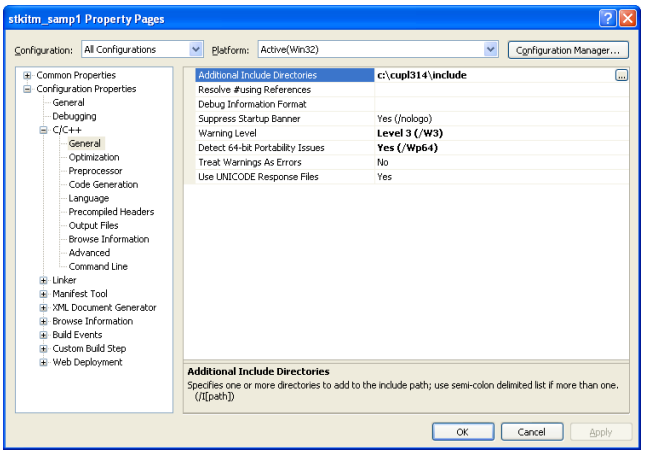
```
%crsp_dstk% 10 samp1.out
```

Command arguments are remarked within the sample programs. In this example, `%crsp_dstk%` is the environment variable pointing to the daily stock database, 10 is the daily stock setid, and samp1.out is the name given to the output file. Enter the location of your working directory - the location of your C project.



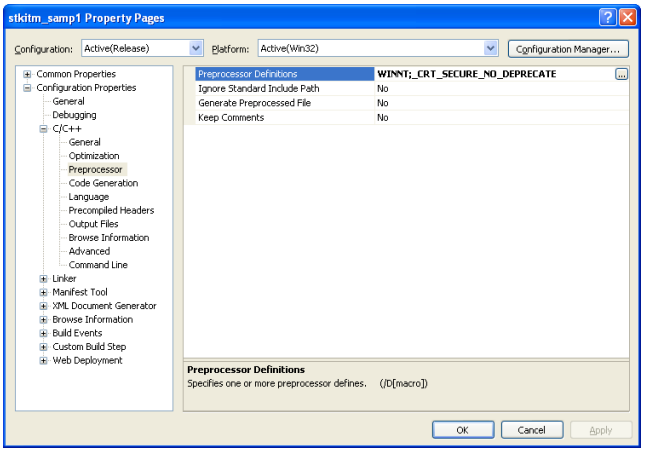
Step 7:

Click on the **C/C++ folder→General**, highlight **Additional Include Directories** and enter the location of your include files. This will be in your `c:\CUPL322\include` directory.



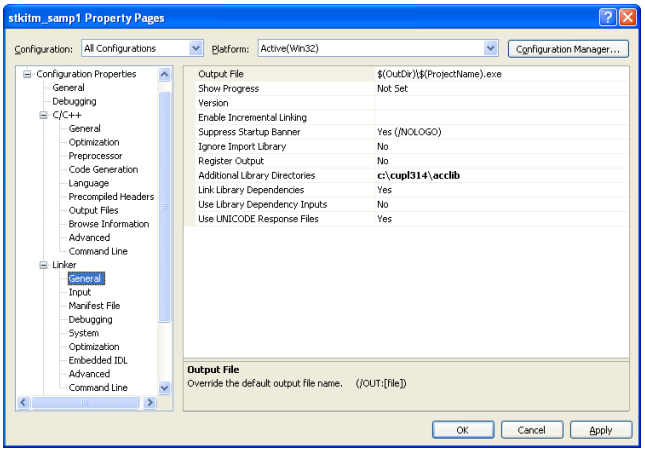
Step 8:

Still within the **C/C++** folder, click on **Preprocessor**. Highlight **Preprocessor Definitions** and add **WINNT** as shown. Optional: Also adding `;_CRT_SECURE_NO_DEPRECATE` can reduce warning messages related to use of standard string functions.



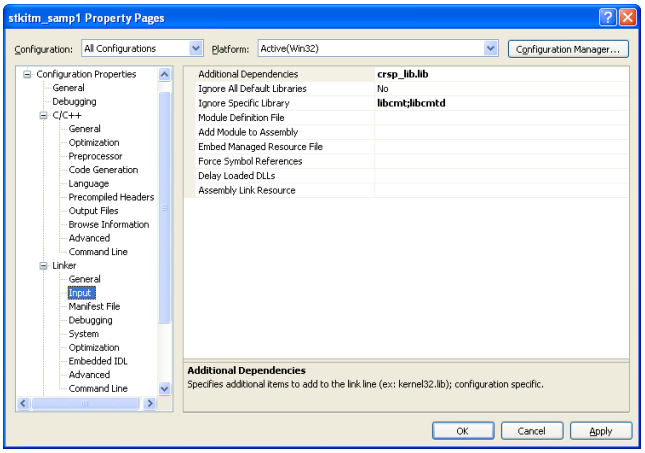
Step 9:

Go next to **Linker** and select **General**. To **Additional Library Directories**, add `c:\CUPL322\acclib`.



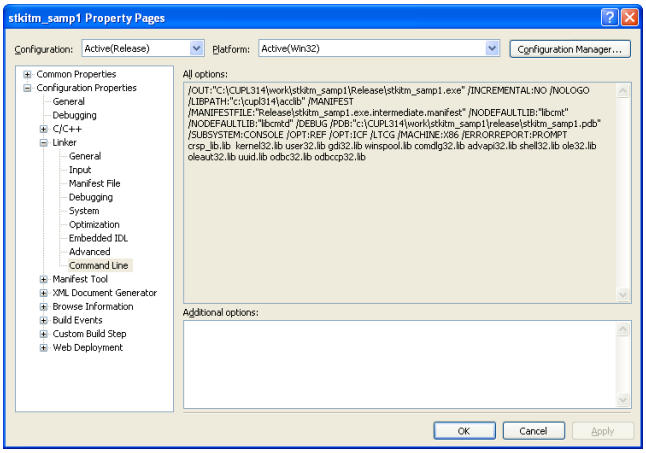
Step 10:

Still under **Linker**, click on the Input folder. To **Additional Dependencies,** add `crsp_lib.lib`. Depending on whether you intend to run your program in Debug or Release mode, you may add to **Ignore Specific Library** either or both: `libcmt; libcmtd.`



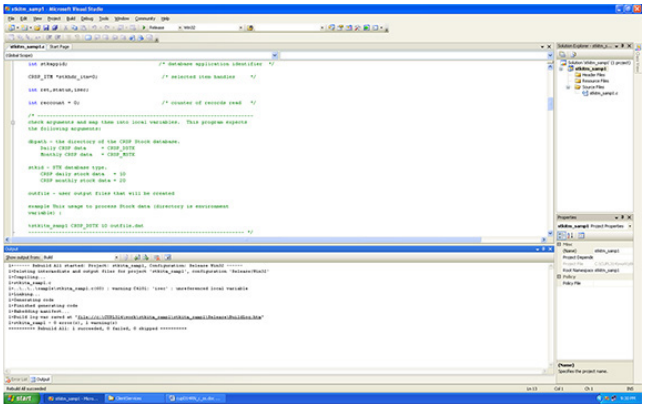
Step 11:

Finally, Select **Command Line** and click Apply. Your screen may appear blank. If you click OK and then reenter the Properties Pages from the stkitn_samp1 project, you will see that the command is set.



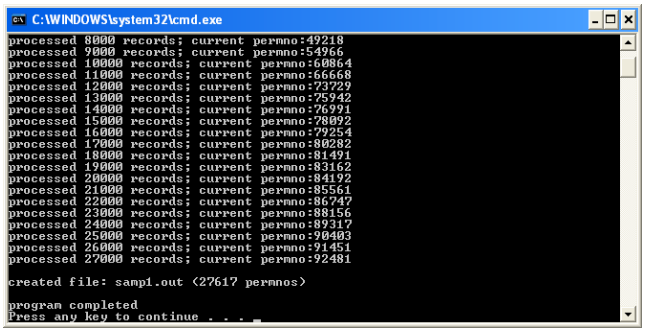
Step 12:

You will return to the design screen where the `stkitm_samp1` program is displayed. You are now ready to build your project. You may build and run the program in either **Debug** or **Release** mode. In the center, top of your screen, select the mode by using the pull-down menu. Either will work, though for this example, select **Debug**. From the Menu bar, click on **Build→Build stkitm_samp1**. In the Output dialog box at the bottom of your screen, you will see the following message: Build: 1 succeeded, 0 failed, 0 skipped – meaning that the build was successful!



Step 13:

To run the program that you have just built, from the Menu bar, click on: **Debug→Start Without Debugging**. The program will commence and work sequentially through the CRSP PERMNOs.



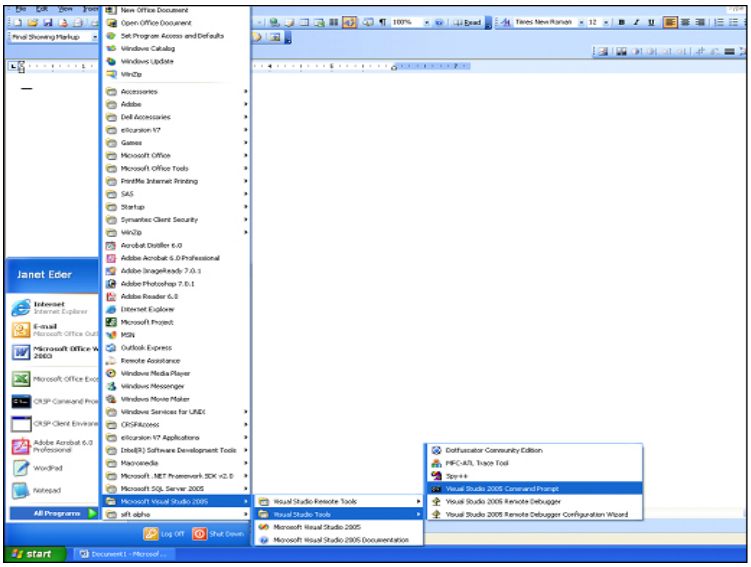
Step 14:

The output that you created by running the stksamp1 program is stored in the folder where you initially created the project: `c:\CUPL322\work\stkitm_samp1`. The output file is a text file called `samp1.out`.

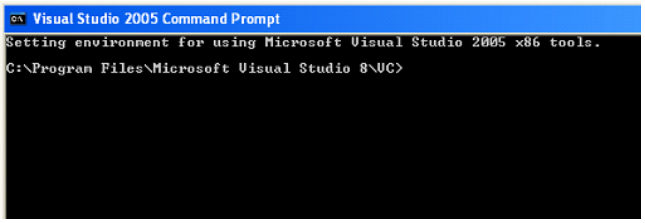
Using the Command Prompt Window

The programs can also be compiled, linked, and run from a command prompt window. In order to do so, the environment variables for Microsoft Visual Studio 2005 must be set.

To set the environment to the Visual Studio 2005 click on **Start→All Programs→Microsoft Visual Studio 2005→Visual Studio Tools→Visual Studio 2005 Command Prompt.** When you do this you will open a prompt like below and then go to your directory using the appropriate DOS command:



A DOS window will open ready for you to run your C++ programs.



Copy the sample program to a local directory using the Explorer utility or the command prompt copy command, or use the Developer Studio to open the file and save to a new location with Save As.

Sample programs can be found in the `%crsp_sample%` directory. The command prompt command, `echo %crsp_sample%` can be used to get the explicit directory needed. The explicit paths for `%crsp_include%` and `%crsp_lib%` will be needed to set up projects in the Microsoft Visual Studio 2005. These too can be identified using the echo command.

```
> copy %crsp_sample%\stkitm_samp1.c .
```

```
> cl /D WINNT=2 /I%crsp_include% stkitm_samp1.c %crsp_lib%\crsp_lib.lib
```

```
> .\stkitm_samp1 %crsp_dstk% 10 myfile.out
```
to run the program

Sample programs can also be compiled and linked using the `nmake` utility. The file *c_samp.mak* in the `%crsp_sample%` directory is a description file to maintain the two stock sample programs. To run, copy the file to your program directory and run the utility with the command:

```
> nmake /f c_samp.mak stkitm_samp1.exe
```
to compile a specific sample program

```
> nmake /f c_samp.mak
```
to compile all sample programs

```
> .\stkitm_samp1 %crsp_dstk% 10 myfile.out
```
to run the program

Sun Solaris Systems

CRSP currently supports Sun Sparc Solaris 2.9 with the Forte Developer 7 C 5.4.

CRSP access relies on environment variables set during installation. Environment variables can be used on Unix with the name preceded by `$`. All file names and environment variable names are case-sensitive on Unix systems. The `env` command can be used in a terminal window to find available environment variables.

Important CRSP files or directories can be found with the following names.

`$CRSP_BIN`	directory containing Executable Sample Programs and Batch Files. This directory is in the PATH so programs can be run from any directory.
`$CRSP_LIB`	directory containing CRSP object library and internal files.
`$CRSP_LIB/crsplib.a`	CRSP object library.
`$CRSP_INCLUDE`	directory containing CRSP Header Files referred to by #INCLUDE statements.
`$CRSP_SAMPLE`	directory containing CRSP sample programs.
`$CRSP_MSTK`	directory containing monthly CRSP stock and index databases.
`$CRSP_DSTK`	directory containing daily CRSP stock and index databases.
`$CRSP_CCM`	directory containing CCM database.

Following is an example of how to modify and to run a sample C program with Sun Solaris.

Sun – SparcCompiler C 5.1

Command line:

```
> cp $CRSP_SAMPLE/stkitm_samp1.c .
```

```
> chmod 660 stkitm_samp1.c
```

> *Use an available text editor to make desired code changes.*

```
> cc -DUNIX=1 -DSOLARIS -I$CRSP_INCLUDE -xarch=v9 -w -KPIC -o stkitm_samp1
  stkitm_samp1.c $CRSP_LIB/crsplib.a -lm
```

```
> ./stkitm_samp1 $CRSP_DSTK 10 myfile.out
```
to run the program

Sample programs can also be compiled and linked using the make utility. The directory `$CRSP_SAMPLE` contains sample `make` description files for Sun Solaris named *c_samp.mk*. To use make, copy the relevant description file to your program directory, edit it to support the program(s) of interest and create local executables, and run with the command:

Make file:

```
> make -f c_samp.mk stkitm_samp1
```
to compile a specific sample program

```
> make -f c_samp.mk
```
to compile all sample programs

```
> ./stkitm_samp1 $CRSP_DSTK 10 myfile.out
```
to run the program

Linux Systems

CRSP supports C programming for Linux Red Hat Enterprise Linux 3.0 on Intel x86 machines. C functions were compiled and tested using the gcc 3.2.3 compiler.

CRSP access depends on environment variables set during installation. Environment variables can be used on Linux with the name preceded by `$`. All file names and environment variable names are case-sensitive on Linux systems. The `env` command can be used in a terminal window to find available environment variables.

Important CRSP files or directories can be found with the following names.

$CRSP_BIN	directory containing Executable Sample Programs and Batch Files. This directory is in the PATH so programs can be run from any directory.
$CRSP_LIB	directory containing CRSP object library and internal files.
$CRSP_LIB/crsplib.a	CRSP object library.
$CRSP_INCLUDE	directory containing CRSP header files referred to by #INCLUDE statements.
$CRSP_SAMPLE	directory containing CRSP sample programs.
$CRSP_MSTK	directory containing monthly CRSP stock and index databases.
$CRSP_DSTK	directory containing daily CRSP stock and index databases.
$CRSP_CCM	directory containing CCM database.

Following is an example of how to modify and to run a sample C program with Linux – gcc 3.2.3:

Command line:

```
> cp $CRSP_SAMPLE/stkitm_samp1.c .

> chmod 660 stkitm_samp1.c
```

> Use an available text editor to make desired code changes.

```
> gcc -DUNIX=1 -DUNIX2=1  -I$CRSP_INCLUDE -w -fPIC stk_samp.c -o stkitm_samp1
  $CRSP_LIB/crsplib.a -lm

> ./stkitm_samp1 $CRSP_DSTK 10 myfile.out
```

to run the program

Sample programs can also be compiled and linked using the `make` utility. The directory `$CRSP_SAMPLE` contains sample `make` description files for Linux, named *c_samp_stk.mk*. To use the `make` file, copy the relevant description file to your program directory, edit it to support the program(s) of interest and create local executables, and run with the commands:

Make file:

```
> make -f c_samp.mk stkitm_samp1
```

to compile a specific sample program

```
> make -f c_samp.mk
```

to compile all sample programs

```
> ./stkitm_samp1 $CRSP_DSTK 10 myfile.out
```

to run the program

CHAPTER 4: USING FORTRAN-95 WITH CUPL

Windows Systems

CRSP supports FORTRAN-95 on Windows XP and on Windows 7. Windows subscribers who are using the CRSP Programming Libraries for Fortran95 will find the following options available to them:

- **32-bit XP: Command Line** – Visual Studio 2008 is supported and will run with Fortran Intel Compilers 9.1 or higher.
- **Visual Studio Interface** – Visual Studio 2005 remains supported when used with the Intel Compiler 9.1. Visual Studio 2008 is now supported but requires Intel Compiler 10 or above.
- **Windows 7** – Intel Visual Fortran 2011/Parallel Studio XE is supported.

To use FORTRAN-95, you must have both Visual Studio and the compatible Fortran compiler installed on your computer. installed on your computer.

CRSP access relies on environment variables set during installation. Environment variables can be used on Windows systems with the name enclosed with `%` characters `(%name%)`. The set command can be used in a command prompt window to view available environment variable settings.

Important CRSP files or folders can be found with the following names:

`%crsp_bin%` -	executable programs and batch files. This folder is in the PATH so programs can be run from any directory. Executable versions of the sample programs can be found in this folder.
`%crsp_lib%` -	folder containing CRSP object library and internal files.
`%crsp_lib%\crsp_lib.lib`	CRSP object library.
`%crsp_lib%\crsp_lib_f95.lib`	CRSP FORTRAN-95 object library.
`%crsp_include%` -	folder containing CRSP FORTRAN header files referred to by INCLUDE statements.
`%crsp_sample%` -	folder containing CRSP sample programs.
`%crsp_mstk%` -	folder containing monthly CRSP stock and index databases.
`%crsp_dstk%` -	folder containing daily CRSP stock and index databases.
`%crsp_ccm%` -	folder containing CCM database.

CRSP has updated instructions for running Fortran-95 programs using Visual Studio 2005 on XP. These instructions will also intuitively work for Visual Studio 2008. The instructions focus on setting options at the project level.

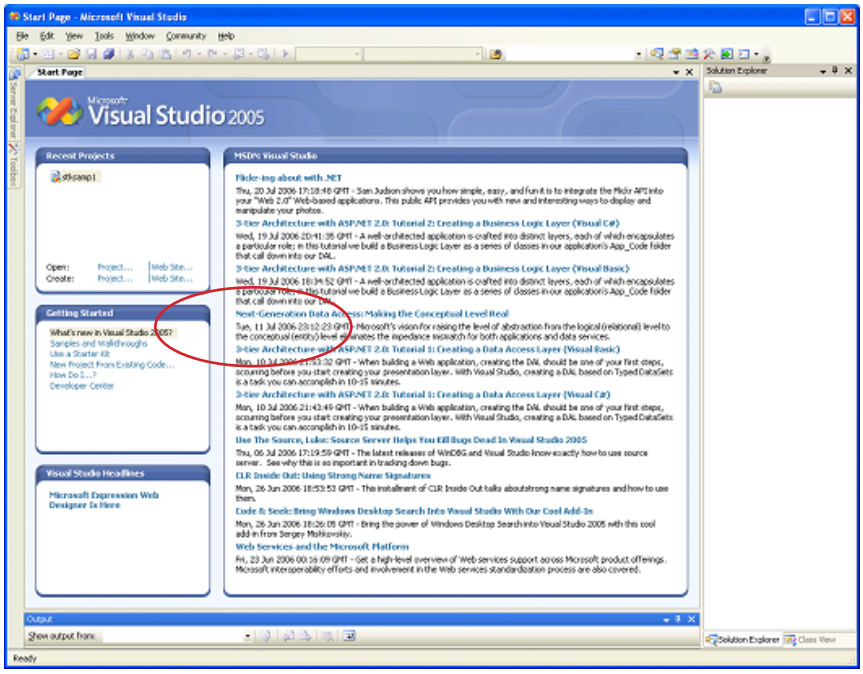
If you have used Visual Studio previously and followed CRSP instructions, it is important to be aware of this change. Setting options for Libraries and Include files under Tools→Options hard-codes paths to these files that may override more updated library and include files. *We recommend that you be sure to clear the Library and Include paths set in Options in order to ensure that you are accessing the correct files.*

Compiling, Building and Executing FORTRAN-95 Programs on Windows Systems

The following walks you through the steps to build and run Stkitm_fsamp1.F90. This sample program is located in the Sample folder in the crsp root directory where you have installed the CRSPAccess software.

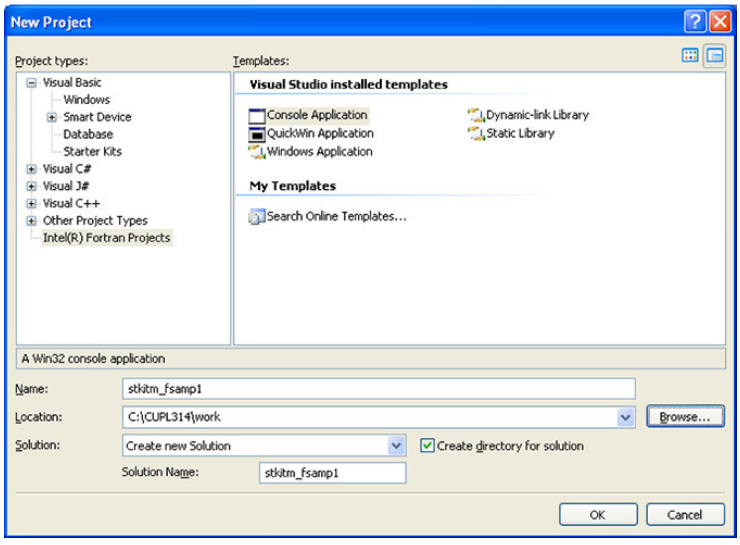
Step 1:

Open the Microsoft Visual Studio 2005 development environment start page. **Start→Programs→Microsoft Visual Studio 2005** opens the screen below. Click on the **Create: Project** button at the upper left of your screen, or from the Menu bar select **File→New→Project**.



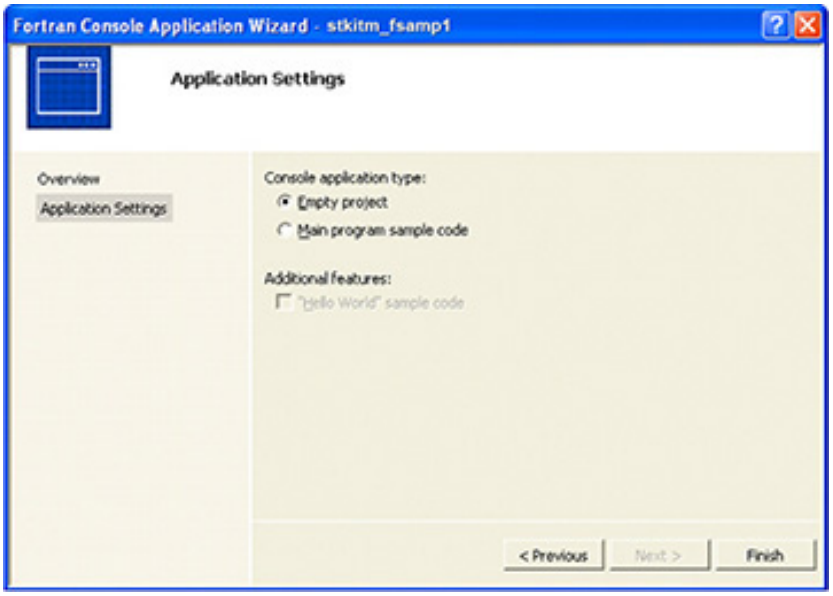
Step 2:

To create a new project, highlight the Intel(R) Fortran Projects folder in the Project Types box on the left and Console Application in the Templates box on the right. Enter the name of the project you are creating in the Name box below as "`stkitm_fsamp1`". Move the cursor to the Location and overwrite as `C:\CUPL322\work\`, or the directory in which you wish to work. Click **OK**.



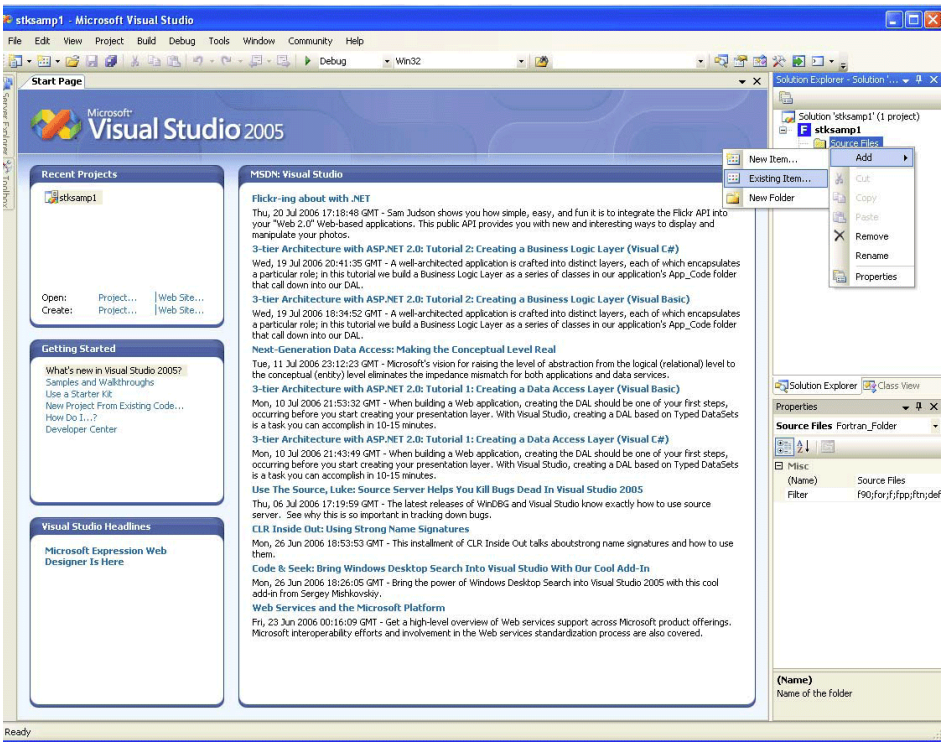
Step 3:

You have now opened the Fortran Console Application Wizard. Click on **Application Settings→Empty project**. Click **Finish**.



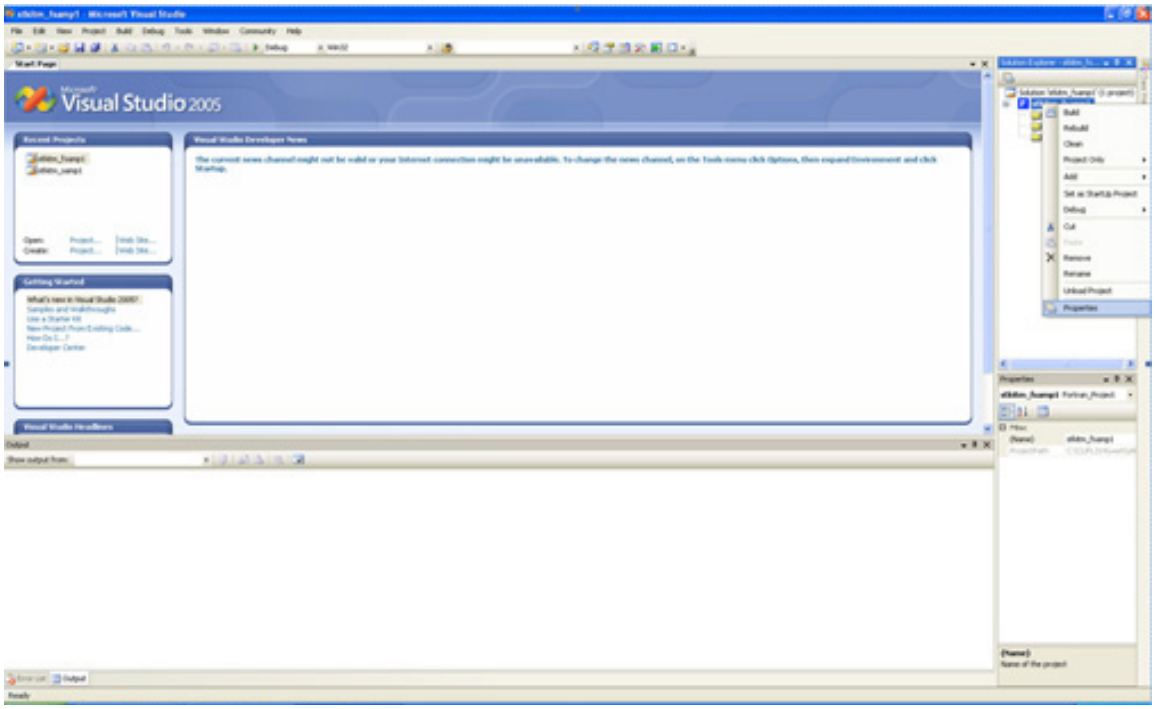
Step 4:

You are ready to add items to the `stkitm_fsamp1` project that you are creating. On the right side of the screen, in the Solution Explorer box, click the plus sign by "stkitm_fsamp1", then right click on Source Files and follow **Add→Add Existing Item**. Browse for the sample program "stkitm_fsamp1.f90" in the directory C:\CUPL322\sample\. Double click to add it. Click on the "plus" sign next to Source Files for stkitm_fsamp1.f90 to appear. Double click on stkitm_fsamp1.f90 to display the program.



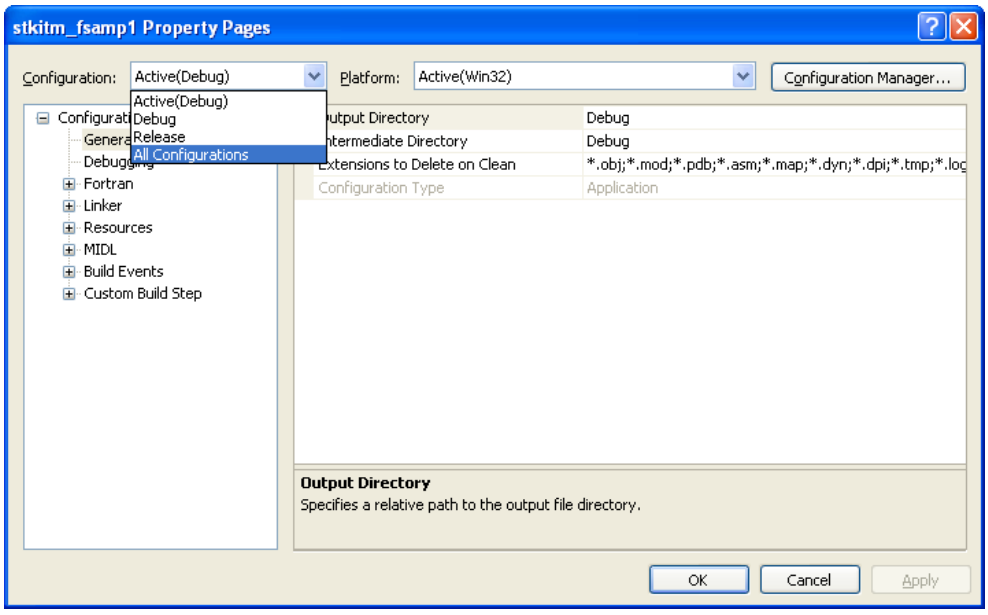
Step 5:

Once your new project is added, right-click on **stkitm_fsamp1** and scroll down the menu panel to select Properties. You will advance to the Properties Pages. Changes to project solutions are reflected in the following 3 steps.



Step 6:

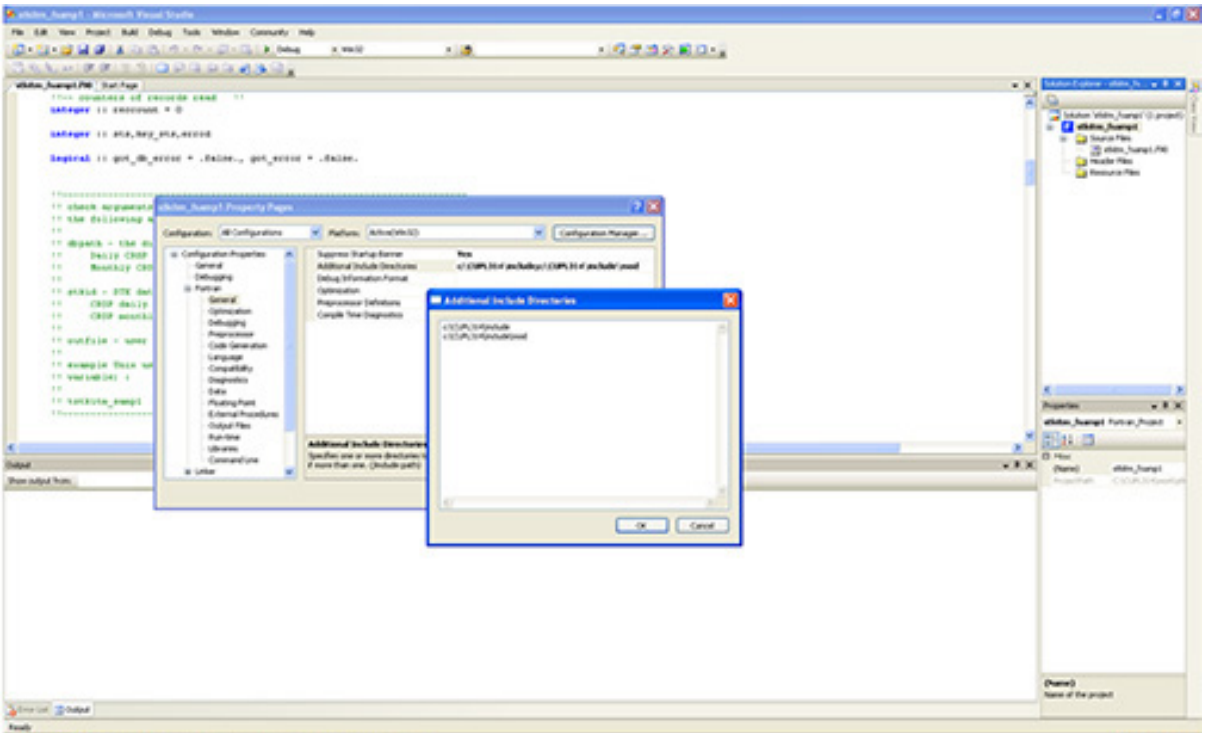
In order to be able to run the sample program in either Debug or Release mode, Click on the Configuration drop-down and select **All Configurations**.



Step 7:

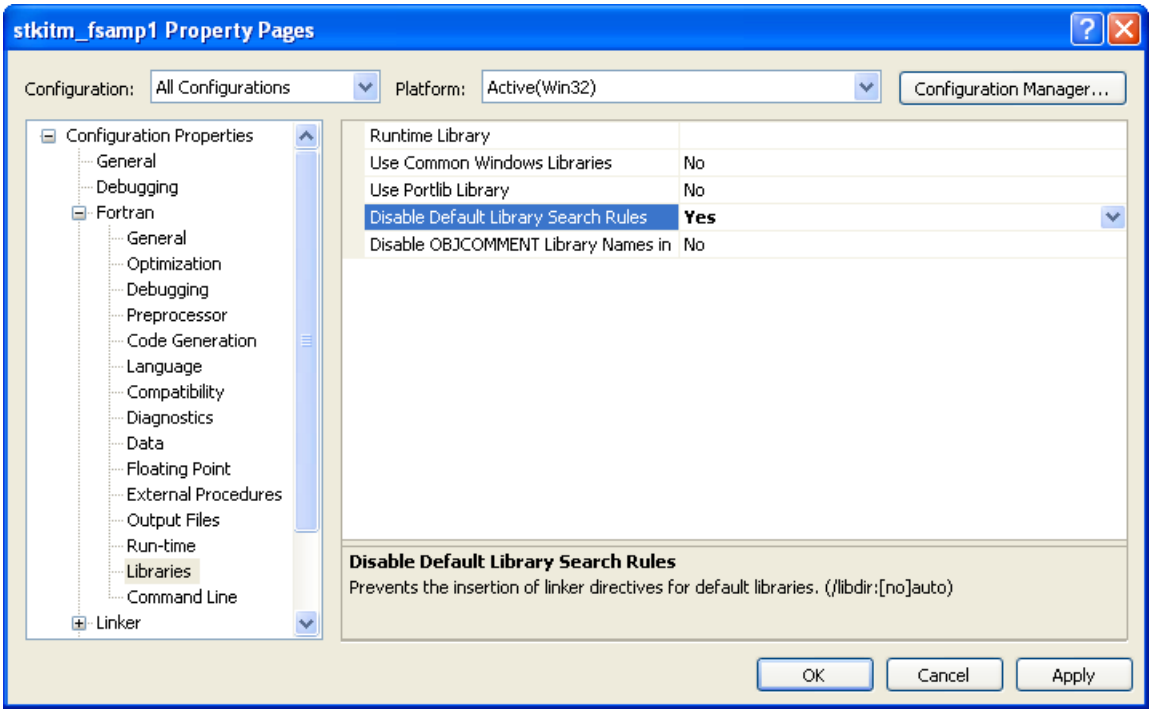
Highlight **Fortran** and select **General**. Click on Additional Include Directories and add

`C:\CUPL322\include` and `C:\CUPL322\include\mod`



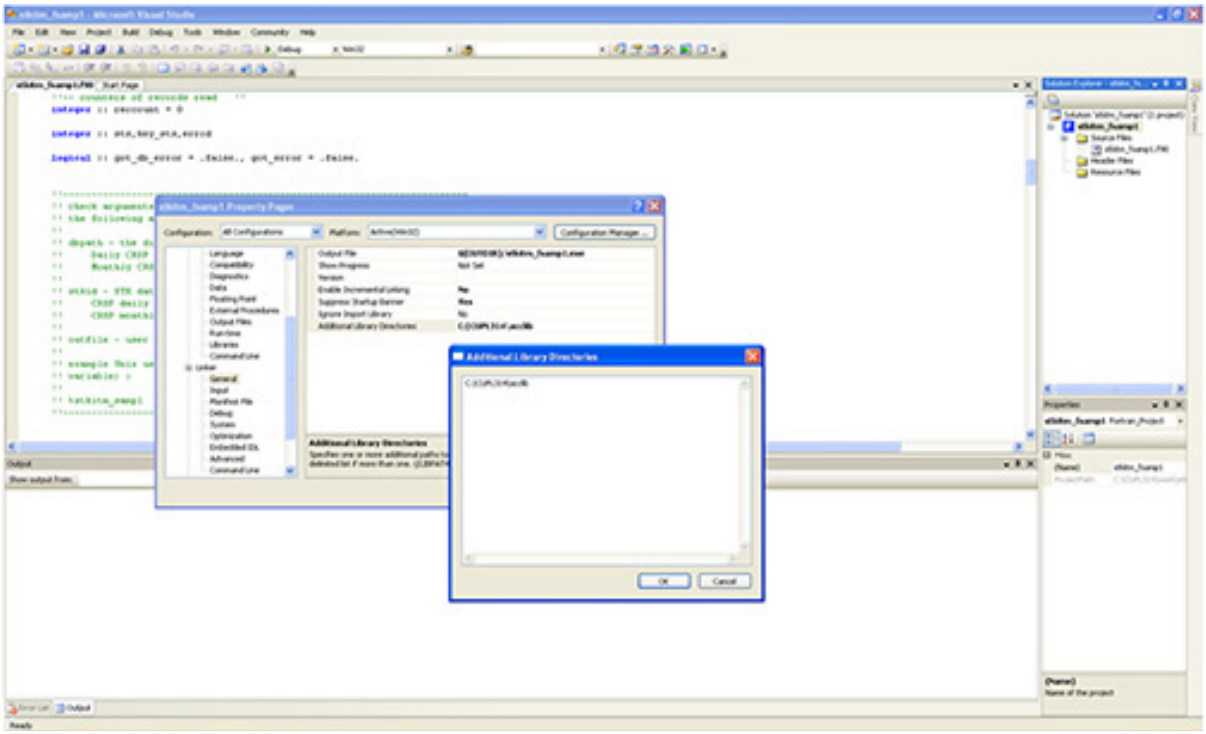
Step 8:

Select **Libraries**, highlight **Disable Default Library Search Rules** and select **Yes**.



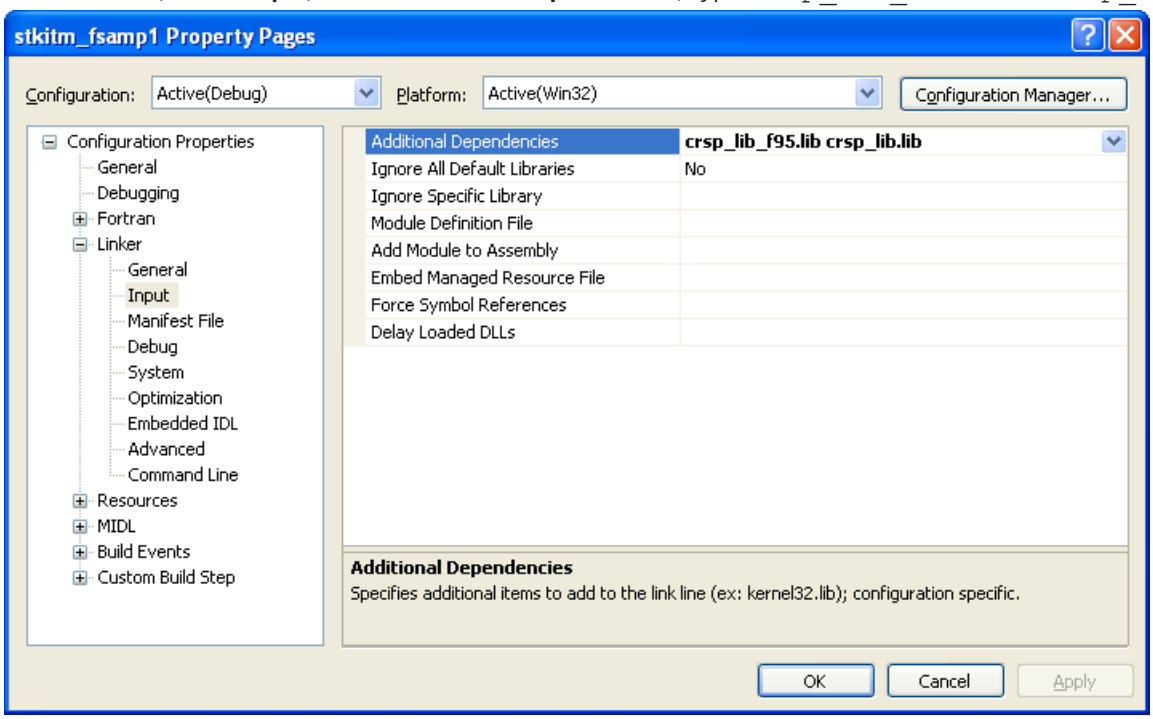
Step 9:

Select **Linker→General, Additional Library Directories** and enter `C:\CUPL322\acclib`.



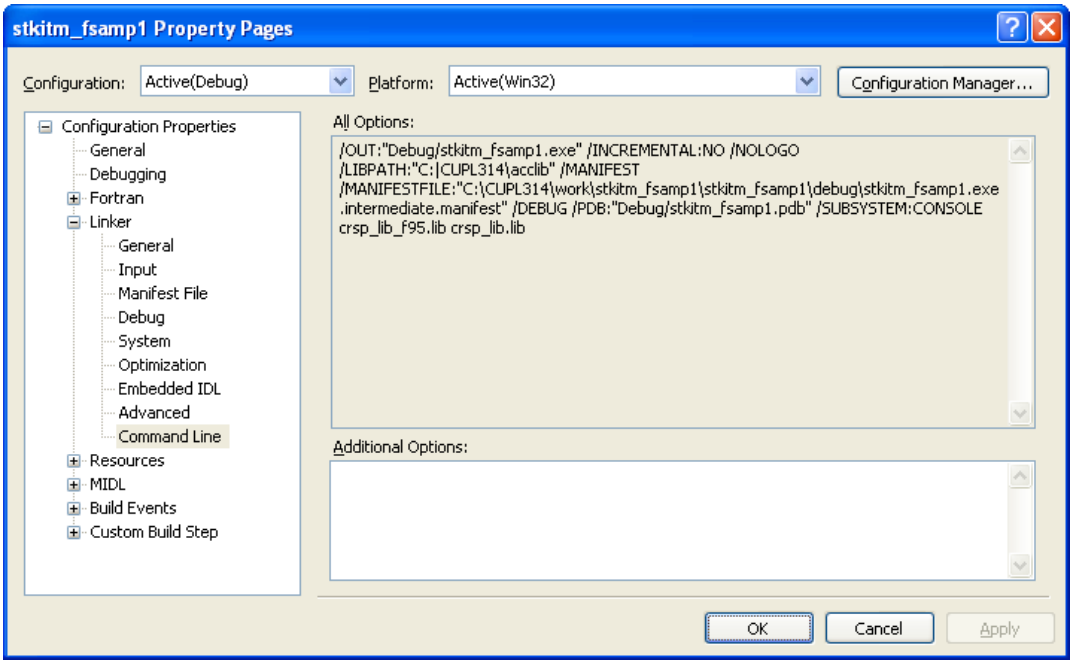
Step 10:

Still in **Linker**, select **Input**, and in **Additional Dependencies**, type `crsp_lib_f95.lib crsp_lib.lib`.



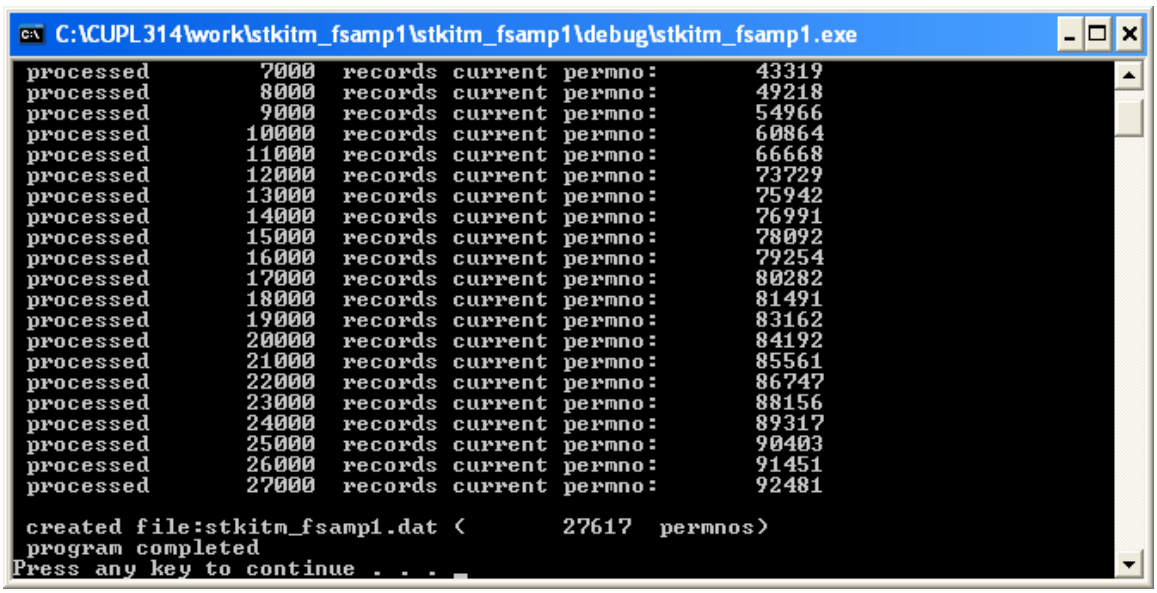
Step 11:

Select Command Line and click on Apply to set the properties of your project.



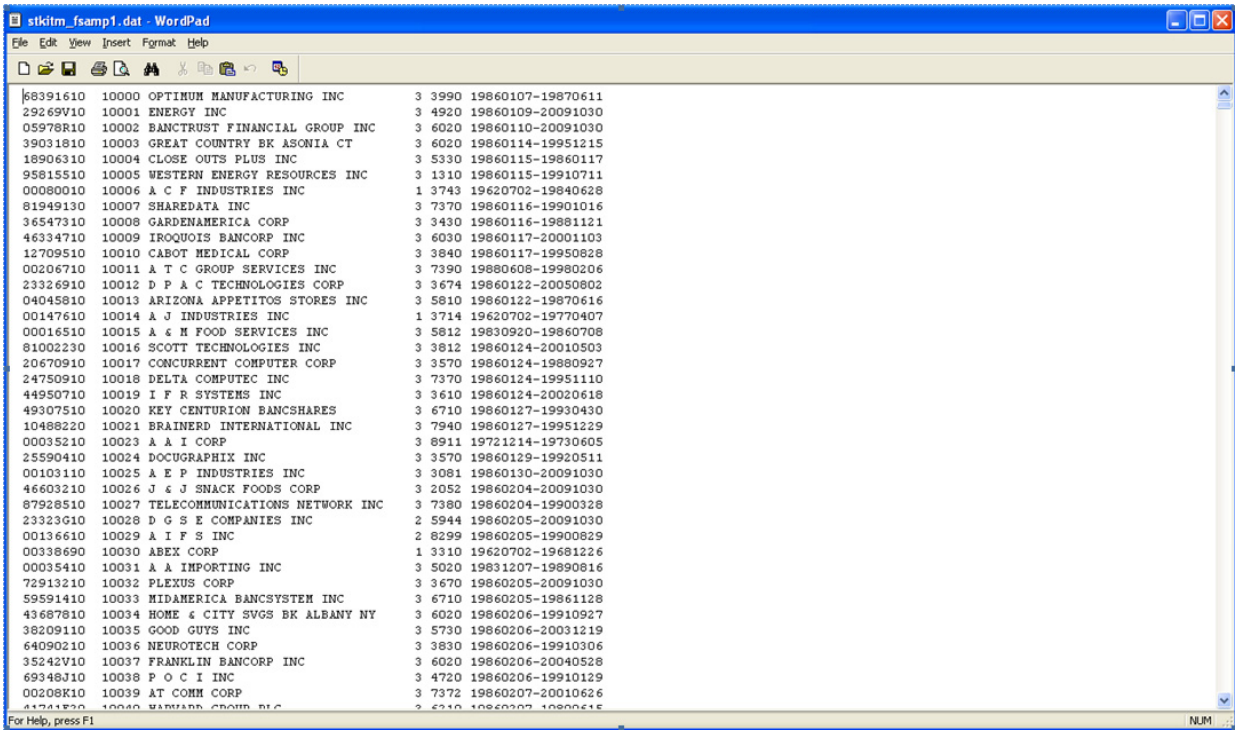
Step 12:

To run the program you have just built, from the menu bar, click on: **Debug→Start without Debugging**. The following screen will appear and indicate that the program is complete:



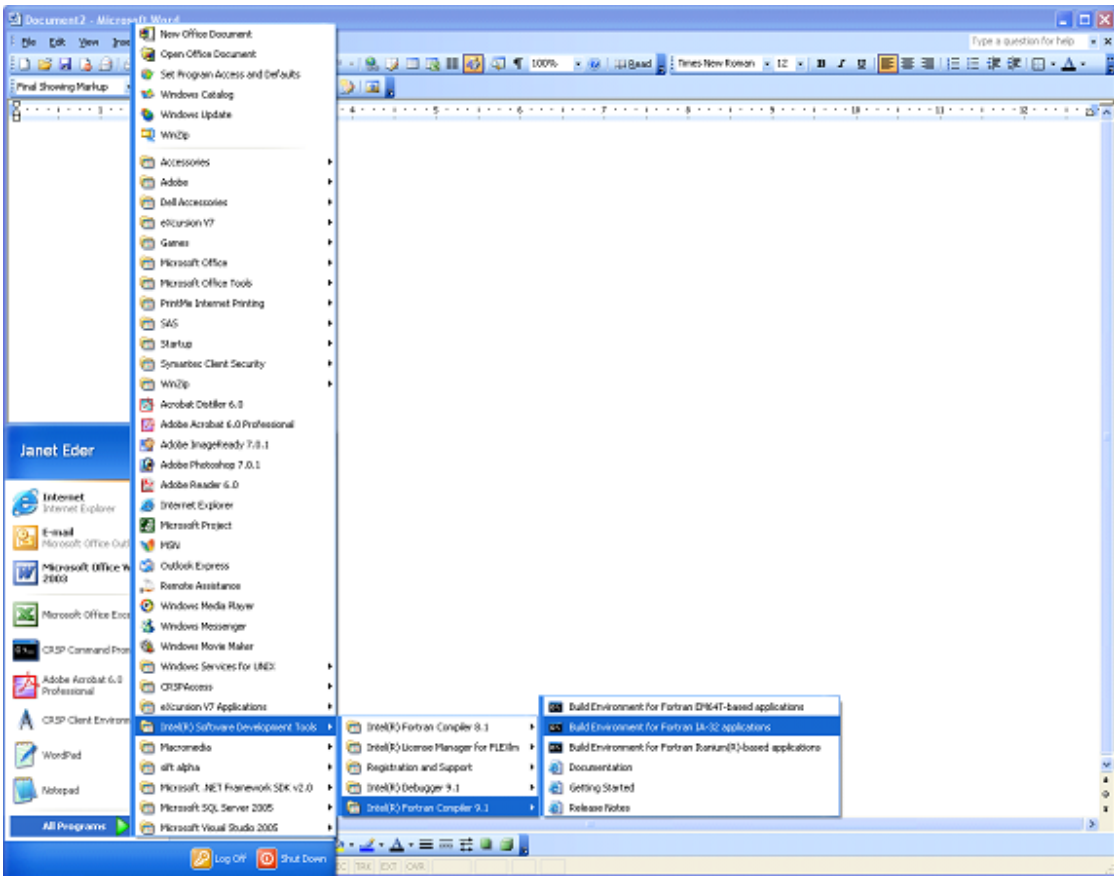
Step 13:

The output that you created by running the stkitm_fsamp1 program is stored in the folder where you initially created the project: `c:\CUPL322\work\stk_samp1`. The output file is a text file called "`stkitm_fsamp1.dat`". Note that your output may differ depending on the end date of the database that you are using.

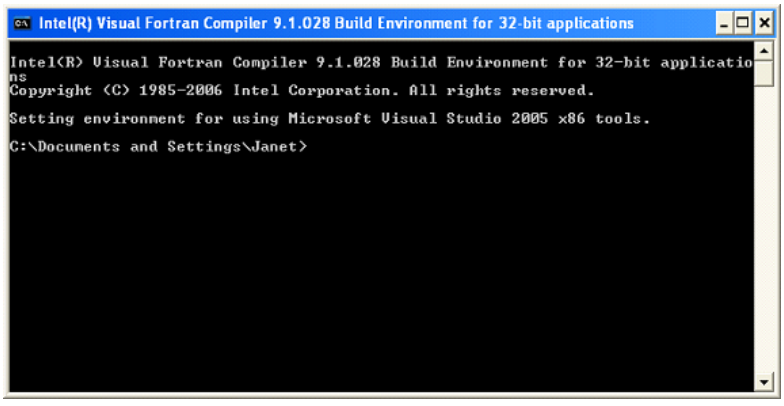


The programs can also be compiled and run from a command prompt window. In order to do so, the environment must be set for Intel FORTRAN to run.

To set the Windows 32-bit environment to Intel(R) Fortran click on **Start→All Programs→Intel(R) Software Development Tools→Intel(R) Fortran Compiler 9.1→Build Environment for Fortran IA-32 applications**. When you do this you will open a prompt like below and then go to your directory using the appropriate DOS command:



A DOS window will open ready for you to run your FORTRAN-95 programs.



To set the environment for Windows 64-bit, click on **Start→All Programs→Intel Parallel Studio XE 2011→Paralell Studio XE with Intel Compiler→Intel 64 Visual Studio 2008 mode.**

Command line:

```
> copy %crsp_sample%\stkitm_fsamp1.f90 .

> ifort /I%crsp_include% /I%crsp_include%\mod stkitm_fsamp1.f90 %crsp_lib%\
  crsp_lib.lib %crsp_lib%\crsp_lib_f95.lib

> .\stkitm_fsamp1                                    to run the program
```

Sample programs can be compiled and linked at the command prompt using the nmake utility. A sample description file, f95_samp.mak, exists in the %crsp_sample% directory. To use the sample description file, copy it to your program directory, modify it to include your program, and run with the command.

Make file:

```
> copy %crsp_sample%\f95_samp.mak .

> nmake /f f95_samp.mak stkitm_fsamp1.exe        to compile a specific sample program

> nmake /f f95_samp.mak                          to compile all sample programs

> .\stkitm_fsamp1                                to run the program
```

Sun Solaris Systems

CRSP currently supports Sun Sparc Solaris 2.9/5.9 with the Forte Developer 7.0, FORTRAN-95 7.0 and Sun X 86 Solaris 2.9/5.9

FORTRAN was compiled and tested using the above compiler. FORTRAN library functions interface with C functions in the CRSP object library. Ordinary sample FORTRAN usage links to the object library, but does not require compiling C programs.

CRSP access depends on environment variables set during installation. Environment variables can be used on Unix with the name preceded by the `$` symbol. All file names and environment variable names are case sensitive on Unix systems. The `env` command can be used in a terminal window to find available environment variables.

Important CRSP files or directories can be found with the following names:

$CRSP_BIN	directory containing executable programs and shell scripts files. This directory is in the `PATH` so programs can be run from any directory. Executable versions of the sample programs can be found in this directory.
$CRSP_LIB	directory containing CRSP object library and internal files.
$CRSP_LIB/crsplib.a	CRSP C object library.
$CRSP_LIB/crsplib_f95.a	CRSP F95 object library.
$CRSP_INCLUDE	directory containing CRSP FORTRAN header files referred to by `INCLUDE` statements.
$CRSP_SAMPLE	directory containing CRSP sample programs.
$CRSP_MSTK	directory containing monthly CRSP stock and index databases.
$CRSP_DSTK	directory containing daily CRSP stock and index databases.
$CRSP_CCM	directory containing CRSP Link and COMPUSTAT database.

$CRSP_WORK directory created to hold user-generated files

Following is an example of modifying and running a sample FORTRAN program with Sun Solaris:

Sun – FORTRAN-95 8.2

Command line:

```
> cp $CRSP_SAMPLE/stkitm_fsamp1.f90 .
> chmod 660 stkitm_fsamp1.f90
```

Sparc:

```
> f95 -ext_names=plain -w -I$CRSP_INCLUDE -xarch=v9 -KPIC -o stkitm_fsamp1
  stkitm_fsamp1.f90 $CRSP_LIB/crsplib.a $CRSP_LIB/crsplib_f95.a
```

X86:

```
> f95 -w -xtarget=generic64 -ext_names=plain -I$CRSP_INCLUDE.() -KPIC -o
  stkitm_fsamp1 stkitm_fsamp1.f90 $CRSP_LIB/crsplib.a $CRSP_LIB/crsplib_f95.a
```

```
> ./stkitm_fsamp1                    to run the program
```

Sample programs can also be compiled and linked using the `make` utility. The sample program directory `$CRSP_SAMPLE` contains sample make description files for Sun Solaris in *f95_samp.mak*. To use `make`, copy the relevant description file to your program directory, edit it to support the program(s) of interest and create local executables, and run with the command:

Make file:

```
> make -f f95_samp.mk stkitm_fsamp1    to compile a specific sample program
>  make -f f95_samp.mk                 to compile all sample programs
> ./stkitm_fsamp1                     to run the program
```

Linux Systems

CRSP currently supports Linux, Red Hat 7.2 32-bit on Intel x86 and RHEL5 64-bit on Intel x86. FORTRAN was compiled and tested using the Lahey FORTRAN-95 Version 6.2 ro 32-bit and g95 Version 0.91 for 32-bit and 64-bit. FORTRAN library functions interface with C functions in the CRSP object library. Ordinary sample FORTRAN usage links to the object library, but does not require compiling C programs.

CRSP access depends on environment variables set during installation. Environment variables can be used on Linux with the name preceded by the `$` symbol. All file names and environment variable names are case sensitive on Linux systems. The `env` command can be used in a terminal window to find available environment variables.

Important CRSP files or directories can be found with the following names:

$CRSP_BIN	directory containing executable programs and shell scripts files. This directory is in the `PATH` so programs can be run from any directory. Executable versions of the sample programs can be found in this directory.
$CRSP_LIB	directory containing CRSP object library and internal files.
$CRSP_LIB/crsplib.a	CRSP object library.
$CRSP_LIB/crsplib_f95.a	CRSP F95 object library.
$CRSP_INCLUDE	directory containing CRSP FORTRAN header files referred to by `INCLUDE` statements.
$CRSP_SAMPLE	directory containing CRSP sample programs.
$CRSP_MSTK	directory containing monthly CRSP stock and index databases.
$CRSP_DSTK	directory containing daily CRSP stock and index databases.
$CRSP_CCM	directory containing CRSP Link and COMPUSTAT database.
$CRSP_WORK	directory created to hold user-generated files

Following is an example of modifying and running a sample FORTRAN program:

Linux – Lahey FORTRAN-95 Ver. 6.2 — 32-bit

Command line:

```
> cp $CRSP_SAMPLE/stkitm_fsamp1.f90 .

> chmod 660 stkitm_fsamp1.f90

> lf95 -w -I$CRSP_INCLUDE stkitm_fsamp1.f90 -o stkitm_fsamp1 $CRSP_LIB/
  crsplib.a $CRSP_LIB/crsplib_f95.a -lm

> ./stkitm_fsamp1                                    to run the program
```

Sample programs can also be compiled and linked using the make utility. The sample program directory `$CRSP_SAMPLE` contains sample make description files for Linux in f*95_samp.mak* for the Lahey compiler. To use the make file, copy the relevant description file to your program directory, edit it to support the program(s) of interest and create local executables, and run with the command:

Make file:

```
> make -f f95_samp.mk stkitm_fsamp1          to compile specific sample program
> make -f f95_samp.mk                        to compile all sample programs
> ./stkitm_fsamp1                            to run the program
```

g95 Ver. 0.91 32-bit, 0.91 64-bit

Command line:

```
> cp $CRSP_SAMPLE/stkitm_fsamp1.f90 .

> chmod 660 stkitm_fsamp1.f90

> g95 -o stkitm_fsamp1 -w stkitm_fsamp1.f90 -I$CRSP_INCLUDE $CRSP_LIB/crsplib.a
  $CRSP_LIB/crsplib_f95.a
  `find /usr/local -name libf95.a 2>&1 | grep libf95\.a -lm

> ./stkitm_fsamp1                            to run the program
```

The sample program directory `$CRSP_SAMPLE` contains sample make description files for Linux in *f95_samp.mkg5* for the g95 compiler. To use the `make` file, copy the relevant description file to your program directory, edit it to support the program(s) of interest and create local executables, and run with the command:

Make file:

```
> make -f f95_samp.mkg5 stkitm_fsamp1        to compile specific sample program
> make -f f95_samp.mkg5                      to compile all sample programs
> ./stkitm_fsamp1                            to run the program
```