CHICAGO BOOTH

CRSP®
Center for Research in Security Prices

CRSP Utilities & Program Libraries Release Notes

Tools for CRSPAccess Version 3.86

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

Updated July 23, 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 3.86

This release of CRSPAccess Version 3.86 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

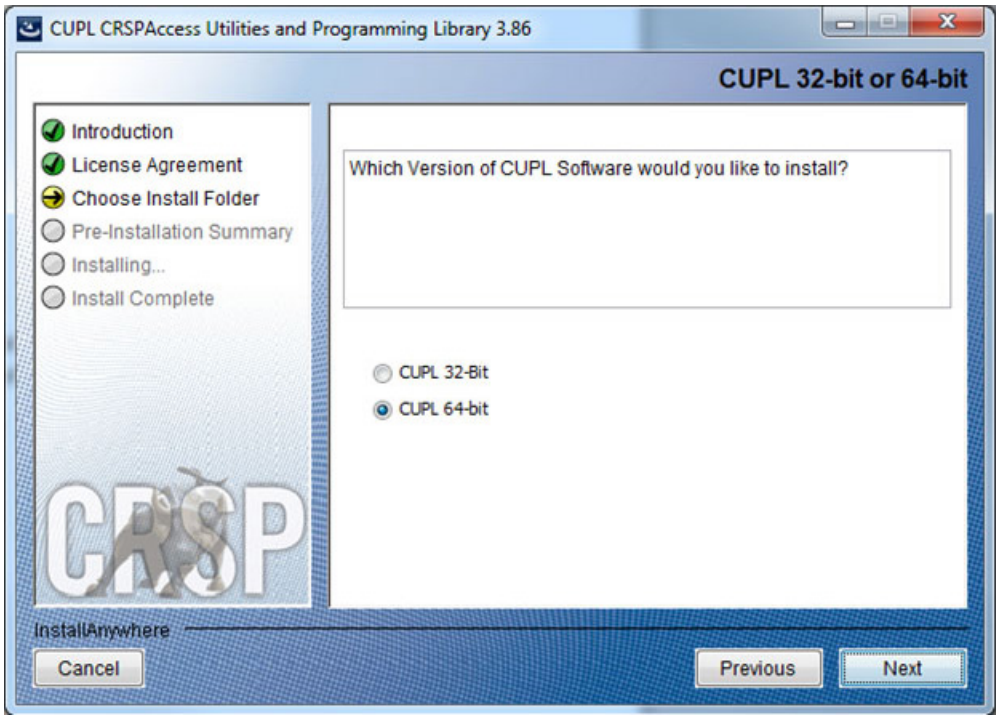
The following installations are now available:

Windows

Windows has a single executable, setupwin.exe. It automatically detects the operating system (32- or 64-bit).

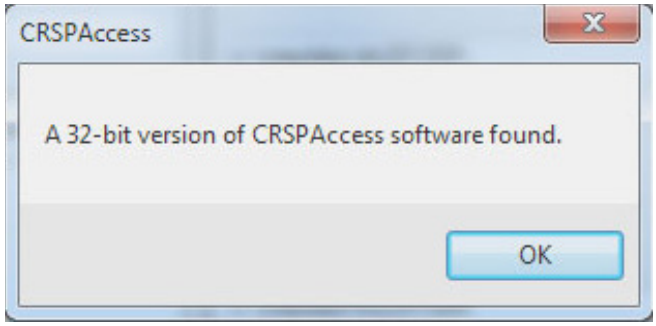
64-bit Installation

When installing on a 64-bit Windows workstation, there will be a pop-up window that allows the choice of either the 32-bit or 64-bit version of the software. By default, the 64-bit will be chosen. CRSP recommends the 64-bit version of the tools.



32-bit Installation

Click on the 32-bit button to change to that installation. If you are installing the 32-bit software on a 64-bit machine, as it completes the following message will be displayed. Click **OK** and the 32-bit version of the software is installed:



On a 32-bit workstation, it will automatically install the 32-bit version of the software.

Linux

- 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

- 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_VER386_SRD]#
```

Preparing for installation

CRSP continues to utilize the InstallAnywhere© wizard-driven installation process for CRSP software. Files are compressed and are not directly accessible until installed.

Please Note:

- *Installation over a previous version of CRSPAccess software:* CRSP strongly recommends executing one of the following two actions before installing CRSPAccess 3.86 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 3.86, 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 3.86 into a folder with the name you wish to use. For example, if you have CRSPAccess 3.22 on your computer in a folder named CRSP, first rename this folder to something such as CRSP322 or CRSP_old. When installing CRSPAccess 3.86, 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, use 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 3.86 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 platforms are clearly noted.

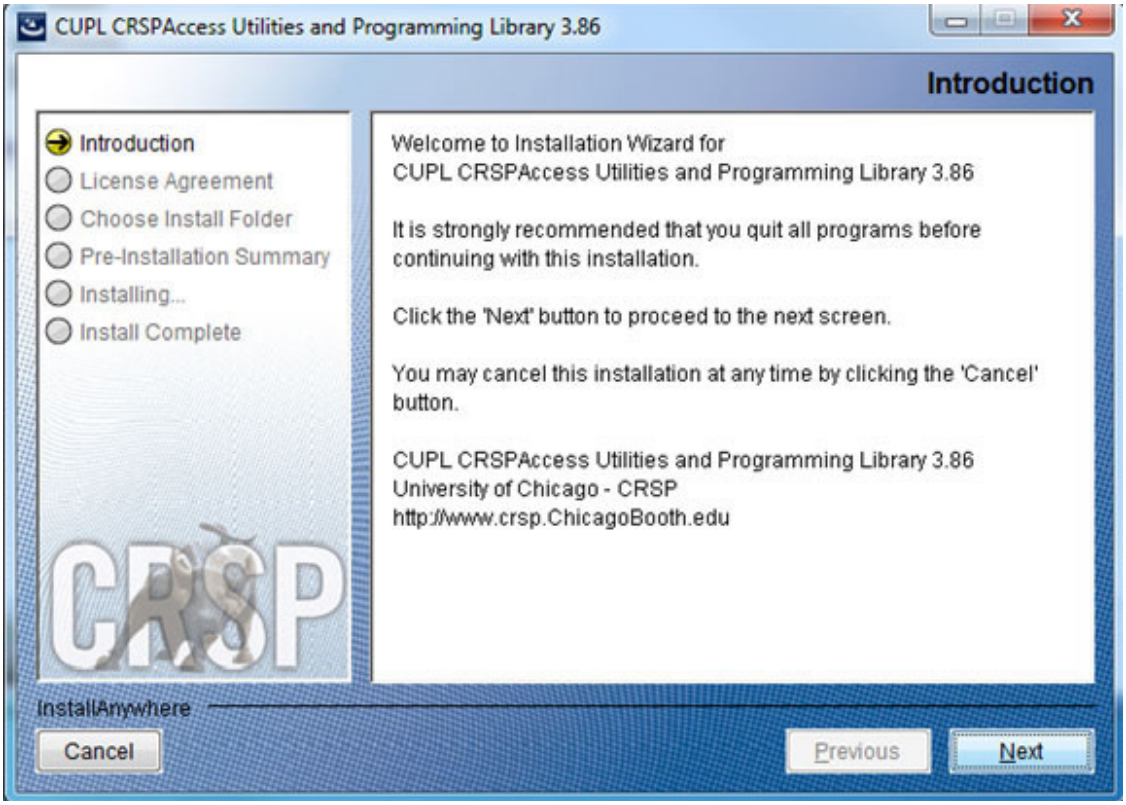
To install the CRSP Access software, download it from the MoveIT Cloud (https://crsp.moveitcloud.com/). Select the file for the platform you are using, as detailed below. The Install Splash screen will pop up on your screen and disappear. A few moments may follow before the install process begins. Once the installation initiates, you will be guided by the InstallAnywhere Wizard.

Windows:

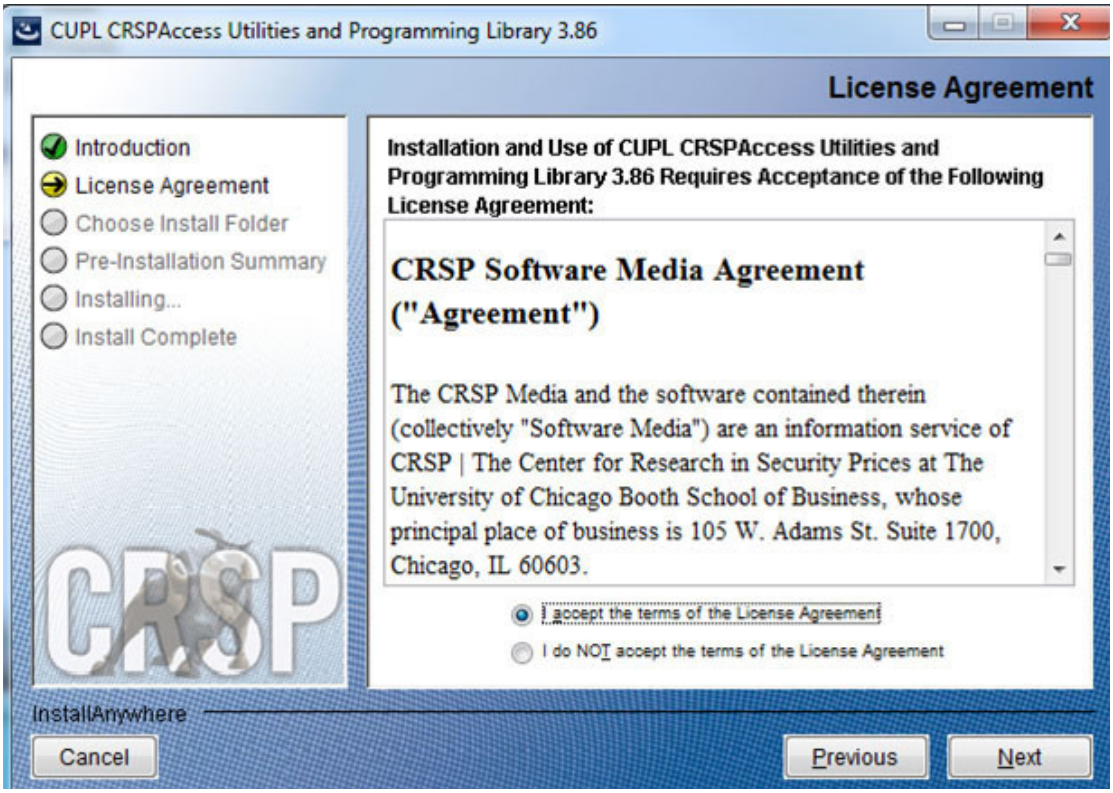
Double click on `setupwin*.exe`.

Linux:

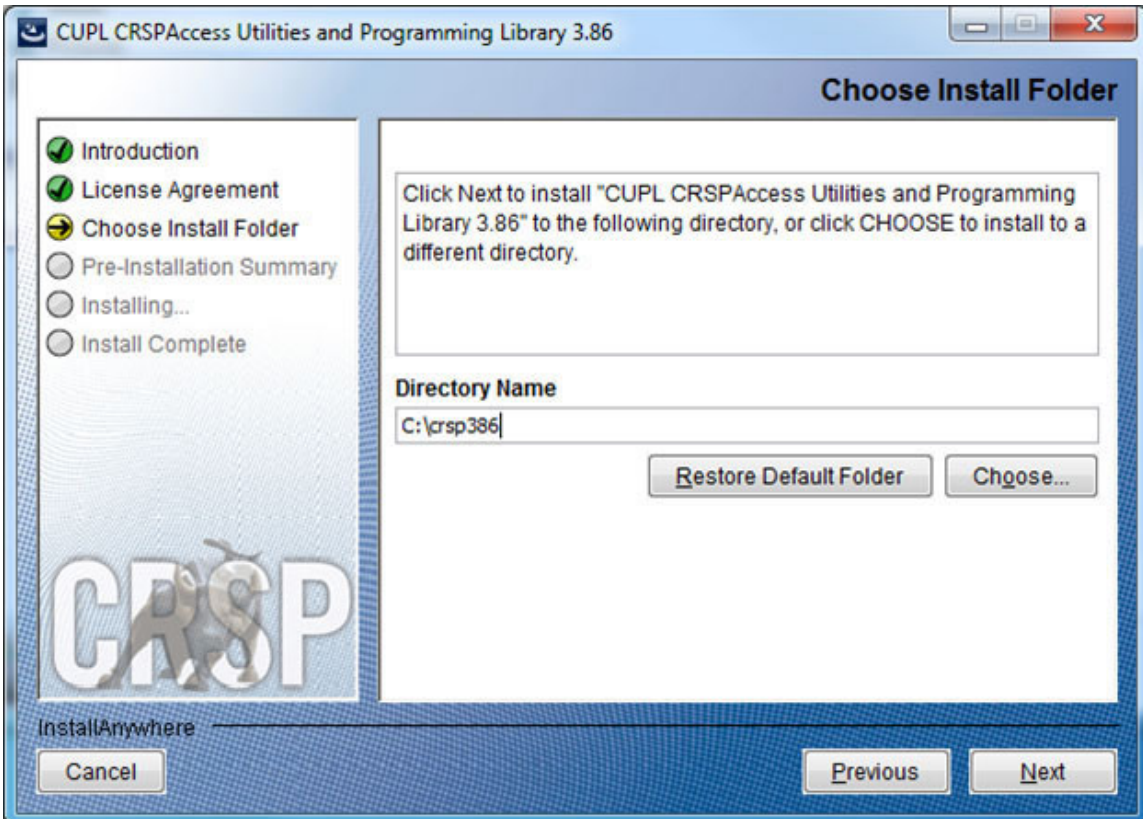
Double click on `setuplinux*.bin` to begin the installation process.



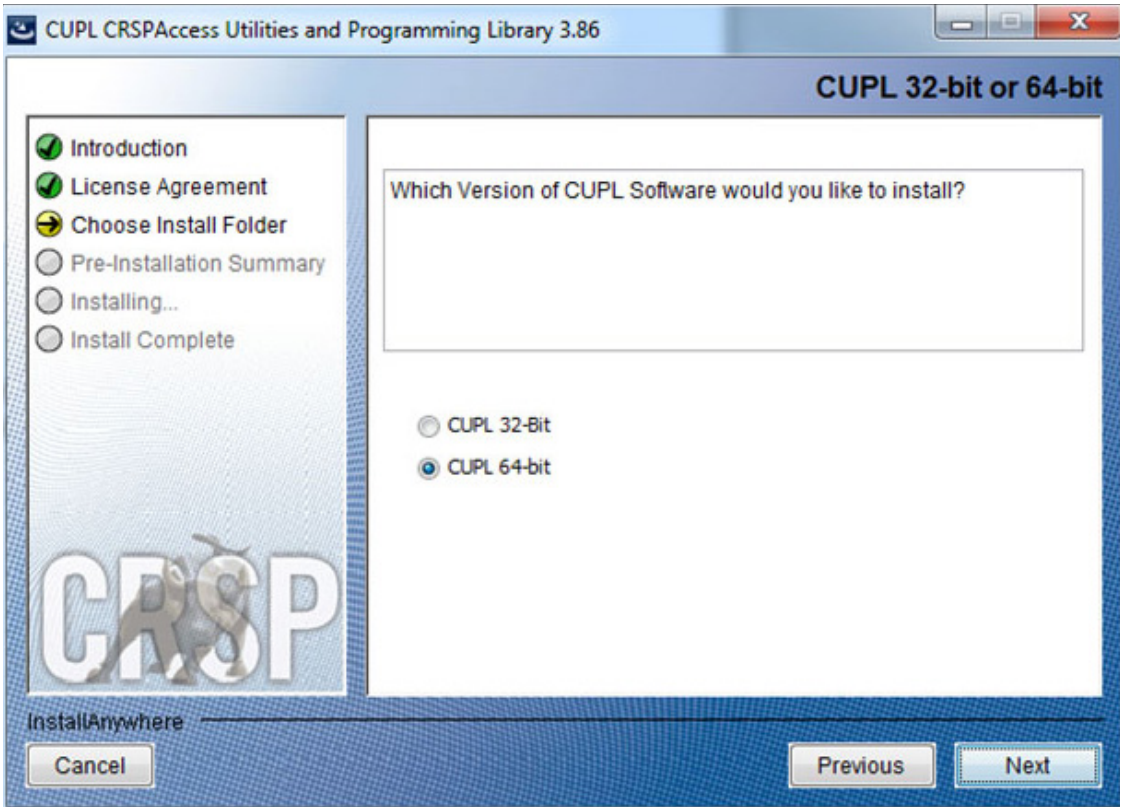
After you have clicked **Next** on the Welcome screen, scroll through and read the CRSP Software Media Agreement. Click to accept the terms of the license agreement, and assuming you do, click **Next**.



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**.



The installer will detect the appropriate software for your computer and set it as the default.

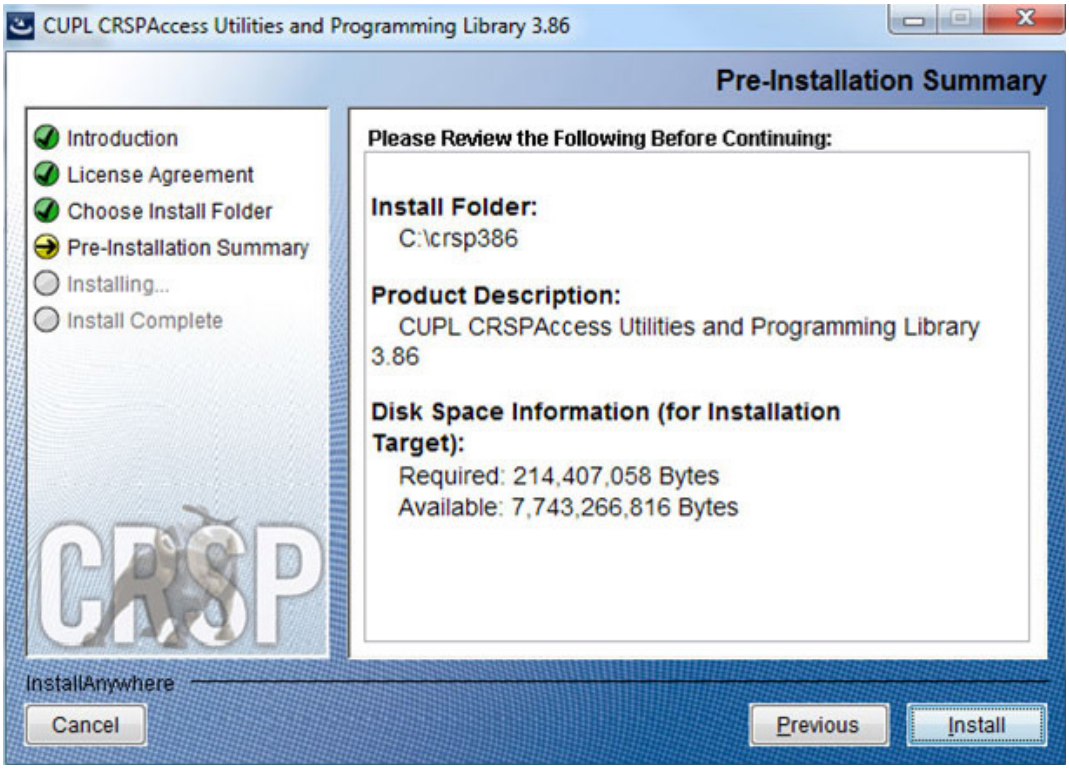


Windows:

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

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.

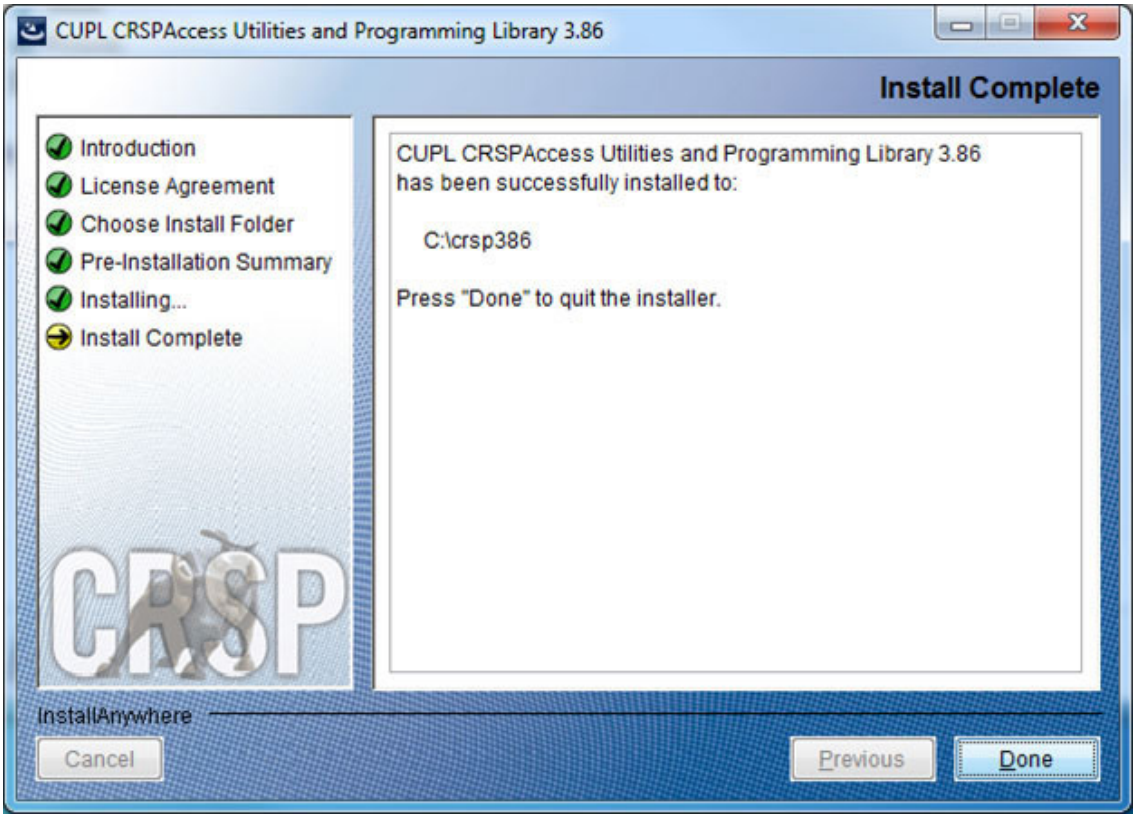


Windows:

The status of your installation will appear.

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**.



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

ENVIRONMENT VARIABLE	USAGE
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 Environment Variables

Important Change for 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 8 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.

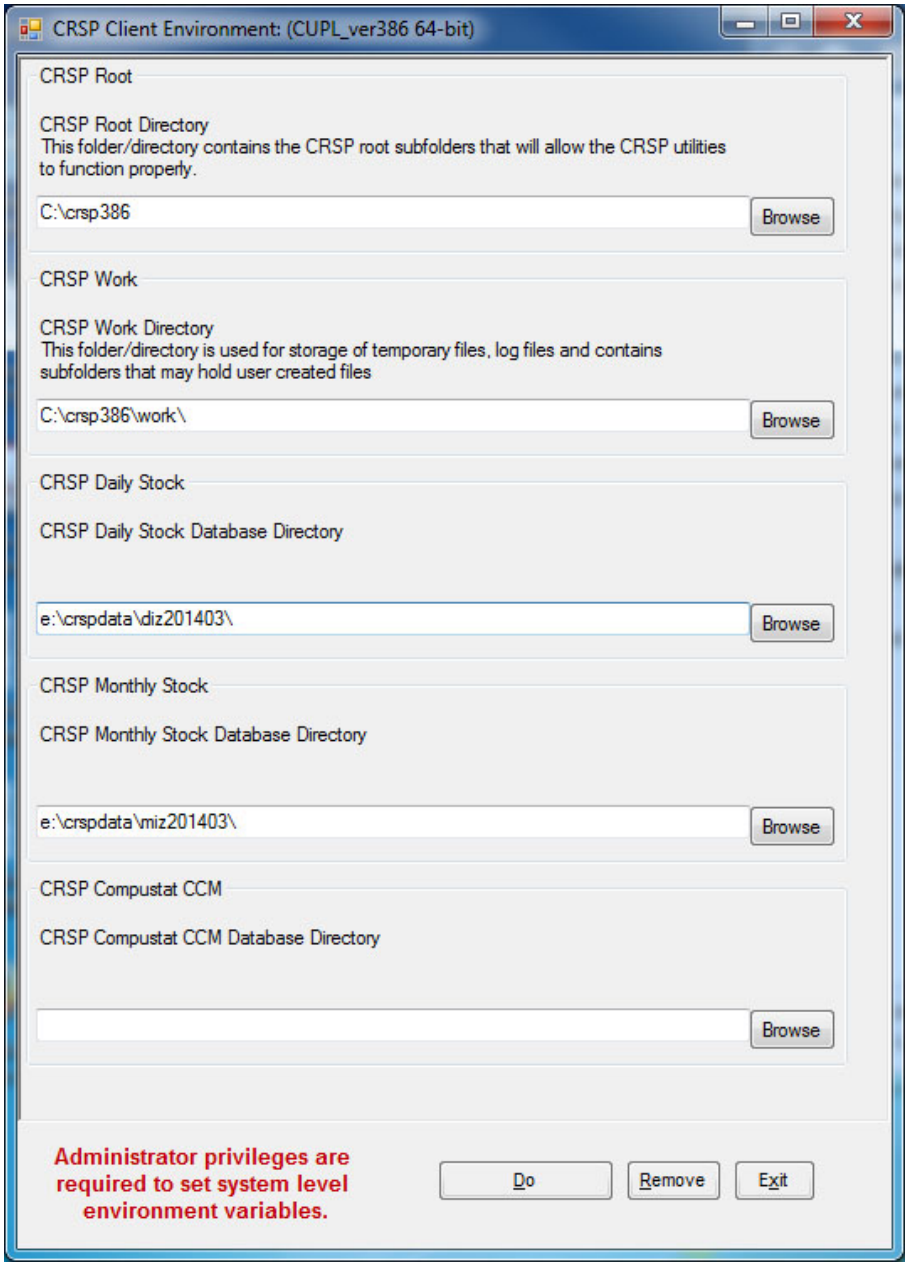
2. From the client workstation, execute the desired client install. client_environment.exe is located in the crsproot\accbin* folder and should be run from this location.

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 8. All C libraries and sample programs were compiled and tested using the Microsoft Visual Studio 2008 and 2010.

CRSP access relies on environment variables set during installation. The environment variables can also be set through the client_environment tool or with administrator privileges, through the Control Panel/System/Advanced/Environment menu on Windows 8. 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.

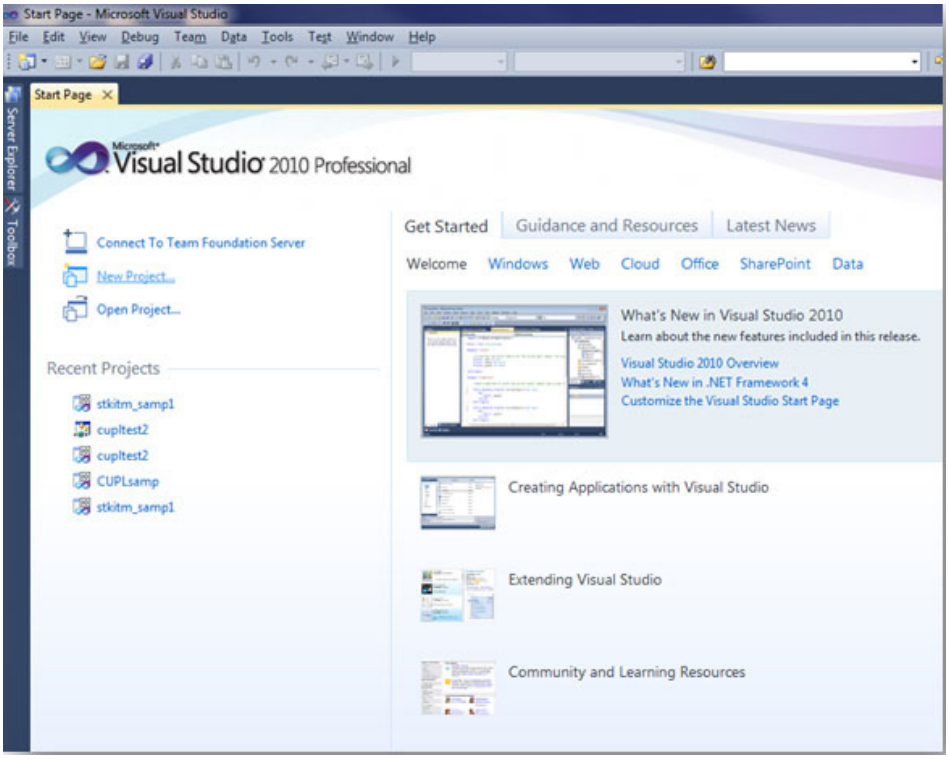
3. The %crsp_bin% folder must be in the PATH at run-time. CRSP installs do not set the PATH automatically unless running in a CRSP Command Prompt window. The user must set PATH directly under Control Panel/System/Advanced/Environment Variables, or with a SET command in the shortcut or Window prior to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

Visual Studio 2010 - C Compiler Instructions

CRSP supports compiling C programs in Windows 32-bit and 64-bit environments. The following example compiles a sample C program provided with the CUPL tools using Microsoft Visual Studio 2010. Use 64-bit options in Visual Studio with a 64-bit install of CUPL, 32-bit options with a 32-bit install.

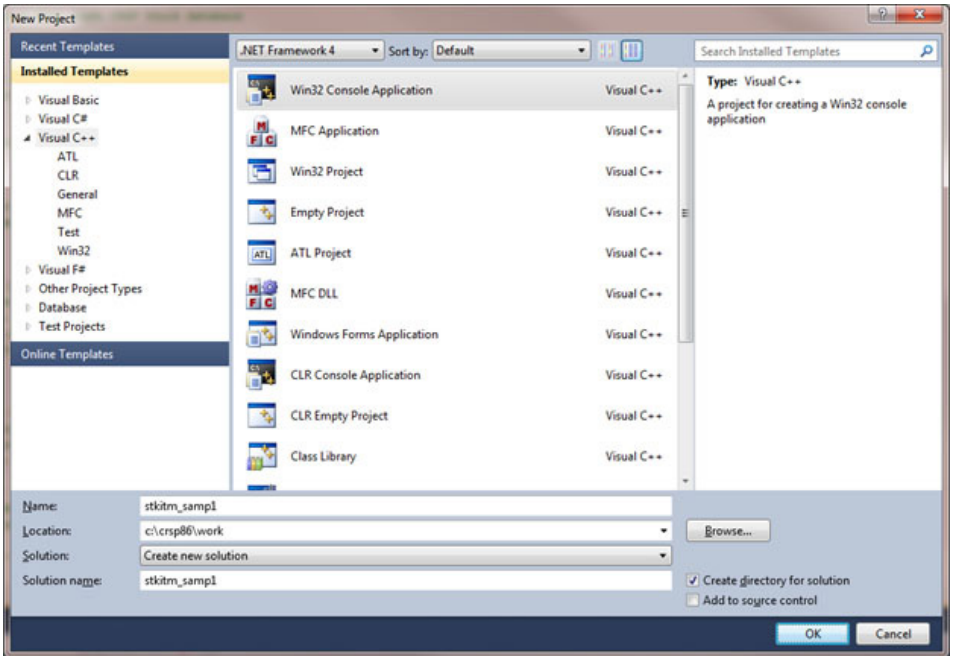
Step 1:

To begin, open Visual Studio 2010. Click on **New Project**.



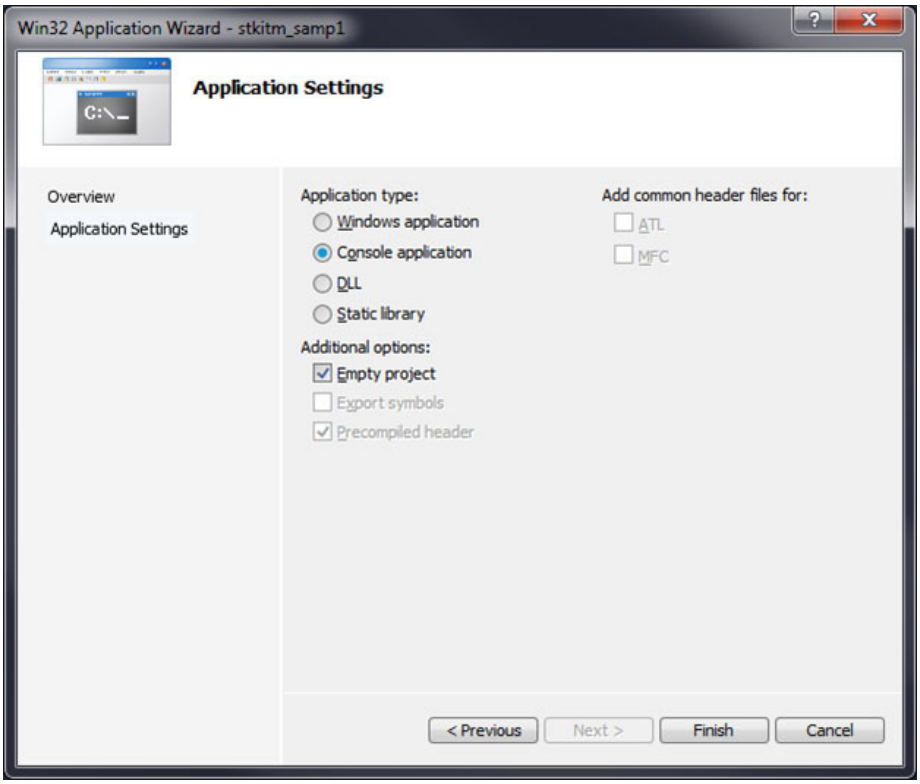
Step 2:

Select Visual C++ and highlight Win32 Console Application. Give the project a name, specify a location, and provide a solution name. Click **OK**.



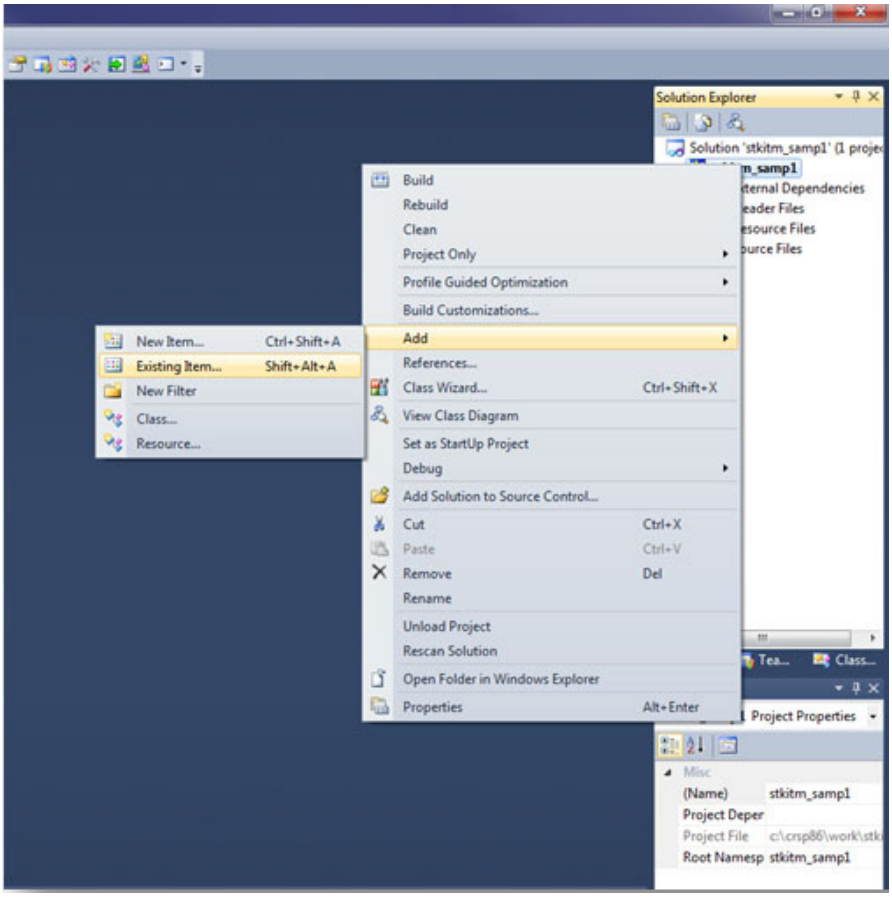
Step 3:

On the **Application Settings** screen, click on **Console** application, and check **Empty** project, then click on **Finish**.



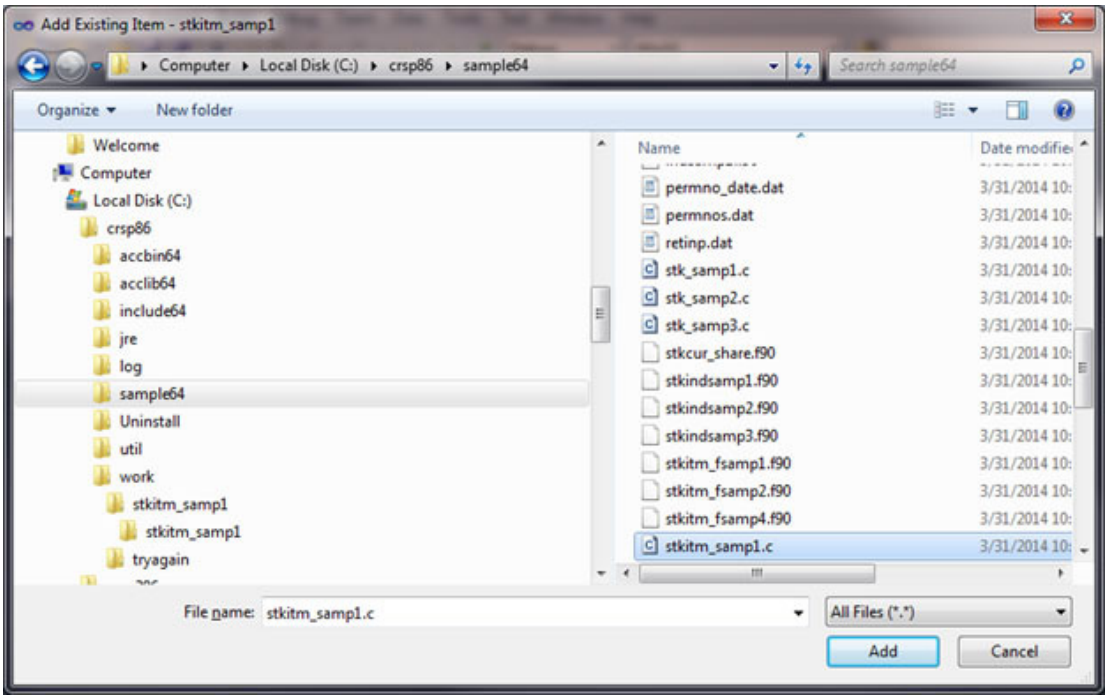
Step 4:

You are ready to add information to the project that you are building. To do so, right click on the project, in this example, `stkitm_samp1` (in bold). On the pop-up screen, select **Add > Existing Item**.



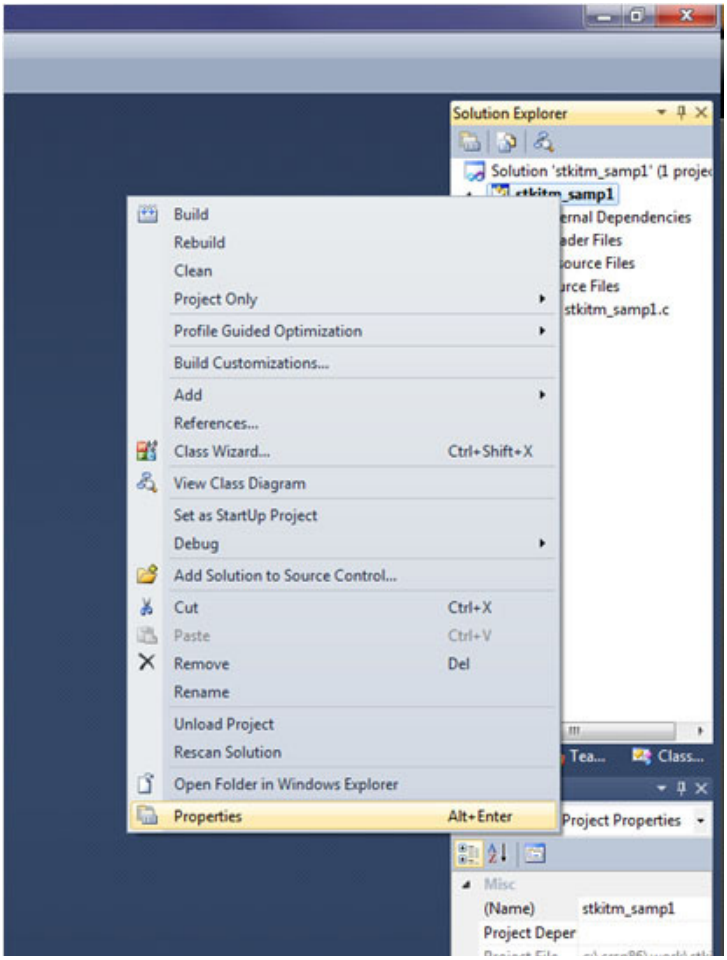
Step 5:

Browse for the program that you would like to add. In the CUPL Version 3.86 – 64-bit tools, sample programs are located in the Sample64 folder. Highlight `stkitm_samp1.c` program and **Add**.



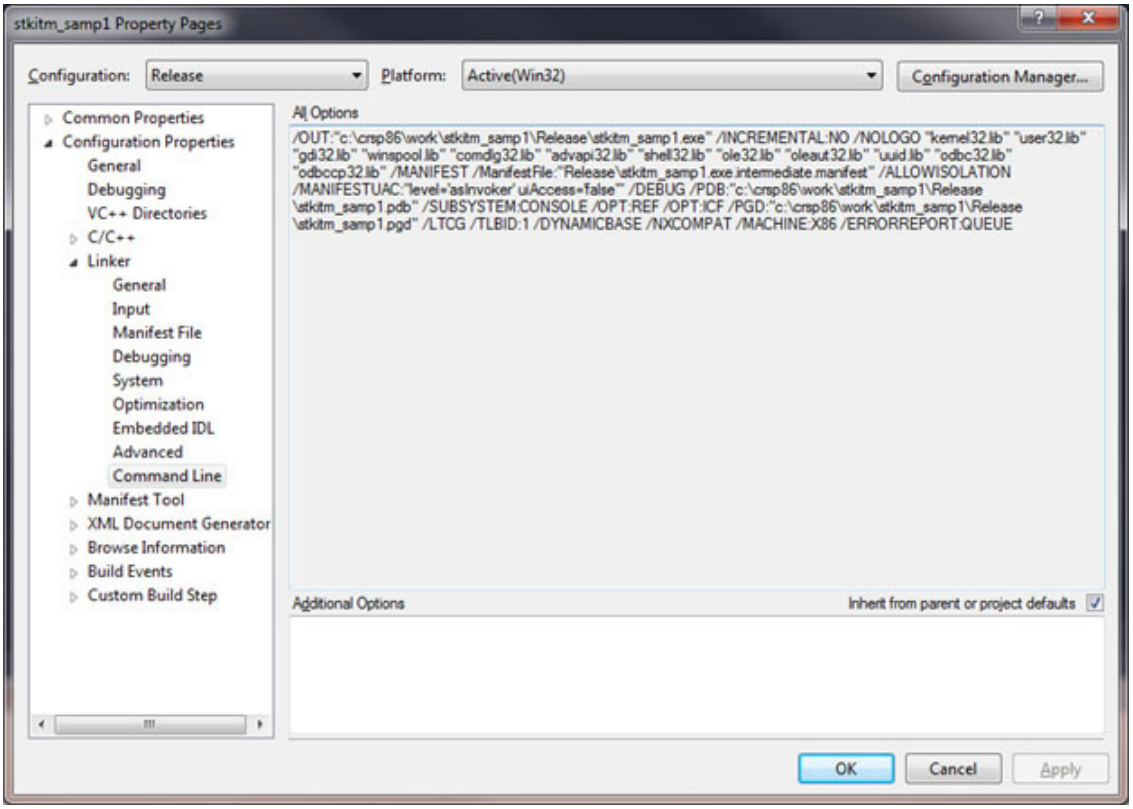
Step 6:

The program will display in the Source folder of the project. Right click on the stkitm_samp1 project again, and at the very bottom of the window, select **Properties**.



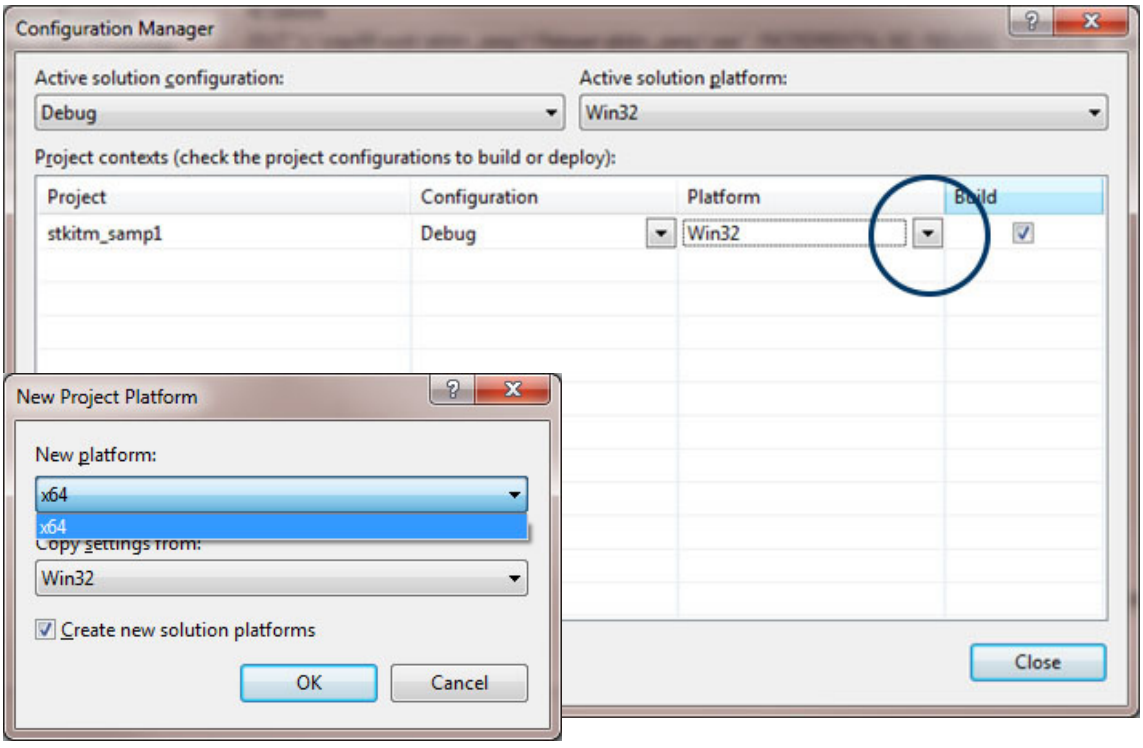
Step 7:

At this point, there are several actions to take and there is no specific order necessary. First, in the **Configuration** options in the upper left corner of the screen, click on the dropdown and select **Release**.



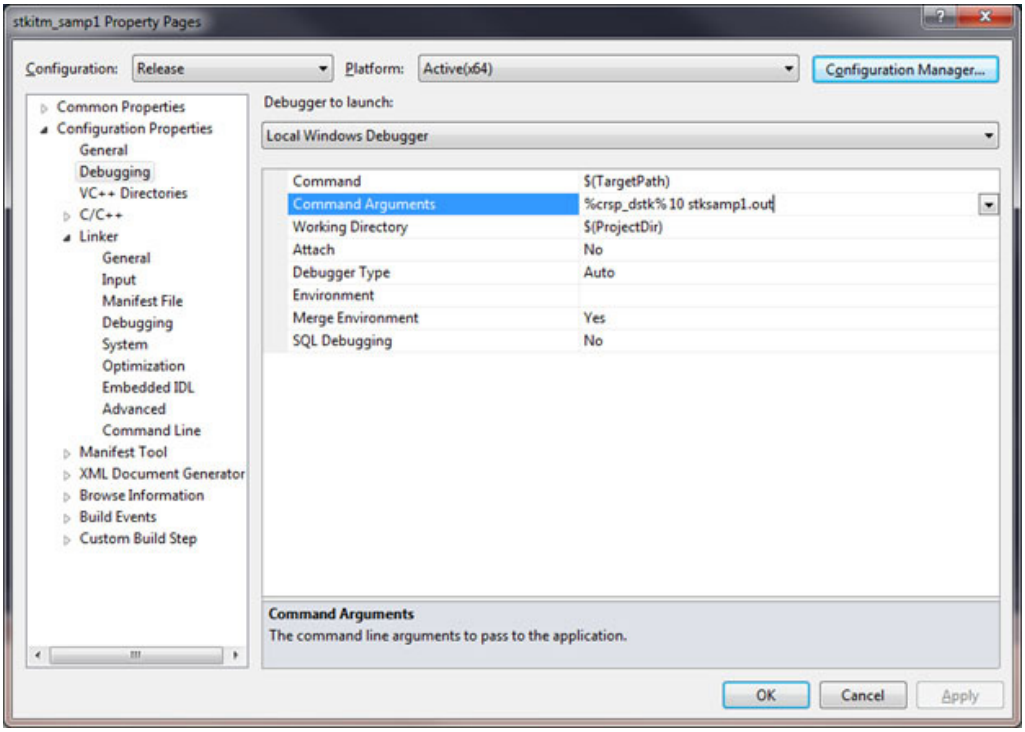
Step 8:

On the top right corner of the same screen (see above), click on the **Configuration Manager**. From the Active solution platform dropdown, select x64 and click **OK**. If x64 doesn't exist as an option, from this same dropdown click on New and add x64 as an option, click **OK**, and then **Close**.



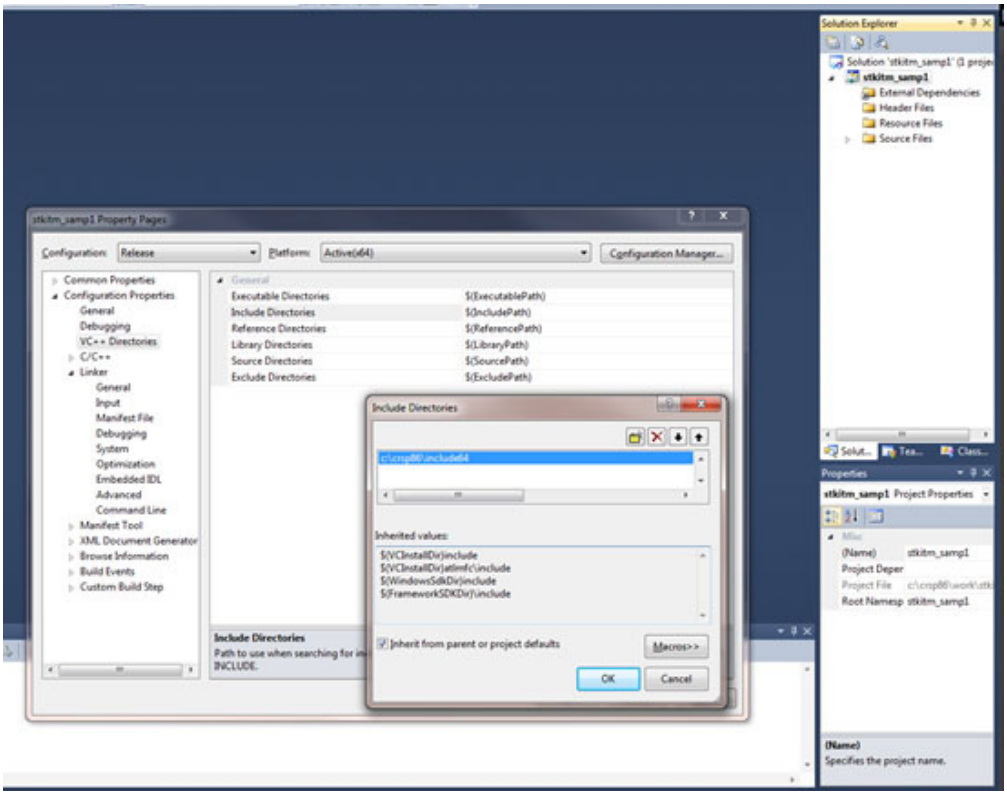
Step 9:

Back to the Property Pages, under **Configuration Properties**, click on **Debugging**. In the **Command** Arguments line, define the database that you will use, and enter a name for the output file. In this example, `%crsp_dstk%` is using environment variables that are pointing to the CRSP daily stock database. "10" is the daily stock setid. `stksamp1.out` is the file that will be generated once the project is built and run.



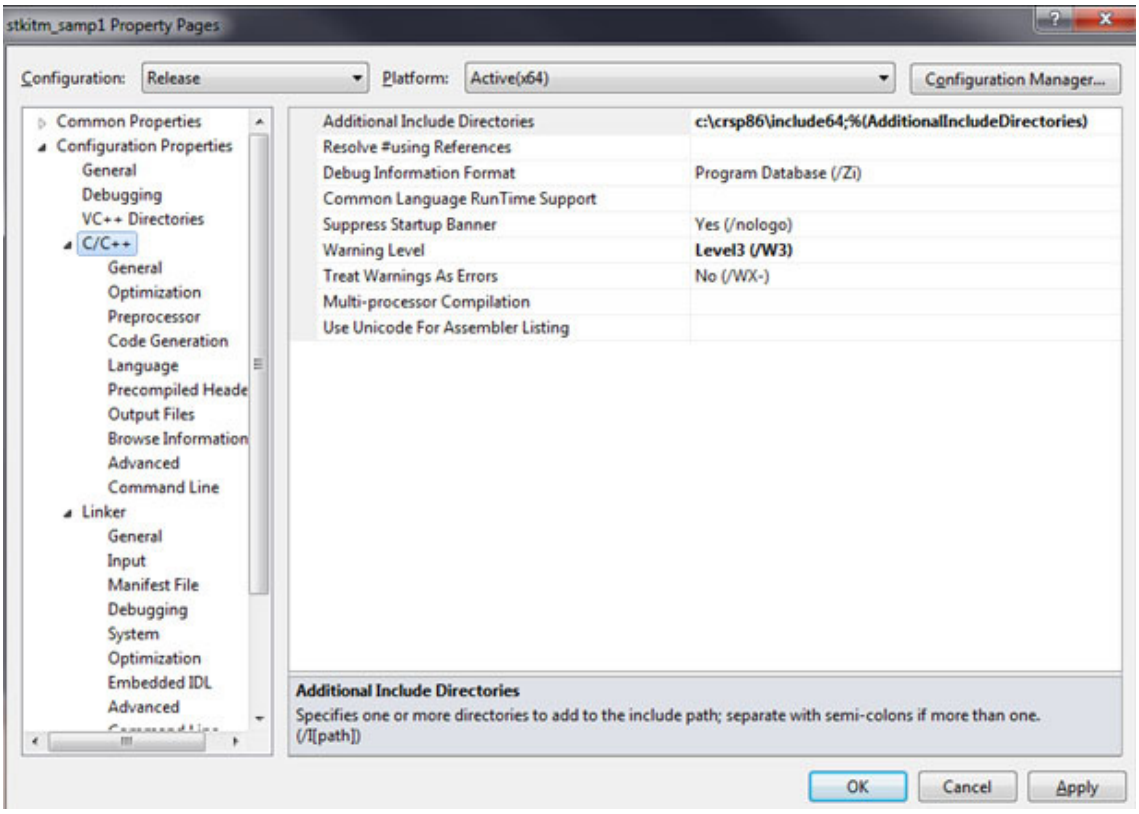
Step 10:

Still under the **Configuration Properties**, click on **VC++ Directories**. Highlight the **Include Directories** row and click on the dropdown. Click on **Edit** and add the location of the Include folder in the CUPL tools.
In this example, `c:\crsp86\include64`. Click **OK**.



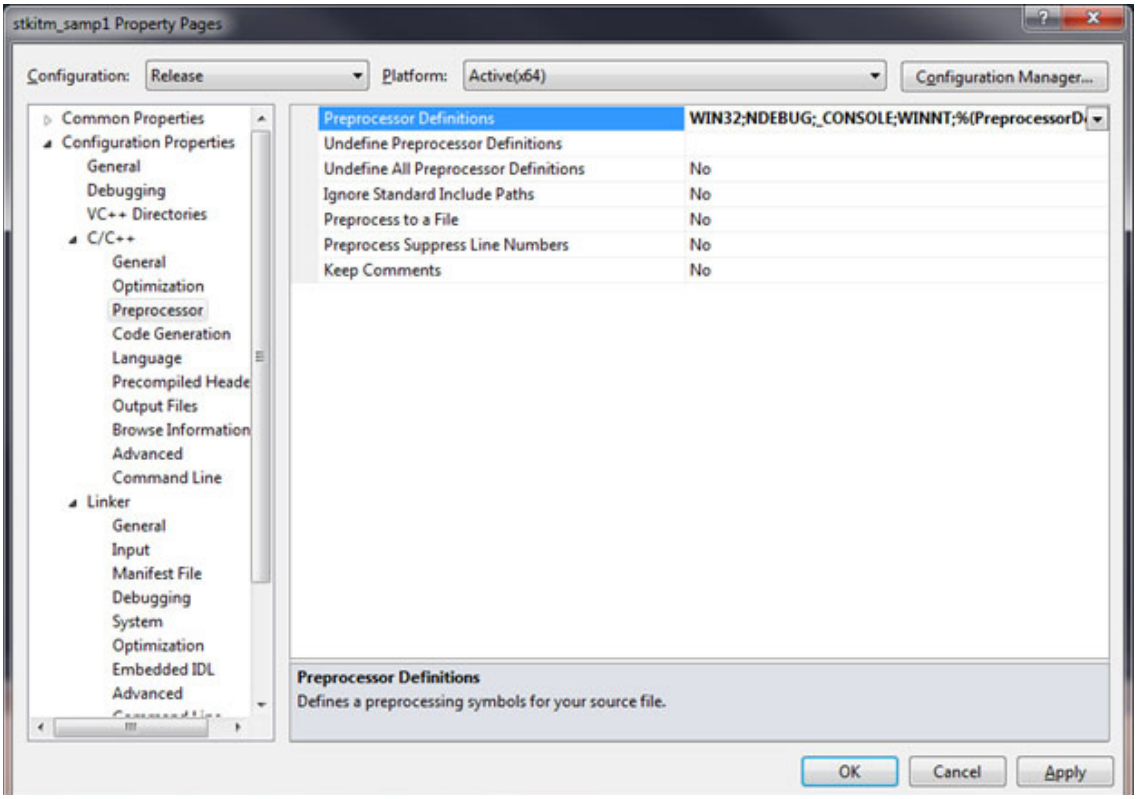
Step 11:

Next, expand the **C/C++** directory select the **General** tab. Highlight the **Additional Include Directories** and click on the dropdown and **Edit**. Enter the path for the CRSP include files. In the example, the path is `c:\crsp86\include64`. Click **OK** to close the window.



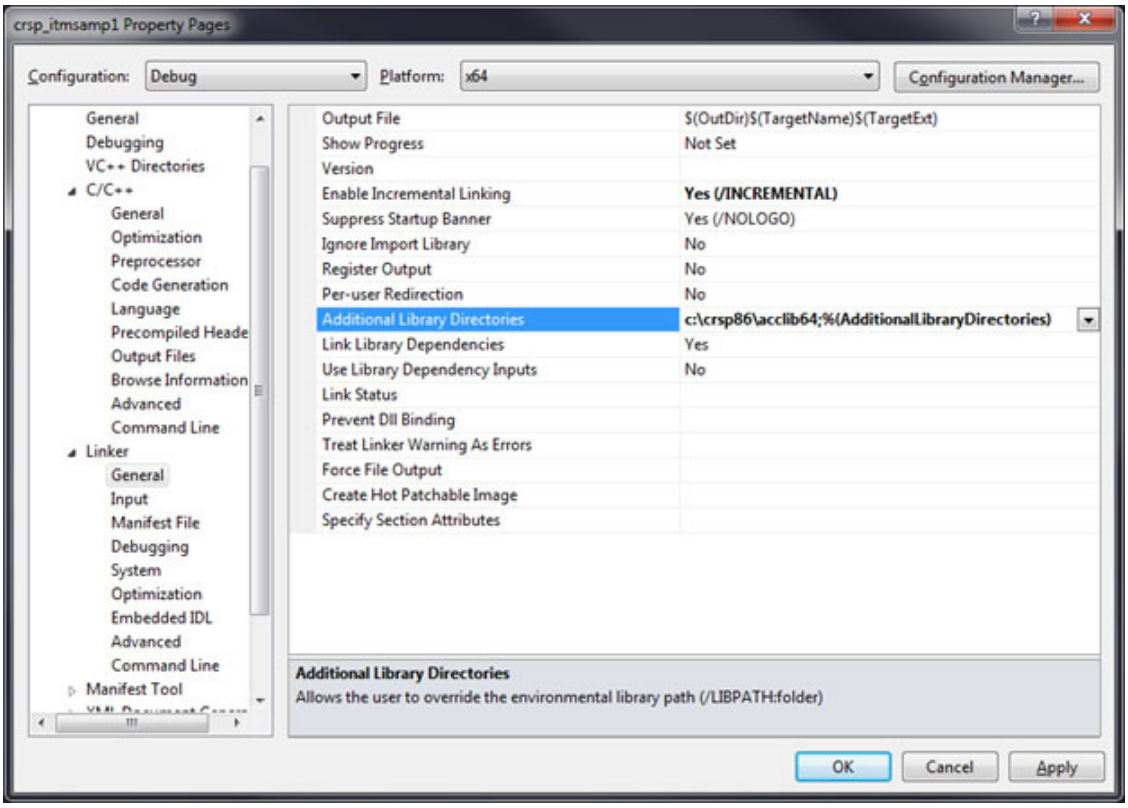
Step 12:

Still in the the **C/C++** folder, select **Preprocessor**. Highlight **Preprocessor Definitions**, click on the dropdown and **Edit**. Enter `WINNT` and click **OK** to close the window.



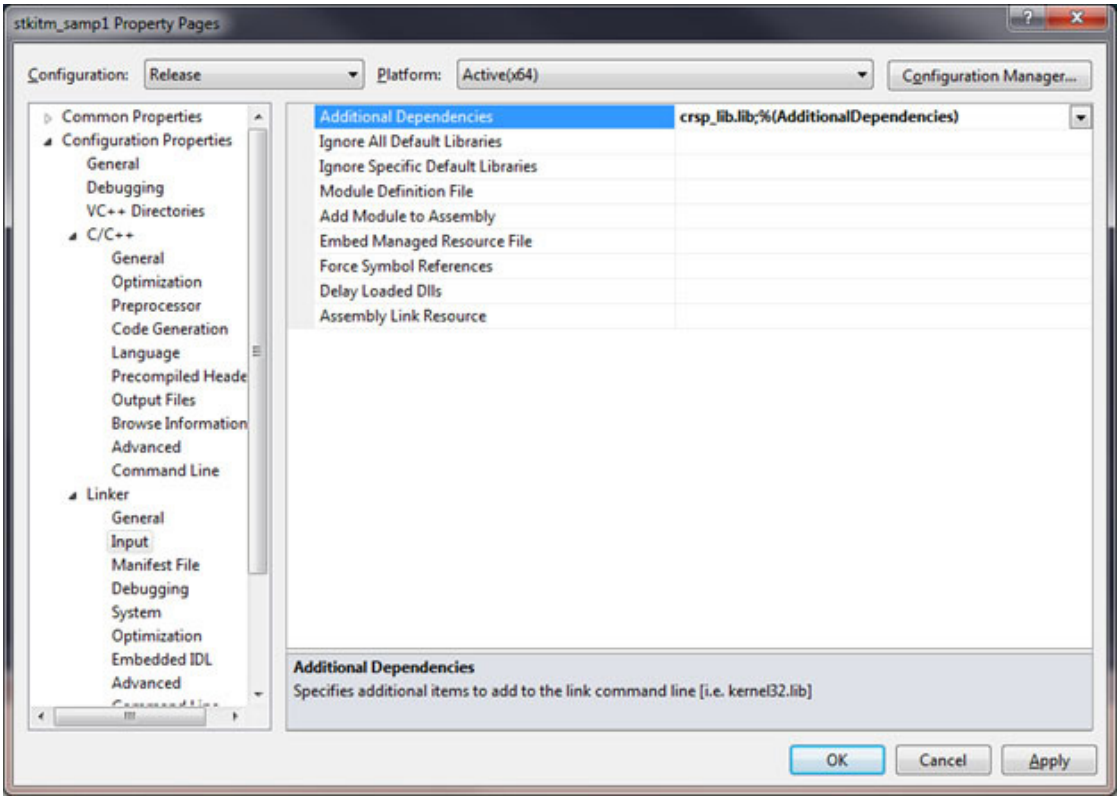
Step 13:

Next in **Configuration Properties**, expand the **Linker** folder and select **General**. Highlight the **Additional Library Directories** row and click on the dropdown. Enter the path for the CRSP libraries. In this example, it is `c:\crsp86\acclib64`.



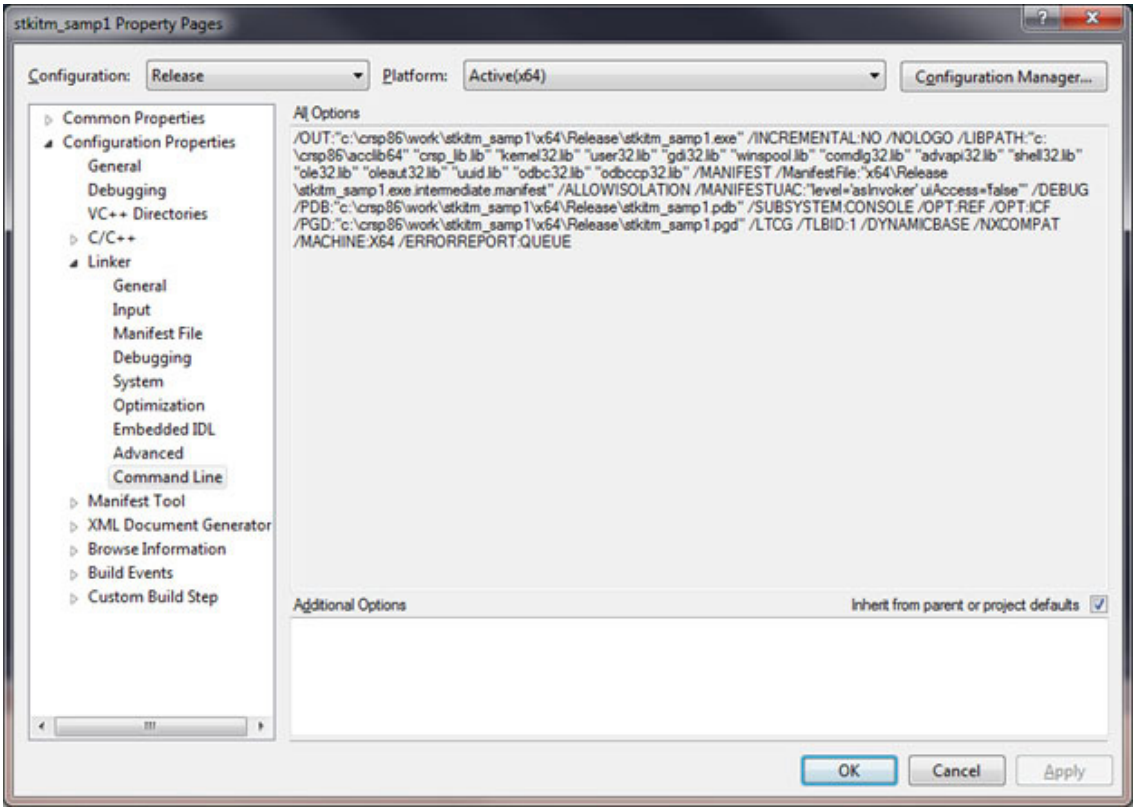
Step 14:

Stay in the **Linker** folder and select **Input**. Click on the **Additional Dependencies** row, click on the dropdown and **Edit**. Enter the CRSP library file name, `crsp_lib.lib` and click **OK** to close the window.



Step 15:

Finally, within **Linker**, select **Command Line** and click **Apply** in the lower right corner of the screen. Click **OK** to close the Properties Pages.

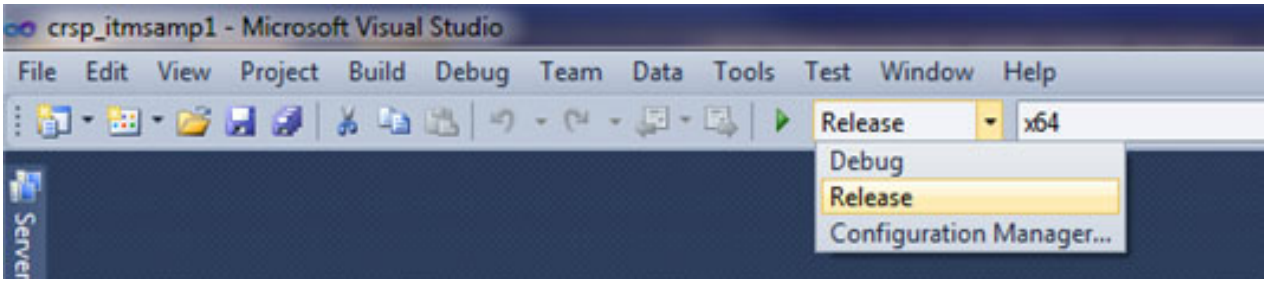


Step 16:

At this point, all entries should have been made in order to build the solution. From the menu bar, select **Build** > **Build stkitm_samp1**. Assuming that the build runs successfully to completion, you ill see the following message once the build is complete:

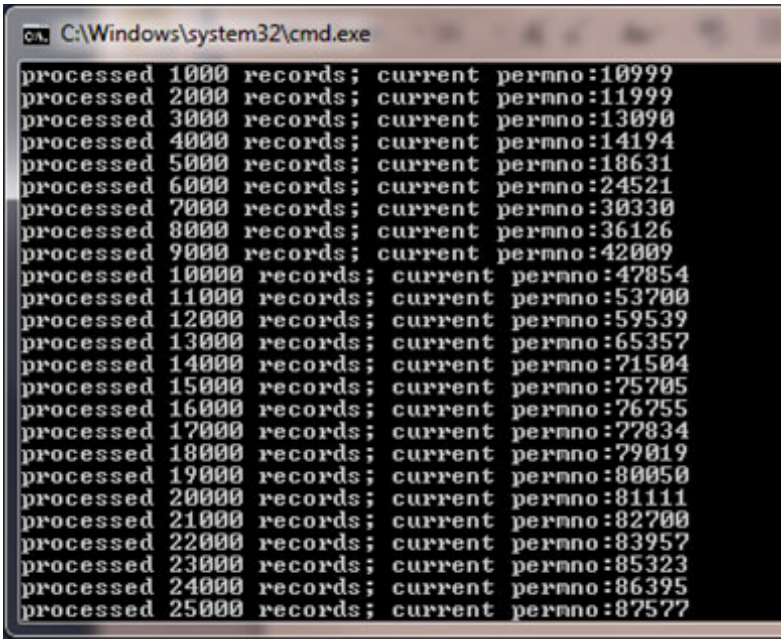
Build: 1 succeeded, 0 failed, 0 skipped.

Prior to running your program, check the Visual Studio Menu bar to confirm that the Solution Configurations set the mode to **Release**. (*Note: At CRSP, if not set to Release mode, we encountered an error message stating that MSVCR100.dll is not found*)

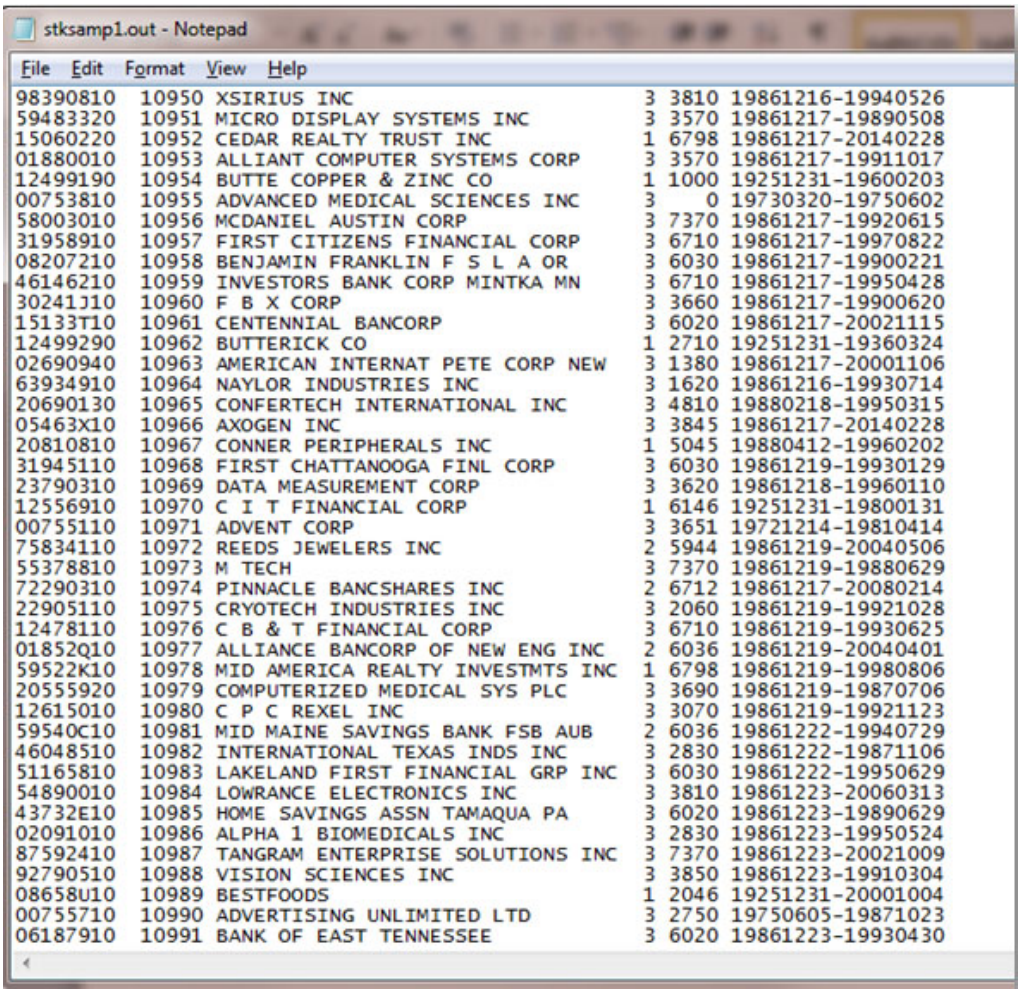


Step 17:

Once you have built your program successfully, you can now run it to generate output. From the Menu Bar, click on **Debug** > S**tart Without Debugging**. The program will begin running and for this example, will work sequentially through the universe of CRSP PERMNOs.



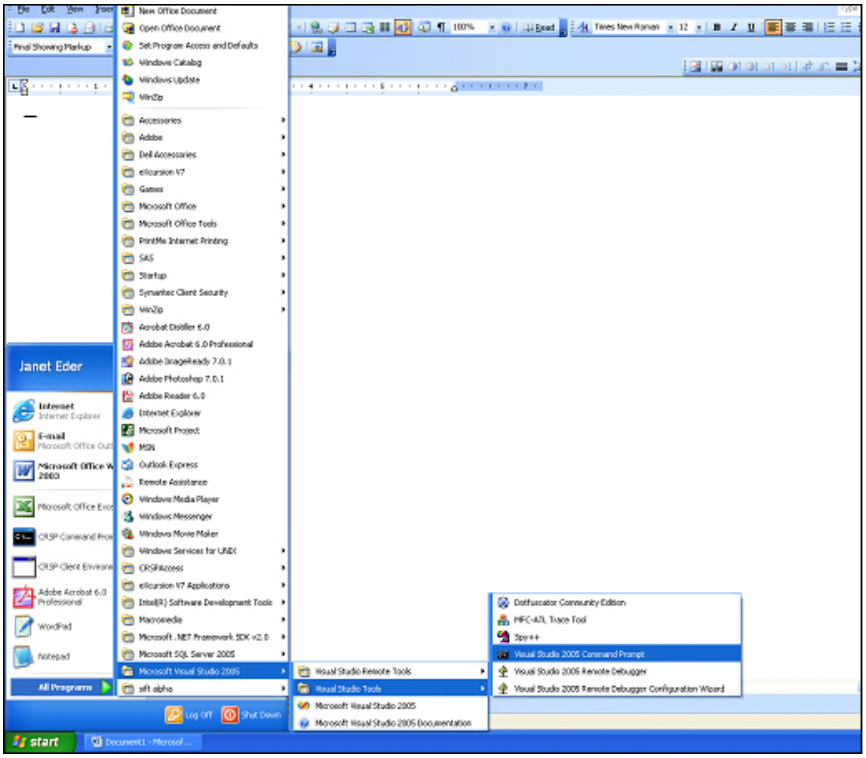
Your output will be located in `c:\CRSP86\work`, or as specified in your project.



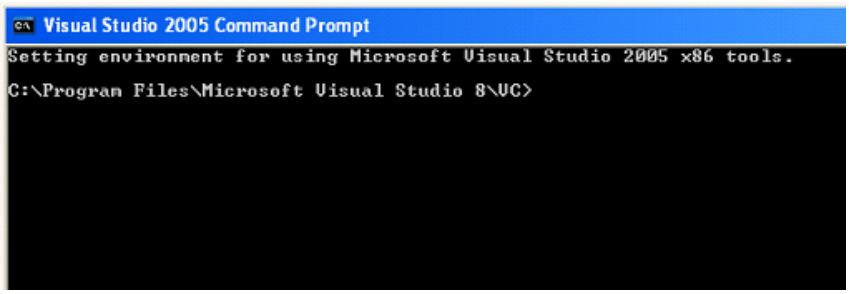
Using the Command Prompt Window

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
```

Linux Systems

CRSP supports C programming for Linux Red Hat Enterprise Linux 5.10 on Intel x86 32-bit and 64-bit machines. C functions were compiled and tested using the gcc 3.2.3 compiler on the 32-bit computer, and gcc 4.1.2 on the 64-bit.

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