CSDA Code Utility

for Alpha Five

Created by

Computer Systems Design & Associates
61 Cheryl Lane
Waltham, MA 02451-2185

www.csda1.com

Release Version 2.68
Last Update November 29, 2007

CSDA Code Utility for Alpha Five

Table of Contents

CSDA Code Utility for Alpha Five

Why use the CSDA Code Utility?

(Or, how much time have I wasted repetitively doing these steps?)

- During the development process of Alpha Five, sometimes you are working on different set of problems, requiring editing of different forms, code and other windows.

  With the CSDA Code Utility, and saving various desktop configurations, you can easily switch quickly between them and pick up editing of that problem where you last left it. And when Client A calls while working on Client B's database application, you can quickly switch to Client A's database application ready to help.

- Various problems often come up in development of an application that requires closing the database (and leaving Alpha Five open) or closing Alpha Five. These include tables that are left open by code that errors, Global variables that are not cleared, certain DLL operations like DECLARE and DECLARESTRUCT that need to have Alpha Five restarted if any changes are made, Alpha Five errors, etc.

  With the CSDA Code Utility, restarting the database or Alpha Five will clear these problems, and it takes a far shorter time to restore the Alpha Five Desktop and continue development than doing this manually

- When you shut down your computer for the day, the next time you boot up, it is not always easy to remember what scripts/UDF's you were editing, or what you were doing the previous time (*Don't you just love those 3am in the morning programming sessions?*)

  With the CSDA Code Utility, it automatically remembers the last configuration of a database when exiting, and restoring the desktop is automatic.

- You have a piece of code that runs slowly. You make a change, but was it faster? The Alpha Five profiler indicates no difference, so you set variables to the time at key points in order to determine execution time, but the results are inconclusive. As it turns out, the resolution of time is limited to approximately 1/20th of a second (0.05 seconds), and can easily be off by as much as 1/10th of a second. If the code time being measured is not large compared to 1/10th of a second, you get un-meaningful results. You could loop the code many times and measure the total time divided by the loop count to get more accuracy, but what count value should you use? Try to iteratively determine the minimum iteration count manually and more test time is wasted.

  With the CSDA Code Utility, it very quickly determines the right iterative count and then times it accurately.

- Testing your scripts is fast by pressing the "lightning bolt" tool bar button or double-clicking on the script name in the control panel, but UDF's require entering a line of code to execute in the interactive window to execute.

  With the CSDA Code Utility, enter the parameters (which are remembered) and run the UDF as needed. It returns the type, value, execution time as well as reports errors.

- During development, you make a change somewhere that makes the application inoperable. Too bad you didn't save the application before making changes. What if you had a single button that quickly backed up the Data Dictionaries (and optionally the data) that would allow you to go back to a previously saved version?

  With the CSDA Code Utility, just click a button and you saved the entire application quickly.

- You have been working on a client's project, but in order to be paid, you need to keep records of work being performed. Wouldn't it be nice to automatically log what you are doing?

  With the CSDA Code Utility, an automatic log records the open, saves and closing of the database applications and the windows that were open to help document your progress. And you can have it prompt you for optional comments too!

- You are debugging Xbasic code. You insert a Debug(1) code to turn it on at the desired point in your code. You save the code. You run the code and debug. Now you go back in, change the debug point, and save, run and debug again.

  With the CSDA Code Utility, just select desired debug area, and debug and run the code with just 1 click!

Use the CSDA Code Utility! Isn't it time you stop wasting your time?

CSDA Code Utility for Alpha Five

Alpha Five Version Compatibility

The CSDA Code Utility works with Alpha Five Version 6 and up.

Vista Compatibility

CSDA Code Utility version 2.00 and up is compatible with Microsoft's Vista windows operating system and Alpha Five Version 8.

Purpose

The CSDA Code Utility has several time-saving features for the Alpha Five development environment and is intended for novice and advanced developers. The standard version includes:

- Save and Restore Alpha Five Desktops – This remembers each Alpha Five session window, whether it is in view or design modes, the Code Editor tabs and all text in the Interactive window, and the cursor position of all Code Editor tabs. It also remembers the current record being displayed for each open form or browse. You can then close the Database or Alpha Five and do a restore of all windows automatically when you open the database again. You can also save alternative desktops for each database to allow different A5 window setups for various needs.
- Provides ***quick*** backup of data dictionaries, and optionally, data (even if currently being shared). This allows you to restore an earlier code version in case of corruption or other problem encountered.
- Navigation in a round-robin style to next and previous A5 windows

The Pro version also has the following features

- Provides code timing (more accurate than profiling), quick debugging and other support for the code editor
- Creates a log of Alpha Five development (useful for work documentation) – Provides database logging with time/date stamp of open, saves and closings of the A5 Desktops with A5 windows and code editor tab lists and optional comments. In addition, you can also set regular logging at user-defined intervals.
- Code operations – Allows quick global searches of code, removing and adding debug statements etc.
- Code Jump – This allows opening and jumping to various development screens or points in Xbasic code with 1 click

Trial Version

When operating in Trial mode, you will be using the Pro version. The trial version has an expiration date but is not limited (crippled) in any way. However, if the license cannot be validated, or if the trial has expired, it will show the main dialog box with "Limited Mode", and will not operate any features except for Help.

When using a trial license #,
you must allow access to the Internet or it will not operate.

To operate a trial version of the CSDA Code Utility for Alpha Five, you must do 1 of the following:

- Operate Alpha Five in it's trial mode (the CSDA Code Utility will have the same expiration date as the Alpha Five trial)
- Request a free trial license # for the CSDA Code Utility at the web site – **http://www.csda1.com/csda_codeutility/CSDA_CodeUtility.html**

Licensing

Each license number of the CSDA Code Utility is licensed for at most 1 concurrent (simultaneous) user of the product, and may be legally loaded onto, at most, 3 different computers (intended for home, work and portable) systems. The computer's name must be the same in all cases. It will not permit simultaneous use of the same license over a network.

CSDA Code Utility for Alpha Five

Quick Installation

If installing the CSDA Code Utility using the executable EXE installation file, just run it and follow the instructions. Check each version of Alpha 5 that you wish to install the utility for. Each version of Alpha Five selected will open, do the installation and then exit Alpha Five.

Manual Installation

To install or upgrade the CSDA Code Utility manually, **unzip** the downloaded file into any directory of your choosing. If you are unfamiliar with the way to unzip a file, navigate using windows explorer or similar to the downloaded program. Right click on the downloaded filename and select Extract All (which is the windows default) or other Zip/Archive program you may have installed in windows.

In the same folder that was unzipped is an acrobat / PDF documentation file on using the product.

Start the Alpha Five (version 6 or later) that you wish to install to. Select Alpha Five's **File**, **Open Database….** Navigate to the folder you unzipped to, select the **A5DesktopInstall.adb** database file then click **open**. (If you don't know where this file is, from the Windows start button, do a **Start**, **Search**, **For Files and Folders…** (Win XP or 2K) or **Find** (Win ME or 98). Enter the filename, **A5DesktopInstall.adb** and start the search)

When **A5DesktopInstall.adb** starts, it will automatically install the **CSDA_CodeUtility.AEX** in the **addins_installed** subdirectory (version 7 and earlier is normally "C:\Program Files\A5Vx\addins_installed"). After installation, the program will exit Alpha Five. The next time you start Alpha Five, **use 1 of the shortcuts created by the program to start the CSDA Code Utility.**

Repeat for each version of Alpha Five Version 6 and later present on your system and that your license(s) is valid for.

It will also install 2 shortcuts on the windows desktop for each version:

- Alternative to the standard Alpha Five shortcut. When a database is opened via toolbars or menus, the CSDA Code Utility starts and the selected database is opened.

- Starts the CSDA Code Utility, and restores the A5 Desktop of the last used database.

<u>If the CSDA Code Utility for Alpha Five does not seem to be working</u>

Some earlier releases of Alpha Five Version 8 may not properly install Addin programs (of which the CSDA Code Utility for Alpha Five is one) properly. If you are experiencing any operation problems with the CSDA Code Utility for Alpha Five, try this after installation. While the Alpha Five Control Panel is selected, select **Tools**, **Addin Manager…** from the menu. Now unselect one or more items, and click **OK**. Repeat, this time re-selecting the ones you previously unselected, and click **OK**. Close Alpha Five and restart with one of the A5 Desktop shortcuts.

If the program still has not started, start Alpha Five, then go to the Alpha Five Control Panel. Select the Interactive Window button on the Alpha Five toolbar above it. Now type in the Interactive Window the following

```
A5desktop()
```

and then press Enter. If any error is displayed, send the error message and information to support@csda1.com

CSDA Code Utility for Alpha Five

Starting the CSDA Code Utility

To start the CSDA Code Utility, just start Alpha Five using the A5Vx A5Desktop Restore or A5Vx Last A5Desktop Restore icons on the windows desktop instead of the normal Alpha Five icon. Now open any database and the utility will start.

Alternate Start of the CSDA Code Utility

An alternative way to start the utility is to use the Control Panel's Addin submenu choice or Right click on the Control Panel, or the Code Editor's Addin submenu. Other key functions (such as Code Jump, A5 and DB Restart, Quick Backup and Code Edit) can also be invoked from the addin menus

Creating Specific Shortcuts for Databases

You can also generate additional database-specific shortcuts as well as regenerate the original A5Desktop icons should they be deleted from within the utility.

If you wish to start a specific database with the A5 Desktop, **you cannot use a standard Alpha Five generated shortcut** without putting the correct target line in the shortcut property, which can be difficult. To easily do this, use the "Create Shortcut" button on the "Rolldown'ed" menu dialog box to create the current database shortcut on the desktop.

Shortcuts placed on the desktop can be moved to any directory, such as the Windows Start Menu, after creation.

Running without the CSDA Code Utility

To use Alpha Five without the CSDA Code Utility, use the normal start of Alpha Five from the original Alpha Five shortcuts or Start, Programs….

Uninstalling the CSDA Code Utility

You should never have to uninstall the CSDA Code Utility. If you run using the standard Alpha Five icon, it will not be running (although the code functions are available to be executed).

However, if you do want to remove it, or want to verify that an extreme problem is not caused by the utility, the method is as follows. First, close Alpha Five. Now using Windows file search ability, search for all **CSDA_CodeUtility.aex** files (the ones located in the affected Alpha Five directories) and delete them. Repeat, searching for and deleting the **Addin_library.aex** files (which are recreated if it does not exist when you start Alpha Five).

All the other files that are created by the CSDA Code Utility are used only by the CSDA Code Utility and will have no affect on Alpha Five's operation. For documentation of the other files used by the CSDA Code Utility, see the section "Files Used" in this documentation

Entering Licenses

When running for the 1st time or anytime there is no valid license file or license #, it will prompt you for a license file path and/or License number. Alternatively, you can edit the license file by pressing the utility's **Help** button, then pressing the **Edit License** button at the bottom of the help page.

Enter in the complete license line (as one complete line), or enter just the license # part (after the 1st comma and before the 1st space).

If you have many license numbers to insert, you can press the "Edit License File" button and enter additional license numbers. The License file can contain multiple licenses from 1 or more companies and 1 or more products. Each product will find only the applicable license line for its needs. Each line of the license file contains a company ID, license number and optionally a comment after a single quote, e.g.

CSDA,1234-5678-9012-3456-1234-5678-9012-3456 ‘ Comment describing what license is for

Note: There should be no spaces between the company name, comma and license #

If entered via the utility's Help button, you will have to close Alpha Five and restart Alpha Five with the utility's desktop icons (See Using the CSDA Code Utility section) for the new license number to take effect.

All licenses are entered into a file called "A5_Licenses.txt". The file is searched for in the following order of directories:

- Alpha Five Private Path
- Alpha Five user's application path
- Alpha Five EXE path
- Alpha Five Shared path
- Master database path

If more than one "A5_Licenses.txt" file is found, all are combined to create a composite license file that is used for license validation.

If none is found, a dialog box appears with ability to set the path and enter the license number. The initial path selected will be where the 1st file was found, or if none was found, then the Alpha Five user's application path (which is Alpha Five EXE path in A5 version 7 and earlier). It but can be changed to any available choice, but the Application path is the recommended for most needs. New licenses will be stored in the selected path.

It you do not have a valid license #, it will place the utility in a "Limited" mode, indicated by the title of the main dialog box. If you are still experiencing problems after entering v alid license numbers and restarting, you can press on the utility's Help screen, the **Diag Info** or **Report Bug** to dump key information to help solve your problem.

No Changes needed for your application

No changes are needed for your application when running Alpha Five with the CSDA Code Utility. This keeps your code pristine and ready for distribution.

Occasionally, some initial security code or autoexec code in your application may cause the CSDA Code Utility not to start (typically due to an error in the autoexec script). In this case, the CSDA Code Utility can be started by placing a single, simple line in your initialization code (ideally the 1st line):

A5Desktop()

Alternatively, another way to start the program is to run the above code in the Code editor's Interactive window.

No interference to your application

The CSDA Code Utility normally loads just after your Autoexec script (if any) finishes execution. It will then restore all Alpha Five windows if previously saved. It does not restore any variables. It installs 3 system events (**on_database_init()**, **can_database_exit(), on_database_exit()**) and displays the CSDA Code Utility's modeless dialog box.

Only when you exit a database or Alpha Five, or when you open another database, or when the CSDA Code Utility 's modeless dialog box is **selected,** is any code being executed. It makes no modification to any database data or dictionary file.

The one exception to this is the **Log at regular intervals?** or the **Backup Dictionaries at regular intervals?** selection. When either is checked, the program will run at the specified time intervals, and could slow down your application by a small amount. If this is the case, and you find it disruptive, uncheck it.

What this means is that there is almost no interference to any application's operation, unless your application jumps from Database to Database under Alpha Five control. Even then, it should cause zero interference with your system event code (if any).

So what this really says, is that it is safe to use the CSDA Code Utility no matter what you are doing in Alpha Five.
If you do find some issue, just start up Alpha Five normally (with your standard shortcut or similar), and the **CSDA Code Utility** will not operate for that specific start.

Technical issues and Limitations

Saving and restoring the windows of an Alpha Five desktop works with a few limitations

- Form & Browse layouts in Change, Enter or Query Modes are restored to View mode intentionally
- Debugger window is not restored intentionally, however it is recorded in the log.
- Form, Browse, Report, Label or Letter Layouts in Preview (print) mode are partially recorded, but can not be restored as the layout type and table/set are not available
- Delete/Mark/Unmark Operations windows cannot determine the operation's name, and hence can't restore it, unless there is only 1 Mark operation for that table or set. However, an entry to the log will be made.
- Restore of operations may restore the incorrect one, if there is more than one operation of the same type with the identical name and identical table/set names (when you have two identically named tables in different paths – a rare situation
- CSDA Code Utility's Help scrolling does not work in Alpha Five Version 6
- When restoring password protected layout and operation windows in design mode, you must enter in the password for each password-protected item. Similarly, scripts and UDF's with passwords prompt for as well, but use the last password as a default. In either case, Alpha Five does not show you what object the password is for, so you have no idea which password it wants. The CSDA Code Utility displays a box just above and behind the password prompt with the name of the object so you can enter the correct password.
- Code Editor's Interactive window text is saved, but no values are saved from its variable space. This is intentional so that a restore of the database starts cleanly
- Alpha Five windows are restored to their original positions, but may not in certain cases. One case is an initial startup form when a A5 Desktop restore occurs without leaving Alpha Five.

There also could be issues with restoring some windows due to the order in which windows are opened, and the particular maintenance release of Alpha Five.

Problems caused by errors in your code

The CSDA Code Utility will restore the windows you have previously saved. If any of those windows execute code upon being opened (e.g. a Form layout in view mode that has ON_INIT event code) that causes an error, that error will occur each time the desktop is restored. If it were a catastrophic failure (very rare) of Alpha Five, you would never be able to edit it before the failure. The solution is to start Alpha Five normally (without the A5 Desktop shortcuts). Edit the problem area or disable the offending code, and then you may continue using the A5 Desktop

Updates to the CSDA Code Utility

To update the application, select the Dialog Box Help button, and click the “Update Program” button at the bottom and follow the prompts. You must be connected to the Internet and allow Alpha Five to access the Internet for the web update to operate. You can also load an update from a local Zip’ed update file. Alternatively, you can download a fresh copy from the web site and just manually re-install over the previous one.

In either case, be sure there is only one Alpha Five database open (in other words, only 1 instance of Alpha Five) while doing the update.

If you select the “Auto-Update box on the help screen, then once a day when you start Alpha Five with the CSDA Code Utility, it will access the internet and compare your version to the current version. If the versions differ, it will prompt you through the steps to upgrade the program to the current version.).

The 1st step will give you an opportunity to download the file. After downloading, you will be prompted as to whether you want to install the update. If selected, the Update process should be invoked and the current A5 window will close. Start Alpha Five with the CSDA Code Utility’s icons, and you are ready to use the new version.

Alternatively, download the current exe installation program or Zip file, and reinstall as if you are installing for the 1st time.

A5 Desktop Save, Logged & Restore Details

The following A5 window types are saved, logged and restored except as indicated in the Technical issues section.

Form *
Browse *
Report
Label
Letter
Preview (Form/Browse /Report/Label/Letter)**
Field Rules (Tables/Sets)
Index Builder (Tables/Sets)
Table Builder
Set Builder
Append Builder
Copy Builder
Crosstab Builder
Export Builder
Import Builder

Intersect/Join/Subtract Builder
Post Builder
Query Builder
Summary Builder
Update Builder
Delete/Mark/Unmark Builder ***
Menu Editor
Toolbar Editor
Control Panel
Code Editor
Window Bar
Status Bar
HTML Editor
Script Recorder
Trace Window
Web Project Control Panel

Xbasic Explorer
OLE Automation Object Browser
Debugger ****
Web component Dialog Builder
Web component Grid Builder
Web component Linker Builder
Web component Tabbed Linker Builder
Web component Navigation System Builder
Web component Login Builder
Web component Maintenance Builder
Alpha DAO Import Builder
Alpha DAO Export Builder
Web Form Style Builder

* Change & Enter modes are restored to View mode
** Partial Save, can not be restored.
*** Can not be restored unless only 1 Delete/Mark/Unmark operation in dictionary of table/set
**** Never restored intentionally

Restoring a design window in a runtime version will either be changed to a view window or ignored.
Windows cannot be restored if not supported by the specific version of Alpha Five.

CSDA Code Utility for Alpha Five

Files Used

The following is a list of files and extensions created by the CSDA Code Utility.

- **databasename.ADD** – Text file containing the stored Desktop information, normally in the database or shadowed database directory. Can be deleted safely.
- **databasename.ABK** – Text file containing the path to use for database backups, normally in the database or shadowed database directory. Can be deleted safely.
- **CSDA_CodeUtility.AEX** – Contains software code, normally installed in **addins_installed** directory
- **A5Desktoplog.DBF, A5Desktoplog.CDX, A5Desktoplog.FPT, A5Desktoplog.DDD, A5Desktoplog.DDX, A5Desktoplog.DDM** stored normally in the Alpha Five program directory. Contains log of A5Desktop Saves, Closes, Restores. Can be deleted but will be recreated by any A5Desktop logging.
- **A5DesktopInstall.ADB, A5DesktopInstall.ALB, A5DesktopInstall.ALX, A5DesktopInstall.ALM** – files used in installation of the CSDA Code Utility. These can be deleted safely after installation.
- **A5Desktop_opt.txt** – This is a small text file in the Alpha Five program directory that contains some A5Desktop settings. It can be deleted and will be recreated if needed, but you will lose the recorded settings.
- **A5Desktop.ADB, A5Desktop.ALB, A5Desktop.ALX, A5Desktop.ALM, A5Desktop.DBF, A5Desktop.DDD, A5Desktop.DDX, A5Desktop.DDM** are a small, temporary database created and can be deleted, but you will not be able to run the **A5Vx Desktop Restore** and **A5Vx Last Desktop Restore** shortcuts
- **A5_Licenses.txt** is a file containing 1 or more licenses to use. It can be manually entered, but is normally created automatically. Blank lines or lines whose 1st non-blank character is a single quote are ignored. Anything after the license on the line is ignored as well. This file must be present to run the CSDA Code Utility. To allow other developers to use it, each non-comment line should have company name, comma, license # and any additional text. This file is normally located in the Alpha Five exe program directory (Ver 7 and earlier) or the Alpha Five user's application directory, but can also be located in the Alpha Five private, shared or master database directories.
- **Codetimer_log.txt** in the Master path of the database, and/or the applications path, contains logs of the Code Timer, and may be safely deleted. If the file exists in the applications path for A5v8, or the ALPHA5.EXE path for earlier versions, that will be the one written to.
- Window **shortcuts** that start the desktop can be safely removed.
- **A5Desktop_AppButtons.txt** is an old format for version 2.12 and earlier of the CSDA Code Utility and can be deleted for later versions
- **A5Desktop_Buttons.txt** created in your Alpha Five exe program directory (Ver 7 and earlier) or the Alpha Five user's application directory that contain the button definitions used by the Rolldown, Rollup and Minimized dialog boxes. It is created the 1st time you use the CSDA Code Utility if the file does not exist, and will be recreated to the default if erased. If the file is modified in a future version of the CSDA Code Utility, it will be automatically renamed to **A5Desktop_Buttons_yyyymmdd.txt** in an update so that the new features will be enabled/available. You can then merge your changes into the new file.
- **A5Desktop_MinimizeButtons.txt** created in your Alpha Five exe program directory (Ver 7 and earlier) or the Alpha Five user's application directory that contain the format of the Minimized dialog box. It is created the 1st time you use the CSDA Code Utility if the file does not exist, and will be recreated to the default if erased. If the file is modified in a future version of the CSDA Code Utility, it will be automatically renamed to **A5Desktop_MinimizeButtons_yyyymmdd.txt** in an update so that the new features will be enabled/available. You can then merge your changes into the new file.
- **A5Desktop_RollupButtons.txt** created in your Alpha Five exe program directory (Ver 7 and earlier) or the Alpha Five user's application directory that contain the format of the Minimized dialog box. It is created the 1st time you use the CSDA Code Utility if the file does not exist, and will be recreated to the default if erased. If the file is modified in a future version of the CSDA Code Utility, it will be automatically renamed to **A5Desktop_Rollup_yyyymmdd.txt** in an update so that the new features will be enabled/available. You can then merge your changes into the new file.
- **<databasename>.abt** in your database application directory contains the button files to use for your application
- **A5Desktop_ButtonFiles.abt** created in your Alpha Five exe program directory (Ver 7 and earlier) or the Alpha Five user's application directory that contains the button files to use.

CSDA Code Utility

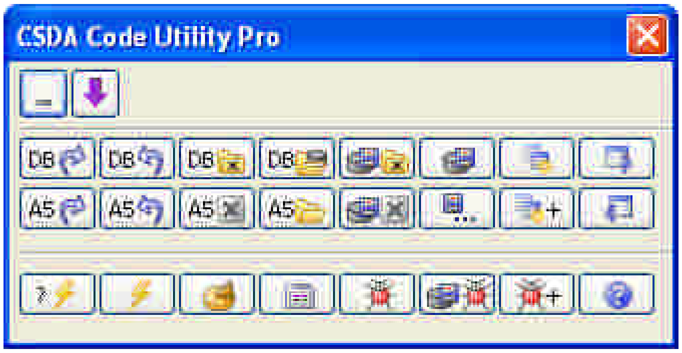
Operation

Start Alpha Five running any of the windows shortcuts created by the CSDA Code Utility. If it starts properly, you should see a floating dialog box similar to those shown below. If for some reason, the shortcuts have been deleted, go to the Interactive window of Alpha Five, type **A5Desktop()** followed by an **Enter**, and the utility should start. Recreate any shortcuts using the Create Shortcut button described below.

There are 3 standard dialog boxes with the utility, referred to as **Rolldown**, **Rollup** and **Minimized**. The **Rolldown** version has the complete set of operations. The **Rolldown** does most of the same, but uses less windows desktop real estate, and the **Minimized** has the most essential functions on it. The **Rolldown** dialog box can also have its sections shrunken to small buttons, or for the large section minimized by clicking on the blue arrows on the right.

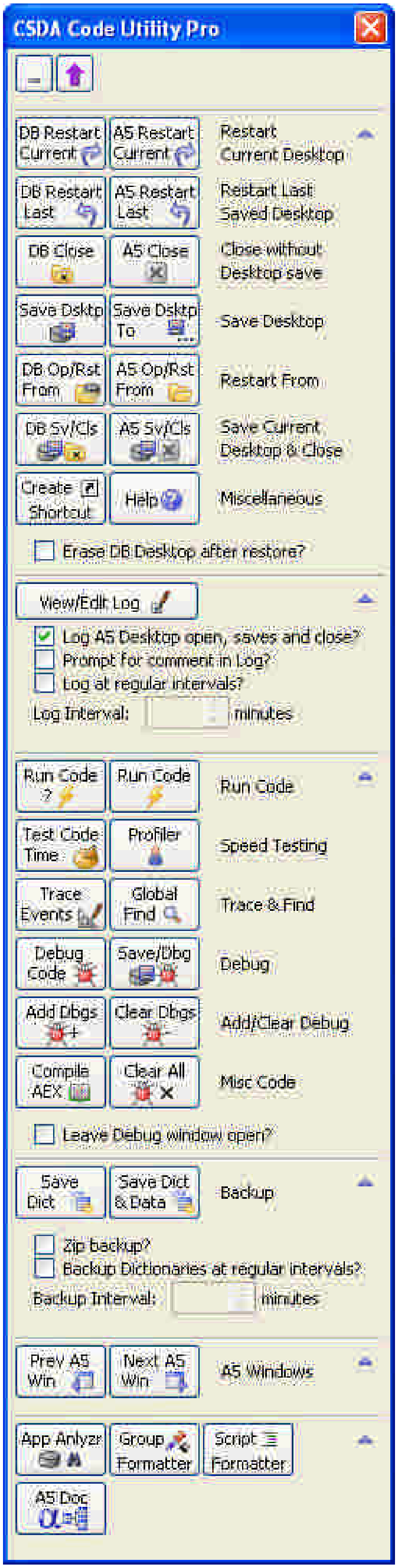
> *Note: The dialog boxes might seem to be complex, however, each of the buttons has a purpose that might have use. If you use only a subset of the buttons, a simple edit of the various dialog button configuration files will simplify them. It is suggested to comment out the supplied button lines with a leading quote, and then to view the results after a restart.*

If you have the standard license rather than the Pro License, the dialogs will not show those buttons, so the dialogs will appear a bit different than those shown.



Rollup Dialog Box

Feel free to explore the product. The program never modifies anything of your Alpha Five database (except adding and removing debug statements in your code under your control). Each button or control is explained in more detail on the following pages and has a popup help when the mouse hovers over them



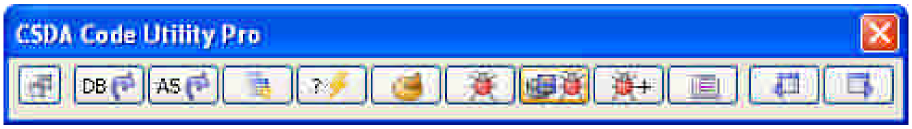
☐ Erase DB Desktop after restore?

☑ Log A5 Desktop open, saves and close?
☐ Prompt for comment in Log?
☐ Log at regular intervals?
Log Interval: minutes

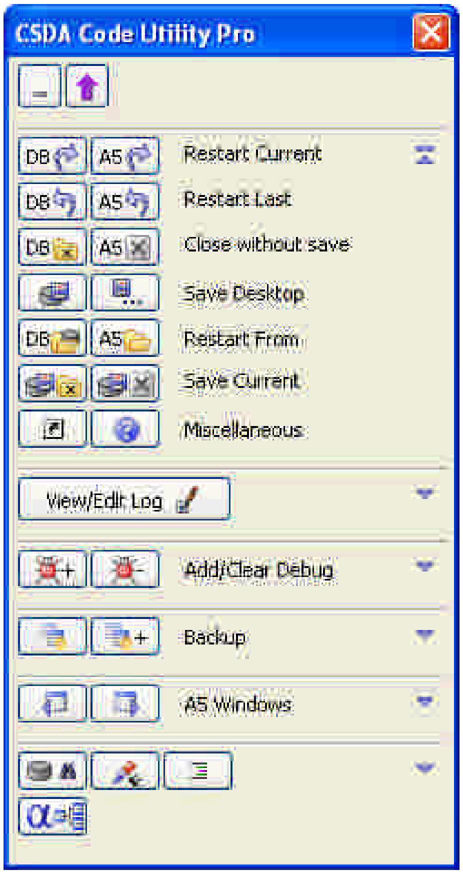
☐ Leave Debug window open?

☐ Zip backup?
☐ Backup Dictionaries at regular intervals?
Backup Interval: minutes

Rolldown Dialog Box



Minimized Dialog Box



Rolldown Dialog Box with shrunken and minimized sections

CSDA Code Utility for Alpha Five

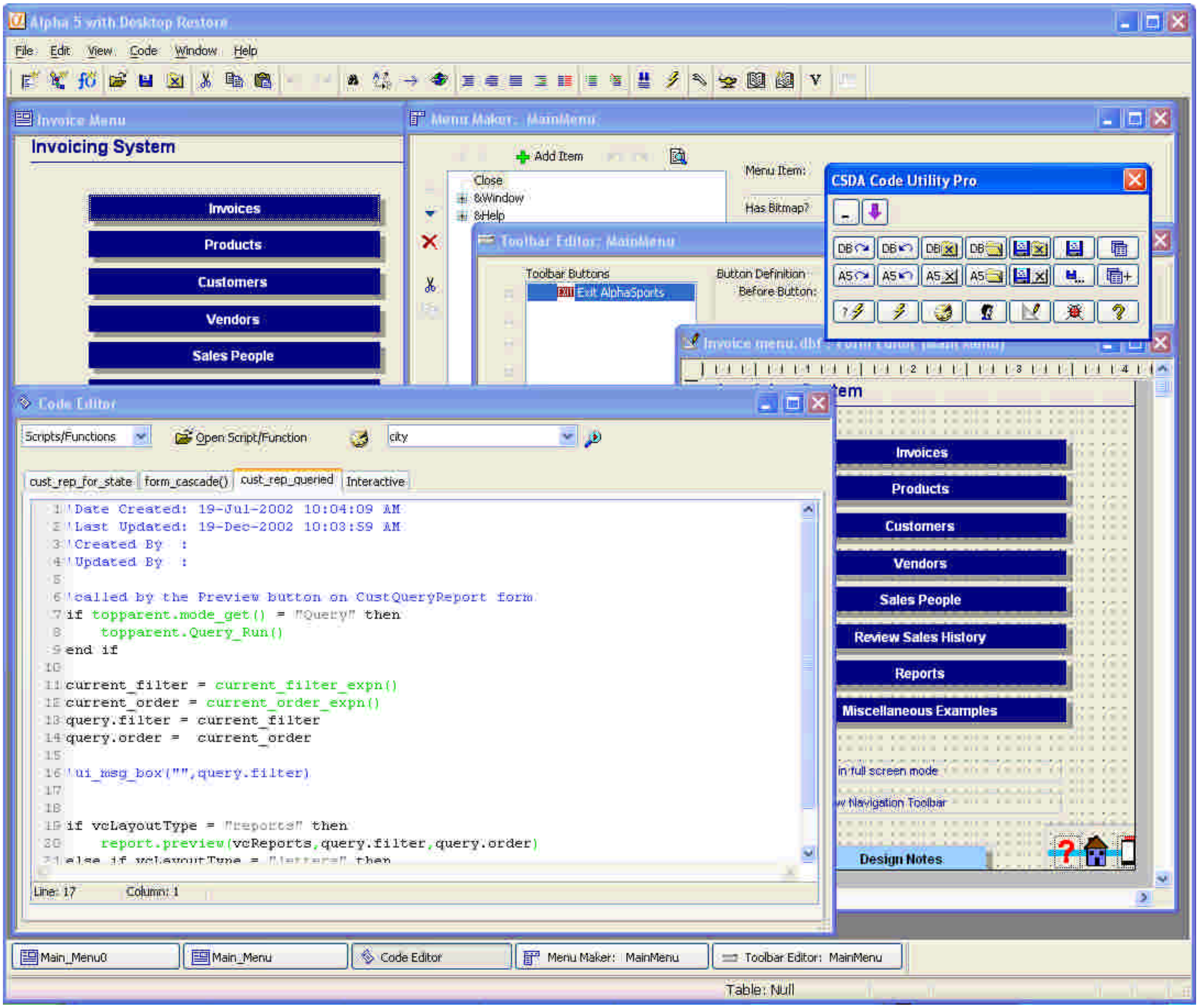
Dialog Box Window Controls

- **Vertical Up arrow** – Open's dialog box as a docked horizontal toolbar on top.
- **Underscore or double boxes** – Switches to and from **Minimized** dialog box
- **Purple Up/Down Arrows** – Switches between **Rollup** and **Rolldown** dialog box
- **Blue section Minimize/Up/Down Arrows** (on right side)
  Switches between **Large, Shrink** and **Minimized** dialog box section sizes

In addition, when the dialog box is selected, **Page Up** and **Page Down** keys will switch between **Minimize**, **Rollup** and **Rolldown** interfaces.

A5 Desktop Save/Restore Buttons

These buttons save the current Alpha Five windows, and allows restoration of these windows at a later time.



```
'Date Created: 19-Jul-2002 10:04:09 AM
'Last Updated: 19-Dec-2002 10:03:59 AM
'Created By  :
'Updated By  :

'called by the Preview button on CustQueryReport form
if topparent.mode_get() = "Query" then
    topparent.Query_Run()
end if

current_filter = current_filter_expn()
current_order = current_order_expn()
query.filter = current_filter
query.order =  current_order

'ui_msg_box("",query.filter)


if vcLayoutType = "reports" then
    report.preview(vcReports,query.filter,query.order)
else if vcLayoutType = "letters" then
```

Restored Desktop of AlphaSports

A5 Desktop Save/Restore Buttons (cont.)

The positions and sizes of each window (where possible) within Alpha Five are remembered and restored relative to the Alpha Five main window's position. If the object is off screen, it will be moved within the screen's boundaries

Most of these options have an A5 or DB version. The A5 options, closes Alpha Five, whereas the DB just closes the database while remaining in Alpha Five. The A5 option results in a fresh state of Alpha Five being loaded, but is slower than the DB option. The DB option is faster, but does not start as cleanly as the A5 option and leaves addin variables unchanged (May or may not be desirable) while switching to new databases. If you open a database from an Alpha Five menu or toolbar button, it will ask if you wish to restore the desktop (if it exists). If you select no, it will load without the restored desktop.

- **DB Restart Current** - Save Current DB Desktop, Close Database (staying in Alpha Five), Restore DB Desktop
- **A5 Restart Current** - Save Current DB Desktop, Close Alpha Five, Restore DB Desktop

- **DB Restart Last** - Close Database (without a desktop save), Restore DB's Last Saved Desktop *(Goes to previously saved desktop)*
- **A5 Restart Last** - Close Alpha Five (without a desktop save), Restore DB's Last Saved Desktop *(Goes to previously saved desktop)*

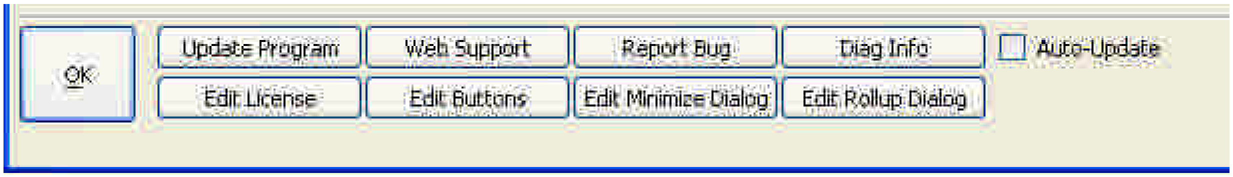
- **DB Close** - Close Database without any save of current desktop *(Doesn't save current desktop at close)*
- **A5 Close** - Close Alpha Five without any save of current desktop *(Doesn't save current desktop at close)*

- **Save Dsktp** - Save Current Desktop (to standard filename – databasename.ADD)
- **Save Dsktp To** - Save Current Desktop To (alternative file – default is databasename.ADD)

- **DB Op/Rst From** - Close Database, Open or Restore Desktop/DB from previously saved alternative desktop file
- **A5 Op/Rst From** - Close Alpha Five, Open or Restore Desktop/DB from previously saved alternative desktop file

- **DB Sv/Cls** - Save Current Desktop, Close Database
- **A5 Sv/Cls** - Save Current Desktop, Close Alpha Five

- **Create Shortcut** - Create Shortcuts on the Windows Desktop. The main purpose is to create a button to *Restore Desktop for current Database*. There are also 2 additional buttons to create shortcuts to:
  - o *Restore Last Database opened*
  - o *A5 Desktop Restore of recent databases*
- **Help** – Help (contains buttons at bottom for updating CSDA Code Utility over internet, web support, Problem/Bug reporting, Diagnostic Info, and editing license file, button definitions and dialog formats). Please note that when updating the CSDA Code Utility, over the Internet or manually, you MUST close all other Alpha Five databases except the current one. The Auto-Update will check for updates over the internet once each day when you 1st start the CSDA Code Utility and prompt you through an update process.



Help Buttons and selections at bottom of Help Screen

- **Erase DB Desktop after restore** – When checked, it erases the last saved desktop after restoring it

A5 Desktop Logging

This feature, which is only available in the Pro version, creates log entries in a standard Alpha Five directory that can be used to track usage and development of Alpha Five as well as used to enhance billing of development for Alpha Five consultants.

The log record contains a date and time stamp, Computer name, user name, Log type, Database, Current Window, list of Windows (except for control panel), Current Code Editor Tab, and a list of code editor tabs and a comment line

When using the A5 Desktop for the 1st time, it will search for a table called A5DesktopLog.dbf in the following order of directories:

- Alpha Five Private Path
- Alpha Five EXE (ver 7 and earlier) or user's application path
- Alpha Five Shared path
- Master database path
- If none is found, it will create one in the Alpha Five EXE path (ver 7 and earlier) or the user's application path with a variety of indexes and a form and browse named A5DesktopLog.

The 1st one found will be used to save log entries each time the A5 Desktop functions are activated as follows;

Type	Description	
"S"	Save of Desktop	
"A"	Save & Close of Alpha Five	
"D"	Save & Close of Database	(remains in Alpha Five)
"O"	Open of Database	
"T"	Timer Log entry	

View/Edit Log – View's A5 Desktop Log and allows edit. If a form with the name "A5DesktopLog" exists, it will use that form (which is created initially). Alternatively, it will use the 1st defined form in the table. If none exists, it will use the default form for the table. Customize the table and form as much as you wish, except do not change the table name or the names and types of the original defined fields.

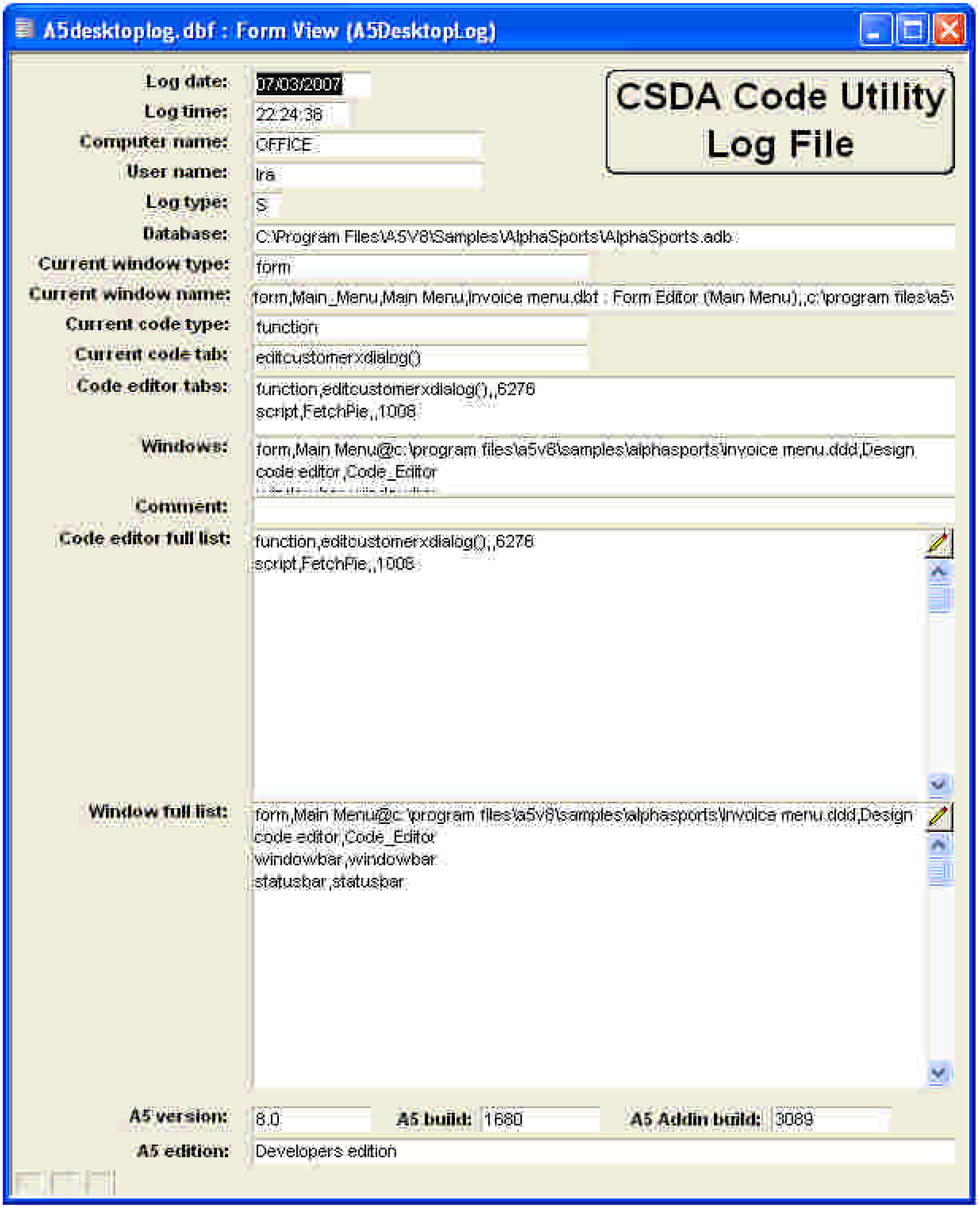
Log Desktop open, saves and close? – When checked, it saves a log record.

Prompt for comment in Log? – If Logging is selected, then prompts for a user entered comment to add to the log line to the above.

Log at regular intervals? – When checked, it saves a log record type "T" at specified intervals. This might cause a slight speed degradation when running your application. If so, uncheck it.

Log Interval – The value in this field specifies the logging interval in minutes. A value of 5 or 10 are good choices for a log of your work progress for most people

CSDA Code Utility for Alpha Five



Log date:	07/03/2007
Log time:	22:24:38
Computer name:	OFFICE
User name:	Ira
Log type:	S
Database:	C:\Program Files\A5V8\Samples\AlphaSports\AlphaSports.adb
Current window type:	form
Current window name:	form,Main_Menu,Main Menu,Invoice menu.dbf : Form Editor (Main Menu),,c:\program files\a5\
Current code type:	function
Current code tab:	editcustomerxdialog()
Code editor tabs:	function,editcustomerxdialog(),,6276 script,FetchPie,,1008
Windows:	form,Main Menu@c:\program files\a5v8\samples\alphasports\invoice menu.ddd,Design code editor,Code_Editor
Comment:	
Code editor full list:	function,editcustomerxdialog(),,6276 script,FetchPie,,1008
Window full list:	form,Main Menu@c:\program files\a5v8\samples\alphasports\invoice menu.ddd,Design code editor,Code_Editor windowbar,windowbar statusbar,statusbar
A5 version:	8.0
A5 build:	1680
A5 Addin build:	3089
A5 edition:	Developers edition

Form View of A5 Desktop log entry of AlphaSports

Code Operations

Available in the Pro version only, these add various code testing and support abilities to your code. The current tab of the Code Editor is used to run the code (even if the code editor is not the selected window). If it's the Interactive Editor window, the selected code (as if you are copying text) will be executed. For both cases, the code runs as if it is in the Interactive window (unless it is a layout's event). Any valid values in the Interactive window may be used in the code parameters. This allows you to set values prior to it's execution. For a layout event, the code is run in the layout's code space.

For most testing buttons, when running User-Defined Functions (UDF) code, you are prompted to enter the parameters. The last parameters values you gave it for that function are the default. Functions that have no parameters (e.g. DATE() function), the prompt is always skipped.

The current code is tested, whether saved or not.

Code Operation Buttons

Run Code – Run Test speed of Code. For UDF Functions, if the functions require 1 or more parameters, it prompts for parameter values. The previously used parameters are the default. The status bar shows the return type, value (1st 30 characters) and execution time.
Run Code, No Prompt – Same as Run Code, except UDF Functions use previously defined parameter values and does not prompt.

Test Code Time - Test speed of current Code. This runs the code many times to get an accurate timing number for the code. This allows measuring small differences in code speed that allow you to improve code speed and techniques. It will always take 5 seconds or greater to get the result. The statusbar window will show timing info, as well as a modeless dialog box that allows you to see the current timing results along with comparisons to the previous timing test.
Profiler - Profile timing of code. To get the overall timing of code speed (though not accurate it can provide clues as to where to spend time in the optimization process), use the Alpha Five profiler on the code

Trace Events - Trace Events of current Code Editor tab or Interactive Window Selected Text in Interactive Window context
Global Find – This brings up the Global Find for objects on the Control Panel's Code tab. If any code is selected in a Code Editor tab or the Interactive Window, that selected text will appear as the object to find in the initial search window

Global Web App Find – Searches Web Application Components for text
Global Table Find – Searches Table data for text

Code Edit – Prompts for and opens non-password protected script or UDF code in separate editing window. This is useful for side-by side comparisons.
Variable Dimensioning – Displays a dynamic window of the dimension statement lines for the variable that the current cursor is on in the code editor. Can show lines for current function/script or all functions on the current code tab

Debug Code - Debug code of current Code Editor tab or Interactive Window Selected Text in Interactive Window context.
Save & Debug – Inserts Debug code (and clears previous) into selected code on current Code Editor tab and saves code tab. Consecutive clicks (without leaving dialog box) on the button rotate though choices

Add Debug Code –Adds Debug on code at cursor for current code tab (no action scripts), or if code is selected, Debug on/Debug off around code and saves code tab.
Clear Debugs (Selected) – Deletes Debug code lines added by CSDA Code Utility in selected code on current Code Editor tab and saves code tab. If no selected code, clears all debug statements added by CSDA Code Utility.

Comment All Debugs – Comments out all active Debug code lines (even if not inserted by the CSDA Code Utility) on current Code Editor tab and saves code tab. Warning: If you have text or code that looks like a debug line, or a debug line that has been commented out, the CSDA Code Utility will comment that line as well.
Clear All Debugs – Deletes all Debug code lines (including those commented out and even if not inserted by the CSDA Code Utility) on current Code Editor tab and saves code tab. Warning: If you have text or code that looks like a debug line, or a debug line that has been commented out, the CSDA Code Utility will delete that line as well.

Compile AEX – Prompts to select scripts/functions and to compile code into aex.
Toggle Edit/View of current form/browse – Switches between Design mode and View mode of current form or browse.

Code Jump – Presents a dynamic dialog box that allows 1 click quick opening (or jumping to if already open) Control Panel objects or lines in the code editor based on text retrieved from the selected text or the word at the cursor on the code editor (or alternatively entered in a text box). Items displayed include:

- All Control Panel objects that contains that text
- All Index names that contains that text
- All lines on current code editor tab that contains that text
- All function declaration lines on the current code editor tab
- All function “end function” lines on the current code editor tab

The “Object Name“ text box can be manually populated with text to process, and the drop down allows easy selection of the previous entries from the Code Editor’s “Find/Replace” box. In this mode, it acts like a dynamic Find box with progressive display

It also provides a button to the Code Editor’s Find/Replace box, and to the Global Find, and prefills with the current text.

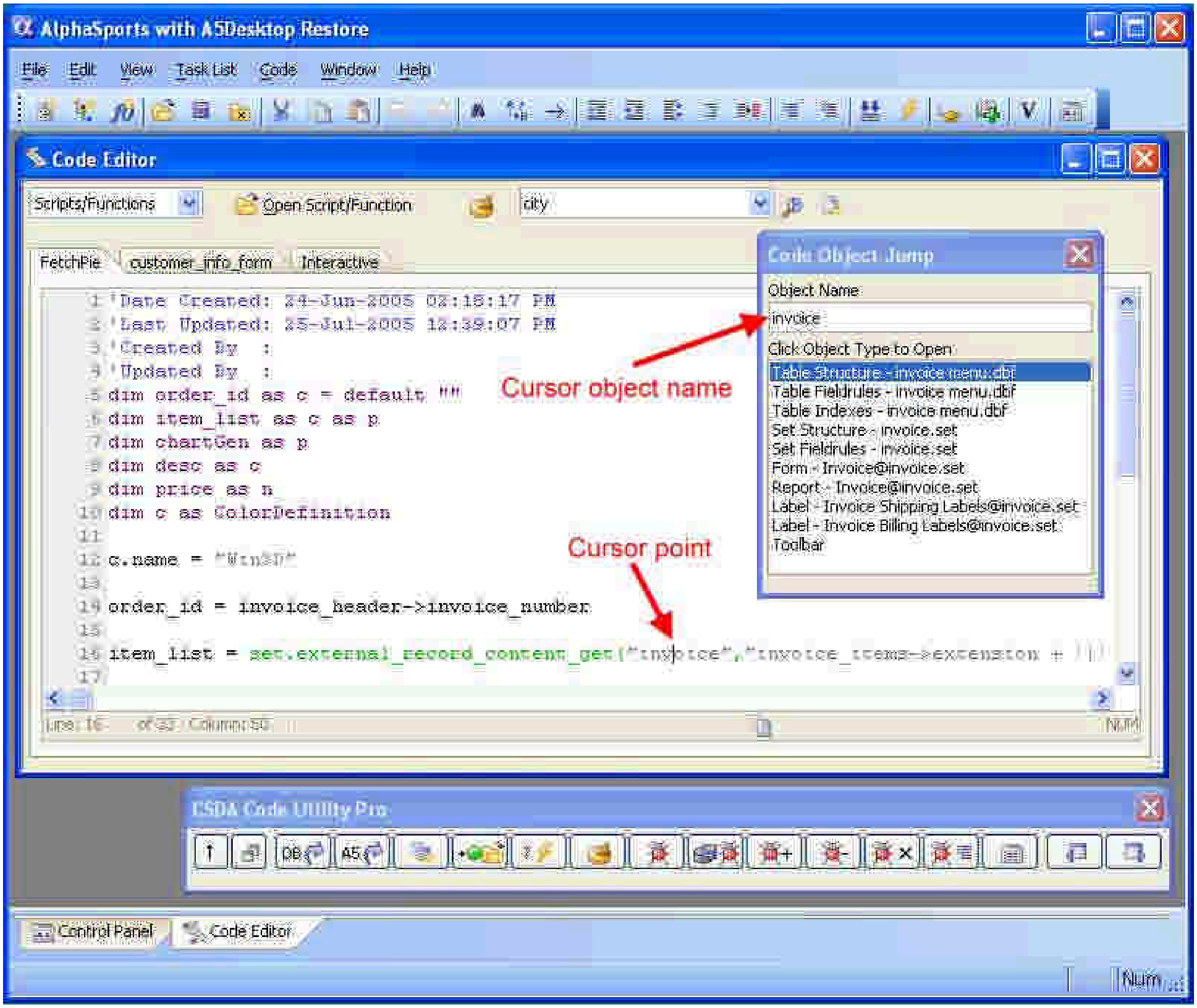
The Go Back button returns to the previous items that were jumped from

Example:
If the cursor is anywhere on the text *user_function_name* in the code editor text *“x=user_function_name(value)”*, clicking on the dialog box object “function” will immediately edit the function called “user_function_name” If there was a form of the same name, selecting “form” would open the form in design mode, and so forth. Occasionally, the current object name may be determined incorrectly. However, by selecting the text, or typing a value into the “Object Name”, the appropriate objects will be displayed. After the various matching Control Panel objects, and Index Names, any code lines in the current Code Editor’s tab that match the object will also be displayed and their positions in various internal functions. Pressing on any object line will open/jump to that object or location in the code.

Tip: If you are not using Xbasic code, the manual input acts like a quick control panel search
Tip: If you use it on an Xbasic code, you will see that it acts like a Find box, except it shows you all lines and the context.

Global Variables – Displays an editable list of Global Variables.

CSDA Code Utility for Alpha Five



```
'Date Created: 24-Jun-2005 02:18:17 PM
'Last Updated: 25-Jul-2005 12:39:07 PM
'Created By  :
'Updated By  :
dim order_id as c = default ""
dim item_list as c as p
dim chartGen as p
dim desc as c
dim price as n
dim c as ColorDefinition

c.name = "Win3D"

order_id = invoice_header->invoice_number

item_list = set.external_record_content_get("invoice","invoice_items->extension + ))
```

Line: 16 of 33 Column: 50

Code Object Jump

Object Name: invoice

Click Object Type to Open:

- Table Structure - invoice menu.dbf
- Table Fieldrules - invoice menu.dbf
- Table Indexes - invoice menu.dbf
- Set Structure - invoice.set
- Set Fieldrules - invoice.set
- Form - Invoice@invoice.set
- Report - Invoice@invoice.set
- Label - Invoice Shipping Labels@invoice.set
- Label - Invoice Billing Labels@invoice.set
- Toolbar

Code Jump Example

Running Profiler, Trace and Debugger on Selected Pieces of a Code Editor tab

If specific text is highlighted in a Code Tab, the profiler, trace window or debugger will be run only on that specific code. There are 3 cases that happens depending upon how code is selected

- No selected code – Runs entire code
- Selected code, 1 Line or more lines (must include at least one CR (Carriage Return)), runs the code tab, enabling the profiler, trace or debugger windows for just that section of selected code
- Selected code is just a portion of a single line and does not include the end CR (Carriage Return) – This runs the code tab, enabling the profiler, trace or debugger windows starting at the beginning of the selected code, but never turning the window off. This can be quickly done by just double-clicking the mouse on 1 of the words on the line

Debug Operations

These buttons insert debug on/off statements into Xbasic code, clearing previous ones, then save the code. This allows you to quickly start testing the code with the debugger on. It does not work with any Action Script coding. In most cases, the currently selected text on the code tab will remain highlighted.

Note: The Update date/time stamp of the script is not changed when the script is saved in this manner. If you had previous changes and wanted the date to be shown, save the script prior to using these buttons.

The **Save & Debug** button inserts debugger on and off code (and clears any previous debug statements added by Save & Debug or Add Debugs buttons) into code on the current Code Editor tab and then saves code tab. Consecutive clicks (without deactivating the dialog box) on the button rotate though the following choices for the current code tab:

1) Set Debugger on at cursor or current start position of selection (never turns off debugger)
2) Set Debugger on at cursor or current start position of selection and off at end of selected code or last code line if no selection
3) Set Debugger starting at 1st line of code till last line of code. If a function, then inserts debug on at the beginning line of the tab's function (not internal functions) and off at the last line of the tab's function
4) Clear all debugger lines added by CSDA Code Utility

After each click of the **Save & Debug** button, the Alpha Five status bar indicates the 4 rotating choices that were just applied.

The **Add Debugs** add additional debug statements at the current cursor position, or around the selected code

The **Clear Debugs** clears all debugs (even if commented out) created by the CSDA Code Utility within the selected code, or if no code selection, for the entire code tab. In general, this button will immediately reverse the effects of the previous **Add Debugs** press

Clear All clears all debugs (even if commented out) for the entire code tab, no matter if created by CSDA Code Utility or otherwise, and saves. This is very handy for removing forgotten debug lines from your code.

Tip: After adding the needed debug lines to your code, you can instantly run that code tab by pressing the **Run Code** or **Run Code ?** buttons.

Timing Results dialog box

After you run the "**Test Code Time**" button, the results are displayed in a **Timing Results** dialog box (as shown on next page).

The top line shows the executed code.

If the code is a User-Defined Function (UDF), the next 3 lines show

- The Alpha Five syntax (prototype) of the UDF
- The "type" of the test's returned value (it will be for the last execution performed in the iterations)
- The value of the test's returned value (it will be for the last execution performed in the iterations)

The next line contains the details of any errors encountered during testing of the code.
Note: The line number of the error may be inaccurate

The Timing results are then shown in the next 3 lines.

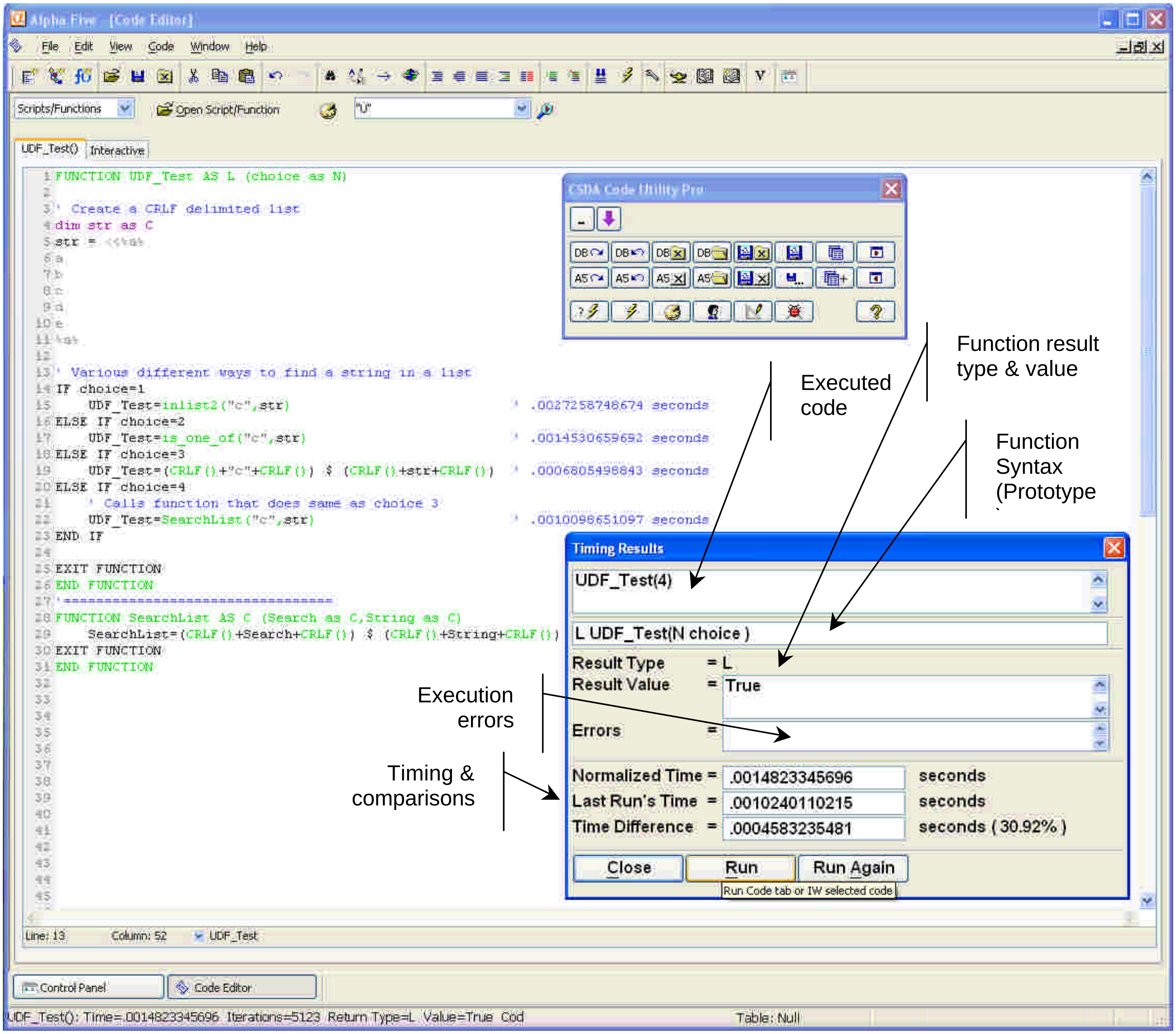
- The Normalized Time shows the average time of the code's execution over many iterations
- Last Run's Time shows the normalized time value of the last test run (if any). It does not need to be the same code tested. This allows you to test two different pieces of code in order to ascertain which is better, or alternatively, see if the code speed changes. The normal timing value for the same code should be within +/-3% but typically is less than +/-1% and get more accurate the longer the code takes to execute. These small differences are attributable to windows and Alpha Five's background operations.
- Time Difference shows the absolute time difference as well as the percentage change

The Run button basically run's the current code editor's tab with a prompt, Run Again without a prompt. You can change the code editor tab between runs, which results in the ability to compare 2 different pieces of code.

Test Code Time button restrictions

- Code that prompts for user input should not generally be tested for speed, as the timing values will be inaccurate due to variations of user input speed
- The code under test must be capable of many repeated consecutive runs without failure to get accurate values. Execution errors are generally displayed when detected. If this is an issue, you can try testing portions of the code

CSDA Code Utility for Alpha Five



```
FUNCTION UDF_Test AS L (choice as N)

' Create a CRLF delimited list
dim str as C
str = <<%a%
a
b
c
d
e
%a%

' Various different ways to find a string in a list
IF choice=1
    UDF_Test=inlist2("c",str)                                ' .0027258748674 seconds
ELSE IF choice=2
    UDF_Test=is_one_of("c",str)                              ' .0014530659692 seconds
ELSE IF choice=3
    UDF_Test=(CRLF()+"c"+CRLF()) $ (CRLF()+str+CRLF())      ' .0006805498843 seconds
ELSE IF choice=4
    ' Calls function that does same as choice 3
    UDF_Test=SearchList("c",str)                             ' .0010098651097 seconds
END IF

EXIT FUNCTION
END FUNCTION
'=================================
FUNCTION SearchList AS C (Search as C,String as C)
    SearchList=(CRLF()+Search+CRLF()) $ (CRLF()+String+CRLF())
EXIT FUNCTION
END FUNCTION
```

Line: 13 Column: 52 UDF_Test

UDF_Test(4)

L UDF_Test(N choice)

Result Type = L

Result Value = True

Errors =

Normalized Time = .0014823345696 seconds

Last Run's Time = .0010240110215 seconds

Time Difference = .0004583235481 seconds (30.92%)

UDF_Test(): Time=.0014823345696 Iterations=5123 Return Type=L Value=True Cod

Table: Null

Timing Results dialog box

Database Save Buttons

During the development process, backups are not often performed in the course of the day as they take too long, or the database is being shared. Many times you make a mistake in the code and have to revert back to an earlier version.

The saves of the Database Dictionaries as well as the data can be safely performed anytime desired, **even if other users are sharing the database**, just as long as no Alpha Five user has an exclusive open of the table or dictionary. If a user has an exclusive open, the table will not be saved. Only those tables/sets listed in the control panel - database files, standard table and set dictionaries, (and optionally the data) are saved.

If the tables of a set's definition are dropped from the table, Alpha Five will display a warning message that it will add the tables to the database in order for the set to operate. The only choice is OK. This is not a typical case, but if you've dropped tables from your database without correcting the set, it will happen.

The save occurs to a folder under the current database's master (not shadowed) folder with a folder name similar to databasename_yyyymmdd_hhmmssddd, e.g. a folder named "Alphasports_20060704_145733275". Restoring all or part of the saved dictionaries/data requires manual copying either through windows or via Alpha Five of the appropriate files.

Save Dictionaries – Saves just the data dictionaries
Save Dictionaries and Data – Saves the data dictionaries & Data

Zip Backup? – After manual backup using Save Dictionaries or Save Dictionaries and Data, zips up files

Backup Dictionaries at regular intervals? – When checked, it backups the data dictionaries (always unzipped – it ignores Zip Backup setting for speed) at specified intervals. This might cause a slight speed degradation when running your application. If so, uncheck it.

Backup Interval – The value in this field specifies the Data Dictionary backup interval in minutes. A value of 5 or 10 are good choices for most people

Alpha Five Windows Navigation Buttons

Many developers prefer to operate in maximized mode of Alpha Five windows. To make it easier to navigate to the other windows within Alpha Five, two buttons allow you to move forward and backward within the windows list.

Prev A5 Win – Navigates to the previous A5 window
Next A5 Win – Navigates to the next A5 window

Adding Other Application and Utility Buttons

There are other Alpha Five utilities that have been developed by other firms, most notably A5DOC (www.partec.net/a5doc) from Partec Database Systems (www.partec.net) and the App Analyzer, Group Formatter and Script Formatter from AIMS Datacom (www.aimsdc.net) that can provide other capabilities. The CSDA Code Utility will detect the presence of these products and add additional buttons to the bottom of the Rolldown menu if any of these are present.

You can also add your own button definitions with simple editing. A file named A5Desktop_Buttons.txt that will be created in your Alpha Five exe program directory (Ver 7 and earlier) or the Alpha Five user's application directory that contain the button definitions to display these and to add additional buttons of your own choosing. To edit the file, select the CSDA Code Utility Help button, then select the "Edit Buttons" at the bottom of the page. The comments at the top of the file gives notes on the format and editing format.

A built-in function,

```
Is_In_Addin(codename, dictionaryname)
```

is available for use in your control enable expression, displayed enable expression, or event code expression lines. The code name is assumed to be a Global UDF if it ends in "()", otherwise it assumes a global script. The dictionary name is optional and defaults to the "addin_library.AEX" A compiled library (extension AEX) can be specified, on an ADB or ALB file. If extension is ADB, it will use the ALB data dictionary.

For more details in adding buttons and changing dialog formats, see the section "Editing the Minimized, Rollup and Rolldown Dialog Box Formats"

Editing the Dialog Box Formats

You can edit the desired buttons and format of the Minimized and Rollup dialog boxes, as well as the Rolldown Dialog Box's Restart & Code Sections, and horizontal and vertical docked toolbars (vertical will be used in a future version).

These can be a different arrangement, your own utilities or the predefined values.

The format of the file and the pre-defined button names are listed in comments at the top for each button file. The current pre-defined button definitions are;

A5 Current Restart	**DB Current Restart**	**A5 Previous Restart**	**DB Previous Restart**
DB Close	**A5 Close**	**Save Current**	**Save Current To**
DB Restore From	**Restore A5 From**	**DB Save Close**	**A5 Save Close**
Shortcut	**Help**	**Run Code?**	**Run Code**
Time Code	**Profiler**	**Trace**	**Global Search**
Debug	**Save Debug**	**Add Debug**	**Clear Debug**
Compile AEX	**Clear Debugs**	**Save DB Dictionaries**	**Save DB Dictionaries and Data**
Previous Window	**Next Window**	**View Log**	**Erase Desktop**
Log Enable	**Log Comment**	**Log Periodic**	**Debug Stay**
Zip Backup	**Backup Periodic**	**Log Interval**	**Backup Interval**
Partec A5 Doc	**AIMS App Analyzer**	**AIMS Group Formatter**	**AIMS Script Formatter**
Code Edit	**Toggle Layout Edit**	**Object Explorer**	**Global Variables**
A5 Help	**A5 Message Board**	**Global Table Search***	**Global Web App Search***
Web Page Security+	**Web User Groups+**	**Web Security+**	**Variable Dim**
Comment All Debugs	**Code Jump**	**Backup Path**	**Control Panel Toggle**

* Not supported in Alpha Five version 6
+ Not supported in Alpha Five version 6 or 7

The comment lines at the top of the button definition file helps explain the button definitions and how to modify the files. These files are backed up to a similar file name with a date when new installs are done. These files may change format from version to version, so you may have to reedit them after doing an upgrade.

To allow for different button formats (skins), a button file location file can be created. These are simple text files with an extension of ".abt" This has a list of 1 or more button file types and their corresponding file that hold's the definition or format. Each button file type is searched for in the following files in the following order with the 1st one found taken as the current value.

- <databasename>.abt in the database folder
- A5Desktop_ButtonFiles.abt in the Alpha Five application path
- default button file names

Thus a database folder definition will override any application path definition, which overrides the default value. This means that you can have a custom "skin" for each database, and always default the others back to a more global definition. The currently active buttons can be seen in the Diag Info button of the Help. New definitions can be stored in text files and copied to the DBF and or A5 application path abt files to change "skins".

Tip: If you delete all text in the location file and save, the next time you edit, it will be restored to the default.

To edit the currently active button files, select the CSDA Code Utility Help button, then select the “Edit DBF Buttons Loc”, “Edit A5 Buttons Loc”, “Edit Buttons”, “Edit Minimize Dialog”, “Edit Rollup Dialog”, “Edit Rolldown Restart”, “Edit Rolldown Code” at the bottom of the page. Please note that when adding a button, if the license does not permit use of that feature, the button will not operate, although it will appear.

The basic button creation steps are as follows

1. If you have a new operation you’d like to add to a button, edit the button definition file and add your button definition. There are basically 3 button definition types. Internally defined, user-defined that show up in the Rolldown App section, and user-defined that do not show up in the Rolldown App section. Format information is located as comments at the top of the Button Definition file
2. Now edit the appropriate button format file and add the name of the button as defined on the name line of the button definition file in the appropriate place. Add additional Xdialog code as needed. Format information is located as comments at the top of the Button format files
3. If you are using different button files, edit the button files location files
4. Save the files and restart the database or Alpha Five.

Using the CSDA Code Utility functions in your code

In order to allow special uses and operations, the functions and documentation are provided for use. To get current documentation on a particular function, go into the Code Editor's Interactive Window and type the following lines and run them (by pressing enter at the end of the line).

- **A5Desktop("H")**
- **CodeTimer()**
- **DatabaseSave("","","H")**
- **LibMake()**

Free Technical support for help in using direct calls to the CSDA Code Utilities functions is not included in the license, but may be provided in some cases or in public message board forums (such as Alpha Software's)

Advanced Dialog Box Modification

The dialog box's dialog and event code can be changed if you wish to provide customization of the interface, but this is not recommended. The other method of defining buttons as described in the section "Editing the Minimized, Rollup and Rolldown Dialog Box Formats" is the preferred method.

The code can be changed using code similar to:

```
dim global CSDA_CodeUtilities as P

' This field is read-only and any changes will have no effect
dim global CSDA_CodeUtilities.Dialog_Title as C

‘ Set these variables to the new code
dim global CSDA_CodeUtilities.Dialog_Code as C
dim global CSDA_CodeUtilities.Dialog_Code_Rollup as C
dim global CSDA_CodeUtilities.Dialog_Code_Minimize as C
dim global CSDA_CodeUtilities.Dialog_Event as C

CSDA_CodeUtilities.Dialog_Code=<new dialog code>
CSDA_CodeUtilities.Dialog_Code_Rollup=<new dialog code>
CSDA_CodeUtilities.Dialog_Code_Minimize=<new dialog code>
CSDA_CodeUtilities.Dialog_Event=<new event code>

 ' Now change to the new dialog box
A5Desktop("BR")
```

Lib Make

Available in the Pro version only, this function can be called from your own build script code

Syntax: C LibMake([* ScriptList[,* LibNumber[,* Destination[,* Dictionary]]]])

Purpose: Create a compiled library with list of scripts/functions/Global Bitmaps to include

Input:

- ScriptList = A list of scripts to create AEX
  Null value uses all of the source database dictionary's scripts and functions
  Leading space prompts for scripts/functions to use, using rest of parameter as default
- LibNumber = Library Number to assign, value of 0 sets it to 64
  Negative number or leading space prompts for value, using absolute value as default
- Destination = The filename of the final AEX file
  Null uses the current database name and path for AEX
  Leading space prompts for file to use, using rest of parameter as default
  1st nonblank character is "+" places it in the "addins_installed" directory and uses the database name if no file spec
  1st nonblank character is "-" places it in the "addins_disabled" directory and uses the database name if no file spec
  No leading + or - stores in it the path and name (with aex extension)
- Dictionary = The name of the dictionary entry (alb) to get items AEX from when not specified in scriptlist
  Null Value uses current database's ALB
  Leading space prompts for database using rest of value as default

Output: Size of AEX file created

Examples:

```
LibMake("",0,"c:\test.aex","")
LibMake(get_from_file(ScriptListFileName),0,"c:\test.aex","")
LibMake(get_from_file(ScriptListFileName),0,"+test","")
LibMake(get_from_file(ScriptListFileName),0,"-test","")
```

Notes:

AEX is stored in the addins directory
Libraries are resychronized after creation (not possible if two Alpha Five windows are open)

ScriptList is a CRLF delimited value with lines like this

```
‘ Type defaults to GUDF
UDFName
UDFName,GUDF
UDFName,GUDF,Filename
UDFName,,Filename
ScriptName,SCRP
ScriptName,SCRP,Filename
' Place a bitmap in the AEX file
GlobalBitMapName,GBMP
GlobalBitMapName,GBMP,Filename
GlobalBitMapName,GBMP,Filename,EncryptionKey
‘ Place a file (any type) in the AEX file as a Global Bitmap
GlobalBitMapName,FILE,Filename
GlobalBitMapName,FILE,Filename,EncryptionKey
‘ Comment line are line's whose 1st non-blank character is a single quote
' Blank lines are ignored as well
```

If filename is included for anything but type "FILE", it is used as a reference to another database dictionary. Alternatively, the name can have a long form, such as

```
UDFname@DictionaryName.ALB
```

To use a file for the list, just use a command parameter

```
get_from_file(ScriptListFileName)
```

CSDA Code Utility for Alpha Five

Release Version History

11/29/2007	2.68	Code Jump enhanced to display Function Prototype and script name on control panel objects
		Control Panel hide loses access to database if no other windows are visible,
		so hide is disabled for that case
		Repaired Addin Start of A5Desktop - did not work if previously opened and then closed
11/28/2007	2.67	Repaired Database Saves not occurring.
11/28/2007	2.66	New Code Jump Features shows all objects and lines in current code tab and allows jumps
		Also allows typing in Code Jump creating a dynamic find
		New GoBack button for Code Jump
		New Find and Global Search buttons use current object's name
		Added ability to use selected text for code jump object
		Repaired leading .NOT.,.AND.,.XOR. in code jump word determination
		Changed EXE installer to use registry keys for alternate Alpha Five Locations
		Changed EXE installer to not display temporary install directory
		Capture failures in rebuild of addins_library.aex due to open A5 windows
		Added a path setting for quick backups that is specific to the database
		Added a code edit flag option for a5desktop to use for "addin" menus
		Added Control Panel/Code Editor "addin" menus
		Added button code to Toggle Control Panel View/hide
		Fixed startup error that caused error when switching databases under special circumstances
		Added Find/Replace button for Code Editor that prefills word at code editor cursor, whether selected or not
		Global Search now prefills word at code editor cursor, whether selected or not
11/10/2007	2.65	Code Jump now displays control panel table indexes that match object
11/04/2007	2.64	Code Jump now displays control panel items with blanks and dashes
10/31/2007	2.63	Docked menu can now be undocked
10/30/2007	2.62	Installed missing internal function for Code Jump
10/30/2007	2.61	Made Code Jump Object Name read only, improved interpretation of current word under cursor
		Code Jump now jumps to an open design object, rather than opening a new one for all
		supported objects
10/27/2007	2.60	Fixed Code Jump feature for Version 7 and earlier limitations
10/27/2007	2.59	Added new Code Jump feature
		Fixed small bug in Var Dimensioning and made faster
10/23/2007	2.58	Fixed Alpha Five Toolbar not working when database was closed
		When Alpha Five is open and Database is closed,
		Restarts work from last database/desktop file
		Fixed Var Dimensioning window not working when Script window not open
		Fixed code editor commands not supported in A5v6
		Fixed broken operations for A5v6
		Add A5 version variable for button enables
		Fixed select of code when commenting out debugs
10/20/2007	2.57	Fixed bug that didn't allow saves when the code editor was not open
10/19/2007	2.56	Saves and restores open forms and browses that are hidden - removes alias restriction
		Changed internal functions return types
10/19/2007	2.55	Fixed STOD not supported in A5v6
		Added comment out all debugs button.
		Saves and restores open forms and browses that are hidden (only with no aliases)
		Added error handler for dialog events
10/17/2007	2.54	Fixed error in A5V7 and earlier A5 versions install not finishing
10/16/2007	2.53	Fixed text that was misspelled
		Labeled update 2.52's date incorrectly
		Fixed error message at exit of Alpha Five at end of update process
10/15/2007	2.52	Fixed section rolldowns for A5V6
		Changed EXE installer to allow A5V6 to not hang installer upon exit of A5v6
		Fixed code tabs not being saved properly if changed and doing a restart
10/07/2007	2.51	Fixed icon image for Variable dimensioning button for A5v7 and earlier
		Fixed odd case licensing validation
10/03/2007	2.50	Fixed dimension var error with browse restores
		Now closes open CodeTimer windows upon closes of database
09/27/2007	2.49	Fixed dimension var error with edit of button files
		Added Var Dimension window for Code Editor
09/25/2007	2.48	Repaired CodeTimer not returning certain type of errors
		Some web components that were changed were saved with the

		dirty flag (*) set in the name and were not being restored
		Code Run and Timer fixed to handle internal functions on code tabs
		Error returns for Code timer improved
		Update dimensioning to be more tolerant of other code names
08/22/2007	2.47	Fixed Interactive variable space not being used with code operations using selected code in Interactive window
08/21/2007	2.46	Fixed A5 close on update/install in A5v8 when mouse does not move
08/18/2007	2.45	Fixed A5 title repeating text
		Size of dock menus now works with new a5v8 way
		Improved window positioning and sizing
08/14/2007	2.44	Accounted for Alpha 5 error in restoring window positions when starting A5
		Better default forms and browses detection
		Improved non-display of additional duplicate forms when application startup opens forms and browses before desktop restore
08/12/2007	2.43	Fixed Alpha 5 default forms and browses error that caused saving of window to be bad
		Improved error message display of unexpected errors
		Added error message when window can not be restored
		Added ability to enable a button format line based upon toolbar type
08/11/2007	2.42	Fixed error with save of Web Control Panel window
		Fixed error with restore of a browse
		Fixed default docked menu format's and added ability to have separator's and line feed options that are vertical or horizontal specific
		Made workaround for Alpha5V8 build 1737-3094 & earlier physical select of checkboxes
08/07/2007	2.41	Made all A5 Desktop saved windows restore to positions relative to Alpha 5 window where possible and more accurate positioning
		Added ability to dock utility on top menu bars
		Now preserves script "Updated" date-time for "Save & Debug" buttons
		Fixed save of append builder
		Fixed enable of "Code Edit" button
		Improved positioning of Utility dialog box when off screen
08/02/2007	2.40	Fixed erroneous error message when doing Update in A5V8
08/02/2007	2.39	Added Save & Restore of Web Form Style Builder window
		Improved runtime detection for various runtime editions
		Fixed interval timers to not process updates on a Date Rollover
		Trapped more potential internet errors when doing updates
		Eliminated double windows when using startup form
07/31/2007	2.38	Added additional error checking for button definitions
		Added support for early versions of A5V7 that had errors due to A5 functions added later
		Fixed error when trying to use code operations in Run Time
		Added creating separate shortcuts for runtime versions
		Added flags to differentiate Runtime in button files
		Made Sectional rollups much faster
07/29/2007	2.37	Fixed incorrect definition in Button definition file that caused program not to operate
		Added error checking and message for Display Enable lines in Button Definition file
07/28/2007	2.36	Changed A5 windows titles to reflect adb file so windows taskbar reflect current database when multiple A5 windows are open. Shadow names are in parenthesis
07/28/2007	2.35	Added windows Save & Restore of Web components Dialog, Grid, Linker, Tabbed Linker, Navigation System, Login, Maintenance, Alpha DAO Import, Alpha DAO Export
07/26/2007	2.34	Added Global Search of code to use selected code as default
		Added Global Web App search buttons
		Added Global table data search buttons
07/24/2007	2.33	Fixed bug in Web Update routine that stopped it from working
07/24/2007	2.32	A5Vx Desktop Restore shortcut now responds to welcome screen setting
		Changed code to better handle A5v8 freezing issue while cursor was off A5 windows
		Added button files redirection so that "skins" can be made
		Fixed Auto-update to check at most once a day, even if check failed
07/20/2007	2.31	Fixed Code timer 2nd click when code is highlighted of Debug & Save
		Fixed Code timer display of range for 2nd click when code is highlighted of Debug & Save

07/20/2007	2.30	Fixed Install bug that would show error box at end of an Update from the help menu or auto-update
		Added ability to update directly from a zip file stored locally
07/19/2007	2.29	Version 2.28 showed as 2.27
		Added small fix to update code to handle a bad web read
07/19/2007	2.28	Fixed Code timer so 2nd click of Debug & Save selects function code that cursor is in if no range
		Fixed Backup dictionary Interval setting which was setting the Log interval
		Fixed Backup dictionary Interval setting disable when not enabled
		Fixed serious bug in periodic logging that might lose changes to code editor tabs
		Fixed periodic logging repainting code screen, scrolling window to put cursor position at bottom
07/19/2007	2.27	Fixed Code timer error when status bar line was too long
		Fixed display of line error when running code
		Fixed using code editor's tab changed text instead of cached text buffer when running code
07/18/2007	2.26	Fixed problem with determining Browse in edit mode
		Added some new custom button definitions for use in modifying dialogs, including A5 Help, Message Board, Toggle layout edit, Object Explorer, Global Variables
		Added ability to specify custom buttons to not show up on Rolldown app section
07/17/2007	2.25	Fixed problem moving dialog box when it is off screen back to on screen
07/16/2007	2.24	Added Page Up and Page Down to move through rollup/Minimize Dialogs
		Added ability to specify restart and code section buttons for Rolldown dialog box
		Fixed display of code being run in code timer
07/14/2007	2.23	Repaired rolldown Dialog box for Standard (not Pro) version
		Added display of release Notes prior to downloading update
07/13/2007	2.22	Refreshes Dialog box when dialog box is reactivated (only works in version 8) to clear A5 redraw issues
		Changed button code to handle AIMS Group Formatter for version 8
		Eliminated unneeded nesting of main code for speed and efficiency
07/13/2007	2.21	Fixed dialog box title's being blank or wrong when moving from 1 database to another within A5
		Minimized control panel when using A5Vx Desktop Restore shortcut and during installation to reduce clutter
		Fixed save & debug insertions on line 1
07/12/2007	2.20	Fixed move of dialog box when dialog box would be off screen
07/11/2007	2.19	Fixed cosmetic "day" instead of "days" for 1 day left in trial
		Fixed initial licensing window going to limited mode when license was given
		Fixed creation of A5Desktop.adb file
07/10/2007	2.18	Fixes bug in rename of button files if previously existing
		Fixes bug in License dialog box coming up when license file was present
07/09/2007	2.17	Add debug, Clear debug, Clear All now have improved post cursor and select range after operation
		Save & Debug operates as intended
		Fixed "Edit License" button A5 version test
07/05/2007	2.16	Improved License entry screen
07/03/2007	2.15	Fixed bug in changing between rollup, rolldown and minimize dialog boxes
		Fixed arrow pointer in addon apps section rollup/rolldown arrow
		Updated documentation photos
		Repaired bugs in finding addins
		Fixed auto creation of Desktoplog form
07/02/2007	2.14	Added ability to define buttons (2 sizes) used in main dialogs which replaces "A5Desktop_AppButtons.txt" definition file previously used
		Added ability do define minimize and rollup dialog buttons
		Added Help screen edit support file buttons
		Added section rollups on Rolldown dialog box which now allows easier use on small displays
		Fixed some code operation bugs
06/25/2007	2.13	Changed code to show what object password prompt is for layouts and operations with A password during windows restore process
		Changed code of install to force recreation of Addin_Library
06/13/2007	2.12	Fixed error handling & error display in CodeTimer section
		Fixed Log Prompts to operate properly and to be only at saves, not restores
06/05/2007	2.11	Autoscroll when not enough vertical for rolldown
06/05/2007	2.10	Fixed Scroll version, and with J option on command line
06/04/2007	2.09	Added Save & Debug feature
		Added code to force dialog boxes to be on screen initially when opened
		Added code to allow automatic backup of data dictionary
		Timing routines are closed during timing and profile operations

		Repaired bug when adding 2nd number to existing license file Added edit/view License file to help
05/25/2007	2.08	Recoded CodeTimer so that it would work with unsaved code always Added selected code support to run debugger, profiler, tracer
05/19/2007	2.07	Added browse to A56Desktop log, Codetimer log alternatively stores in applications path Libmake destination path corrected for A5V8 Made wait for idle work in version 8 in install Autoexec
05/19/2007	2.06	Disable Interval Logging while running Code Timer
05/18/2007	2.05	Fixed Addin_Library.aex rebuild on updates
05/18/2007	2.04	
05/17/2007	2.03	
05/17/2007	2.02	
05/17/2007	2.01	
05/17/2007	2.00	Version to Support Vista and A5V8 paths
12/28/2006	0.9997	Added last loaded, last saved add files to diag info file Added dialog box to display current script for password prompt
12/24/2006	0.9996	Added auto update option to Help screen to check for updated version when 1st started at most once a day Added scrolling windows for dialog box screens 768 pixels high and under Added codetimer execute when unsaved options
12/18/2006	0.9995	Fixed small bug with Mark operation save with set Fixed version/serial # license issue Fixed issue of dictionary.external not liking drive label for some paths and some dictionaries, but assumes current drive. Added dialog boxes for install process Install process now closes Alpha 5 Added company name to license file to allow general usage
12/16/2006	0.9994	Added other App button support on the rolldown dialog box Added provision for version and serial number licenses
12/15/2006	0.9993	Fixed bug in Interactive window code select for timing
12/14/2006	0.9992	Disabled opening of objects in design mode when using runtime Created better Codetimer interface and displays Bad MAPI email for web support now handled correctly Added diagnostic screen Fixed License bug
12/02/2006	0.998	Fixed bug in Codetimer loop using "i", same as user code, and improved iteration count code fixed code with multi-serial number licensing, now only uses 1st serial number
12/02/2006	0.997	Added A5 trial version enable, fixed window logging to be consistent between saves and restores, Added next and previous A5 windows
12/02/2006	0.996	Changed dates to be me more date format independent
12/02/2006	0.995	
12/02/2006	0.994	Added vertical dpi fix, added save of menus and toolbar design windows, repair of action script restore, fixed code timer issues with action scripts
12/02/2006	0.993	Fixed date format bug for UK
12/01/2006	0.992	1st Public beta release
08/07/2006		Added saving of layouts in design or normal display, If not in design mode, last record displayed is restored Now saves cursors of all code windows, including interactive Restores text of interactive window (but not variable state) No longer restores script multiple times Determines Alpha5 EXE in version 8 and later No longer deletes existing "___" and "__1" scripts layout scripts are not restored text files being edited are not restored Note: Interactive window cursor is restored only if last Code editor tab was Interactive
06/02/2006		Changed code to save current scripts on can exit Stopped redundant event creation, Fixed bug with no scripts opened

05/30/2006	Fixed timing of database exit event to save current scripts
05/21/2006	Improved timing a bit for restart and close of A5
05/18/2006	Code Editor tab being edited is now the last script restored (i.e. the selected code editor tab) and cursor is placed in the same location before restart
05/12/2006	Add Control Panel show in case it's not showing
05/11/2006	Added auto-creation of "___" script, eliminated need for autoexec script modification
04/16/2006	Initial version