String Studio Add-In for Visual Basic 4.0
Version 1.0 - Shareware

Contact Samuel Prosser via Internet email at: **samuel_p@msn.com**

String Studio brings the power of C-like String Tables to Visual Basic, and the ease of use of visual String, MsgBox, and InputBox editors, with absolutely no manual coding required. Visually create, edit, browse, and organize every String, MsgBox, and InputBox in your project from one location, or from your source code with the click of a button.

STRING STUDIO'S FEATURES:

- Create, edit, browse, and manage all of your project's String Constants, MsgBoxes, and InputBoxes from one central interface. You no longer have to search through your source code to find these items!
- Organize your project's String Constants, MsgBoxes, and InputBoxes together into Groups of related types to aid in locating and managing those items. You can create, remove, and rename Groups. You can also import Groups from other projects, instead of recreating all of those common Strings, MsgBoxes, and InputBoxes in every new project.
- **PowerEdit!** Use PowerEdit to "automagically" insert the necessary code into your code window. Or, click the name of any item tracked by String Studio right in your source code window, such as a MsgBox, and then click the PowerEdit button—and bingo! A visual MsgBox editor opens with the contents of your MsgBox displayed and ready to Preview. Make your changes, click OK, and bingo! Your changes have been saved.
- Encapsulates complicated MsgBox and InputBox calls so that your source code does not show all those messy arguments and parameters. Simplify your source code!
- Easily Duplicates and Converts items. You can convert, for example, a MsgBox into an InputBox, or an InputBox into a standard String Constant, or whatever combination you desire.
- Flexible "**Pipe Codes**" allow you to easily insert any variable text into your Strings, MsgBoxes, and InputBoxes so that variable information can be displayed at run-time.

A NOTE ABOUT THIS MANUAL:

String Studio has been designed to be as intuitive and easy to use as is possible, and the manual has been designed to be as concise and functional as is possible. While every effort has been made to make this application self-explanatory, it is strongly recommended that you browse through this entire manual (it's not that long!) so that you don't miss anything.

This manual assumes you are familiar with using Windows 95 and Visual Basic 4.0.

REQUIREMENTS:

String Studio requires an installed 32-bit version of Visual Basic 4.0. Basically, if you're running a 32-bit version of Visual Basic, you have everything you need.

String Studio runs best in resolutions higher than 640x480, such as 800x600 and 1024x768.

INSTALLATION:

After unzipping the String Studio archive into a directory on your hardrive or to diskette, run the Setup application (SETUP.EXE) to install it and follow the on-screen instructions. Setup will install String Studio into the directory of your choice and add shortcuts to your Start Menu for its documentation and Support Studio.

String Studio will add itself to Visual Basic's list of available Add-Ins, after which it will be available from Visual Basic's Add-In Manager. Enabling it from Visual Basic's Add-In manager will add String Studio to Visual Basic's Add-In menu itself.

If for some reason String Studio is removed from your list of available Add-Ins, simply run String Studio itself (SSTUDIO.EXE) from Explorer, the Run command-line, or from a DOS window, and it will add itself to Visual Basic's list of available Add-Ins again.

You may UNINSTALL String Studio via the Add/Remove Control Panel.

SUPPORT NOTES:

Please use Support Studio to contact Samuel Prosser for support questions, comments, suggestions, bug reports, and so forth. Support Studio came with your String Studio archive, and should have been installed in the same directory or folder as String Studio.

Follow its on-screen instructions to email Samuel Prosser via Internet email at: **samuel_p@msn.com**

REGISTRATION:

Registration benefits:

- String Studio will continue to function fully beyond the 21 day evaluation period.
- No registration reminder screens will appear.
- You will receive priority customer support.
- All 1.x upgrade versions of String Studio will be free of charge to you.
- You will receive discounts on major upgrades of String Studio in the future, and you may be entitled to receive discounts on other products released by Samuel Prosser in the future.

Registration pricing for String Studio Add-In for Visual Basic 4.0 - Version 1.0 - is as follows:

Single user license:	$29.95 (USD)
Multi-user site license (2 or more)	$49.95 (USD) + 19.95 (USD) for each additional user
Multi-user site license (10 or more)	$199.95 (USD) + 9.95 (USD) for each additional user

For your convenience, we have contracted another company, NorthStar Solutions, to process any orders you may wish to place with your Visa, MasterCard, Discover card, a Check, or a money order.

Shortly after you register, you will be contacted via Internet email by the author with your registration key and information on how to use it.

IMPORTANT: (1) NorthStar processes registrations only. Please contact Samuel Prosser using Support Studio for any product/technical support.

NorthStar Solutions can be easily contacted for orders only via any of the following methods:

PHONE ORDERS:

Available 10 AM - 8 PM, EST, Monday - Saturday.
1-800-699-6395 (calls from the U.S. only)
1-803-699-6395 (calls from outside U.S.)

FAX ORDERS:

Available 24 hours. International and business orders encouraged.
1-803-699-5465

INTERNET ORDERS:

Simply fill out the online order form at:

http://ourworld.compuserve.com/homepages/starmail

EMAIL ORDERS:

Internet:

71561.2751@compuserve.com

America Online:

STARMAIL

CompuServe:

71561,2751

MAIL ORDERS:

You may register with check or money order (U.S. currency).
Make them payable to "NorthStar Solutions" and send them to:

NorthStar Solutions
PO BOX 25262
Columbia, SC, 29224
USA

Please provide (or be prepared to provide) the following information when ordering:

- The name and version of the program you are registering.
- Your name, business name (if applicable) and mailing address.
- Your Visa, MasterCard, or Discover card number and its expiration date (if using credit card).
- Your email address (so NorthStar solutions can send an email confirming your order and so the author can contact you easily with your registration key and any important follow-up information, upgrade announcements, etc.).

RUNNING STRING STUDIO:

Once String Studio is available on your Add-In menu, simply click the menu item labeled "String Studio" to use it with the currently open project.

NOTE: Once you have added String Studio code to your project, all of your code will run and compile whether or not String Studio is actually loaded. But String Studio must be loaded if you wish to edit any items created by String Studio.

HOW STRING STUDIO WORKS:

First of all, String Studio actually adds a read-only code module to your project, called "MODSSTUDIO.BAS." Every String, MsgBox, and InputBox you create with String Studio is encoded and stored into a Public String Constant within that module. The module contains the necessary functions required to decode these items, process Pipe Codes (we'll explain Pipe Codes later), and display them, in the case of MsgBoxes and InputBoxes.

Each String, MsgBox, and InputBox that you create will be given a name, which you assign. Thereafter, any reference made to that item in your source code will be made using that name.

You use String Constants just as you would use any other String — there is no need to call a function to use them. To display a MsgBox or InputBox, you must call the appropriate function and pass the name of the item you wish to display as an argument. The callable function names in MODSSTUDIO.BAS are user-definable. For MsgBoxes and InputBoxes, they are called "MBOX" and "IBOX" by default, respectively.

For example, if you have a MsgBox named "m_DiskNotReady," you would display this MsgBox with a line like:

```
MBOX m_DiskNotReady
```

If you want to grab the return value, you would call it like any other function. For example, your source code might look like:

```
X = MBOX(m_DiskNotReady)
```

When making changes in String Studio, String Studio will save those changes to MODSSTUDIO.BAS immediately, and therefore changes will be reflected in your program immediately if you choose to run it or compile it.

Please read the next section entitled "Using String Studio."

USING STRING STUDIO:

Once you've started String Studio from the Add-In menu with a new project, it will ask you if you wish to add a String Studio module to your project, and if so, where you would like to put it.

NOTE: By default, String Studio creates 4 Groups when adding a new module to your project. These Groups are called "Errors," "Messages," "InputBoxes," and "Strings." **HOWEVER**, these are simply suggested Group names to help you get started. String Studio does not actually correlate Group names with what they contain, as Group names are completely arbitrary to String Studio, and they are defined by the user for the sake of organization. For example, MsgBoxes will not automatically be added to the default "Messages" Group, because the user may not even choose to keep the "Messages" Group. This is by design and allows you, the user, greater flexibility in how you use String Studio.

Once loaded, String Studio will present its main window. The primary window functions are as follows:

PowerEdit - ⚡

PowerEdit is your "secret weapon." It is your key to accessing String Studio items from your code window. If you click PowerEdit while your blinking cursor is on an empty line in a code window, you will be asked if you wish to add an item, and if you do, the necessary code will be added directly to your code window.

Or, if you click PowerEdit while your blinking cursor is on the name of an existing String Studio item in your source code window, String Studio will "automagically" open that item in the appropriate editor for you.

Undo - ✕

As the name implies, Undo will "automagically" undo all of the changes you have made since the beginning of your current String Studio session, or since the last time you clicked "Save."

String Studio keeps a backup of your String Studio module (MODSSTUDIO.BAS) in the same directory as the actual module. When you "Undo" your changes, it simply reverts back to this file.

Save - ✓

As the name implies, Save will save your changes, and commit them to your backup module. See "Undo" for more information.

Setup -

Clicking on Setup will open a Setup window, which will allow you to change the following settings:

General Tab:

Save main window position, state, and size between sessions: Check this item if you want String Studio to remember how you had the main window placed between sessions.

Automatically save changes at end of session. Check this item if you ***do not*** want String Studio to automatically save changes to your String Studio module file when you end a session.

Save after every change. Check this item if you would like changes to be reflected in your project immediately.

Default number of tabs displayed on start (1-5): Change this number if you want a different number of Tabs to be opened on the main window when you start String Studio. Note, you cannot have more Tabs open that you have Groups in a project. For example, you can't have 5 Tabs open if you only have 4 Groups.

Color System. The Color System allows you to change the color coding used in the main window list display. The Color System makes it easier to visually locate different types of items in a list.

Module Tab:

This tab allows you to customize the different function names required to display MsgBoxes, InputBoxes, and process Pipe Codes in Strings.

To change a name, simply enter a new name in the appropriate combo box. To reset the function names to the default names, simply choose the name from the list portion of the combo box.

About Tab:

Aside from displaying version information, this tab displays the name and serial number for users of String Studio Professional Edition.

Roll Button - ▲

The Roll Button allows you to "Roll" the String Studio main window up or down, to conserve room on your screen. Try it! You could play with it all day!

QuickSearch:

QuickSearch allows you to type in the name of any String, MsgBox, or InputBox (located it any Group) and it will find it as you type it. Once it has highlighted the item you want, simply press Enter to open it. To see items previously opened with QuickSearch, click the list portion of the QuickSearch combo box.

This feature is mainly useful to those with large projects that contain many items.

Group Manager:

The Group Manager button opens Group Manager (imagine that!). Group Manager allows you to add or remove Groups, as well as rename any of them. If you remove a Group, any items within that Group will be lost.

In the Profession Edition of String Studio, you are able to Import Groups from other String Studio module project files. To do so, click Browse, select a String Studio module file, and click OK. Then, once the file has been loaded, you may select a Group, and click Import to import it into your current project.

NOTE: If any names of imported Groups or items match those of existing items or Groups, a number will be appended to the end of the Group or item name. For example, if you already have a Group named "Errors" and you import another Group named "Errors," the imported Group's actual name will be "Errors1."

Tabs:

Each Tab shows a different Group. The number of open Tabs can be changed by right-clicking on an empty part of the displayed list and adding or removing a Tab as desired. You can have a maximum of 5 Tabs open at any given time. The default number of Tabs you have open will depend on your settings in Setup.

Group Combo Box:

The Group Combo Box allows you to select which Group is displayed in each Tab. The Group Combo Box allows you to choose any Group that currently exists in your project.

As you begin to use String Studio, you will see that having both Tabs and a Group Combo Box is not redundant. You can only have 5 Tabs open at any time, so you can control which Groups are displayed in which Tab with the Group Combo Box. If a Group is not currently displayed in a Tab, you can use the Group Combo Box to select it. The Group Combo Box is particularly useful for large projects with more than 5 Groups.

Item List:

Right-clicking on an empty part of the Item List will display a Context-Sensitive menu. You may add or remove tabs, or Add Strings, MsgBoxes, and InputBoxes to the currently open Group.

Right-clicking on an item in the Item List will display another Context-Sensitive menu. You may Edit the item (double-clicking the item results in the same action), Duplicate it, Convert it (any information not applicable to the new item type will be lost), Move it to another Group in the current project, or Remove it.

By clicking the column headers of the Item List, you can change the sort order. You can sort by item name, or the item text. Click the same column header again to switch between ascending and descending order.

String Editor:

Once you've chosen to add or edit a String, the String Editor will be opened.

The Constant Name is the name used to identify this String throughout String Studio and your whole project. The naming rules are identical to variable naming rules. It is recommended that you use a standard naming convention. For example, if you use "s_" as a prefix for all String names, "m_" as prefix for all MsgBox names, and "i_" as a prefix for all InputBox names, String Studio will sort your items by type in the Item List on the main window. Also, this lessens any chance of duplicating names across different types of items.

Enter your String in the "String Text" box. Press Enter to start a new line or let it wrap.

The "Add To" combo box lists all of the Groups in the current project. By default the Group selected is the one open in the main window. You can change the destination Group by selecting a different one from this list.

If you're adding an item in PowerEdit mode, a new frame of options will be available. You may select whether you want to insert just the Constant Name in your source code window, or if you want to insert a SFORMAT function call with the current Constant Name as an argument. More details on SFORMAT can be found under the section "Using Pipe Codes."

Message Editor:

Once you've chosen to add or edit a Message, the Message Editor will be opened.

The Constant Name is the name used to identify this MsgBox throughout String Studio and your whole project. The naming rules are identical to variable naming rules. It is recommended that you use a standard naming convention. For example, if you use "s_" as a prefix for all String names, "m_" as prefix for all MsgBox names, and "i_" as a prefix for all InputBox names, String Studio will sort your items by type in the Item List on the main window. Also, this lessens any chance of duplicating names across different types of items.

Type the Title of the MsgBox in the Title combo box. If you simply want the application's title to appear, ensure that "[Application Title]" is selected from the combo list. The combo list will also track other MsgBox titles you've used in the current session of String Studio.

Enter your Message Text in the "Message" box. Press Enter to start a new line or let it wrap.

Click on the button "set" you want in the MsgBox. Note, clicking on a specific button in a set will change the default button, not just the button set used.

Select the icon you wish to use, if any.

If you want this MsgBox to work with a Help File, enter the Help File name in the Help File box, and enter the appropriate context number in the Context ID box.

Click Preview if you wish to see your MsgBox in action. Note, the actual application title will not be displayed until run-time.

The "Add To" combo box lists all of the Groups in the current project. By default the Group selected is the one open in the main window. You can change the destination Group by selecting a different one from this list.

If you're adding an item in PowerEdit mode, a new frame of options will be available. You may select whether you want to insert a MBOX statement, or an MBOX function call, the difference being equivalent to the different between a MsgBox statement and a MsgBox function call.

InputBox Editor:

Once you've chosen to add or edit an InputBox, the InputBox Editor will be opened.

The Constant Name is the name used to identify this InputBox throughout String Studio and your whole project. The naming rules are identical to variable naming rules. It is recommended that you use a standard naming convention. For example, if you use "s_" as a prefix for all String names, "m_" as prefix for all MsgBox names, and "i_" as a prefix for all InputBox names, String Studio will sort your items by type in the Item List on the main window. Also, this lessens any chance of duplicating names across different types of items.

Type the Title of the InputBox in the Title combo box. If you simply want the application's title to appear, ensure that "[Application Title]" is selected from the combo list. The combo list will also track other InputBox titles you've used in the current session of String Studio.

Enter your Prompt Text in the "Prompt" box. Press Enter to start a new line or let it wrap.

Enter your Default Text in the "Default Text" box.

If you want to specify exactly where your InputBox is displayed, enter a number, in twips, in the X and Y position boxes. Note, 0 is not a valid number due to design limitations. Enter 1 if you wish to specify the extreme top or extreme left edge of the screen.

If you want this InputBox to work with a Help File, enter the Help File name in the Help File box, and enter the appropriate context number in the Context ID box.

Click Preview if you wish to see your InputBox in action. Note, the actual application title will not be displayed until run-time.

The "Add To" combo box lists all of the Groups in the current project. By default the Group selected is the one open in the main window. You can change the destination Group by selecting a different one from this list.

USING PIPE CODES:

Suppose you need a MsgBox to display information only available at run time. For example, suppose you have a MsgBox named "m_DiskNotReady" and it must show the user which disk is not ready. How can this be done?

Using Pipe Codes, you can insert pieces of information at run-time. Pipe Codes work essentially like variables. Pipe Codes consist of the pipe character "|" followed by a number. You can place these Pipe Codes anywhere in your Message text, or even Default Text in the case of InputBoxes, in Titles, or anywhere in a String in the case of String Studio Strings.

So, for example, the message text of m_DiskNotReady might look like:

"The disk in drive '|1' is not ready."

Then you pass the run-time information to MBOX as an argument. When you display this MsgBox, your source code might look like:

```
MBOX m_DiskNotReady, drv$
```

Assuming drv$ = "A" then the message text, when displayed, will read:

"The disk in drive 'A' is not ready."

Basically, the first argument passed to MBOX will replace any instances of Pipe Code 1 ("|1"), and the second argument passed to MBOX will replace any instances of Pipe Code 2 ("|2"), and so on. There is no limit to the number of Pipe Codes you can use.

IBOX, and SFORMAT work the same way, the only difference being that SFORMAT simply returns a processed String rather than displaying anything.

Used properly, Pipe Codes are just as flexible as variables.

LICENSE AGREEMENT:

String Studio Add-In for Visual Basic 4.0
Version 1.0 - Shareware
Copyright © 1996, Samuel J. Prosser.
All Rights Reserved.

String Studio Add-In for Visual Basic and all accompanying documentation will herein be referred to as "THE PRODUCT."

Samuel J. Prosser will herein be referred to as "THE AUTHOR."

THE AUTHOR DISCLAIMS ALL WARRANTIES AS TO THIS SOFTWARE, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, FUNCTIONALITY OR DATA INTEGRITY OR PROTECTION.

The user must assume the entire risk of using THE PRODUCT.

The user is hereby licensed to use the unregistered shareware version of THE PRODUCT for evaluation purposes only without charge for a period of 21 days, after which it must be legally registered, or removed from the user's computer.

If the user has legally registered THE PRODUCT, the user is hereby licensed to use THE PRODUCT on one machine only, unless the user has purchased a site license, at which time the user is thereby licensed to use as many copies of THE PRODUCT as the user has legally registered.

The user is hereby permitted to copy and redistribute THE PRODUCT in its unmodified form with its documentation via electronic means, subject to the conditions in this license, without specific written permission.

The user is specifically prohibited from charging, or requesting donations, for any such copies.

###