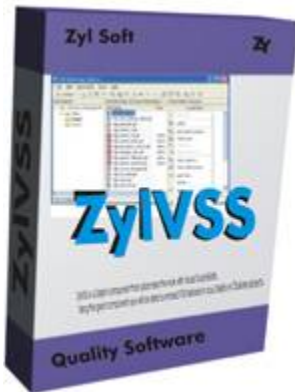


## ZyIVSS 2.26



ZyIVSS is a Delphi / C++Builder component that makes the work with Visual SourceSafe easier than you think.

Using this great component you will be able to embed version control features in your Delphi or C++Builder projects.

### **Supported Operating Systems:** Windows

NT/2000/XP/Server2003/Vista/Server2008/7/8/Server2012/10

**Available for:** Delphi 10.1, Delphi 10, Delphi XE8, Delphi XE7, Delphi XE6, Delphi XE5, Delphi XE4, Delphi XE3, Delphi XE2, Delphi XE, Delphi 2010, Delphi 2009, Delphi 2007, Delphi 2006, Delphi 2005, Delphi 7, Delphi 6, Delphi 5, C++Builder 10.1, C++Builder 10, C++Builder XE8, C++Builder XE7, C++Builder XE6, C++Builder XE5, C++Builder XE4, C++Builder XE3, C++Builder XE2, C++Builder XE, C++Builder 2010, C++Builder 2009, C++Builder 6, Turbo Delphi

### **Remarks:**

- The Delphi 2006 version is fully compatible with Turbo Delphi

### **Installation:**

If you have a previous version of the component installed, you must remove it completely before installing this version. To remove a previous installation, proceed as follows:

- Start the IDE, open the packages page by selecting Component - Install Packages
- Select ZyIVSSPack package in the list and click the Remove button
- Open Tools - Environment Options - Library and remove the library path pointing to ZyIVSS folder
- Close the IDE
- Browse to the folder where your bpl and dcp files are located (default is \$(DELPHI)\Projects\Bpl for Delphi, \$(BCB)\Projects\Bpl for C++ Builder). -Delete all of the files related to ZyIVSS
- Delete or rename the top folder where ZyIVSS is installed
- Start regedit (click Start - Run, type "regedit.exe" and hit Enter). Open the key HKEY\_CURRENT\_USER\Software\Borland\<compiler>\<version>\Palette and delete all name/value items in the list related to ZyIVSS. (<compiler> is either "Delphi" or "C++Builder", <version> is the IDE version you have installed)

-If you don't have Visual SourceSafe installed then firstly you need to register the SSAPI.dll executing the register.bat batch file.

-Unzip the zip file and open the ZyIVSSPack.dpk file in Delphi (ZyIVSSPack.bpk file in C++Builder), compile and install it and add to Tools/Environment Options/Library (in older Delphi/C++Builder menu) or Tools/Options/Delphi Options/Library/Library Path (in newer Delphi menu) or Tools/Options/C++

Options/Paths and Directories/Library Path & Include Path (in newer C++Builder menu) the path of the installation (where the ZylVSS.dcu file is located). The component will be added to the "Zyl Soft" tab of the component palette. After you have the component on your component palette, you can drag and drop it to any form, where you can set its properties by the Object Inspector and you can write event handlers selecting the Events tab of the Object Inspector and double clicking the preferred event.

If you still have problems in C++Builder, running an application, which contains the component, then open the project and in C++Builder menu, Project/Options/Packages and uncheck "Build with runtime packages".

-It is indicated to use this component with "Stop on Delphi exception" option deactivated. You can do this from Delphi / C++Builder menu, Tools/Debugger Options/Language Exceptions/Stop on Delphi exceptions in older versions or Tools/Options/Debugger Options/Embarcadero Debuggers/Language Exceptions/Notify on language exceptions in newer versions, otherwise you will have a break at all the handled exceptions.

### Deployment:

You need to deploy and register the SSAPI.dll with your application.

### Types:

**TVSSCheckout = class**

**public**

UserName: WideString;

Date: TDateTime;

LocalSpec: WideString;

Machine: WideString;

Project: WideString;

Comment: WideString;

VersionNumber: Integer;

constructor Create;

**end;**

### Methods:

**procedure Add(VSSItem : WideString; LocalFile : WideString; Comment : WideString; bReadOnly : Boolean; bKeepCheck: Boolean; bDelete : Boolean; var Error : WideString) -**

adds an item (file or directory) to the project that is specified by the VSSItem. VSSItem specifies the Visual SourceSafe item. LocalFile specifies the local file to add to the VSS project. Comment represents a comment to add. bReadOnly Boolean value indicating if the local file will be read only or not. bKeepCheck indicates whether Visual SourceSafe should keep the file checked out. bDelete indicates whether the local file should be deleted or not. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure Branch(VSSItem : WideString; Comment : WideString; var Error : WideString) -** branches a shared item. VSSItem specifies the Visual SourceSafe item. Comment is the comment to add. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure CheckIn(VSSItem : WideString; LocalFile : WideString; Comment : WideString; bKeepCheckedOut : Boolean; var Error : WideString) -** checks in a checked-out file or project. VSSItem specifies the Visual SourceSafe item. LocalFile is the local spec to check in the file from. Comment is the check in comment. bKeepCheckedOut indicates whether Visual SourceSafe should keep the file checked out. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure CheckOut(VSSItem : WideString; LocalFile : WideString; Comment : WideString; bGet : Boolean; bReplace : Boolean; var Error : WideString) -** checks out a file or project. VSSItem specifies the Visual SourceSafe item. LocalFile is the local spec to retrieve the file to. Comment is the checkout comment. bGet indicates whether the file should be retrieved. bReplace indicates whether to replace the local file with the VSS version. Error is a string that contains an

error message if the method failed, otherwise is empty.

**function Count(VSSItem : WideString; var Error : WideString) : Integer** - gets the number of children (files and subprojects) of a project. VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**constructor Create(AOwner: TComponent)** - constructor

**destructor Destroy** - destructor

**procedure DestroyItem(VSSItem : WideString; var Error : WideString)** - destroys (permanently removes) a VSS item.

VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure Get(VSSItem : WideString; var LocalFile : WideString; var Error : WideString)** - gets a file or project. VSSItem specifies the Visual SourceSafe item. LocalFile is the local file name to get it to. If is empty string (") the file is placed in a temporary directory and the full path is placed in LocalFile. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetDatabaseName(var Error : WideString):WideString** - gets the user-defined name of the Visual SourceSafe database. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetDatabaseNames: TStringList** - returns the existing database names that is necessary for the OpenDatabase function.

**function GetDatabasePaths: TStringList** - returns the existing database paths that is necessary for the OpenDatabase function.

**function GetItem(VSSItem : WideString; Index : Integer; var Error : WideString) : WideString** - gets the children (file or subproject) of a VSS project specified by index. VSSItem specifies the Visual SourceSafe item. Index specifies the index of the children to get. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetItems(VSSItem : WideString; var Error : WideString) : TStringList** - gets all the children (files and subprojects) of a project. VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetLinks(VSSItem : WideString; var Error : WideString) : TStringList** - returns the shares of the specified VSS item. VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetLocalSpec(VSSItem : WideString; var Error : WideString) : WideString** - gets the local item that corresponds to the specified Visual SourceSafe item. Valid for files and projects. VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetName(VSSItem : WideString; var Error : WideString) : WideString** - gets the name of a project or file without the path. E.g.: for \$/A/B/FU.cpp the name is FU.cpp. VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetParentProject(VSSItem : WideString; var Error : WideString) : WideString** - gets the VSSItem's parent project. E.g.: for \$/A/B/FU.cpp the parent is \$/A/B. VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetStatus(VSSItem : WideString; var Error : WideString) : Integer** - gets the VSSItem's status. Returns 0 if the item is not checked out, 1 if the item is checked out to another user and 2 if is checked out to the current user. VSSItem specifies the Visual SourceSafe item. This method works only with files. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetStatusEx(VSSItem : WideString; var Error : WideString) : TList** - gets the VSSItem's status. Returns a list of TVSSCheckout objects which contains the username, date, comment of the users who checked the file out. If the file is not checked out, the method returns **nil**. This method works only with files. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetType(VSSItem : WideString; var Error : WideString) : Integer** - gets the

VSSItem's type. Returns 0 if VSSItem is file and 1 if is a project. VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetUserName(var Error : WideString):WideString** - gets the logged-in user name. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetVersionAction(VSSItem : WideString; Version: Integer; var Error : WideString) : WideString** - gives you a string representing the action that caused the version. VSSItem specifies the Visual SourceSafe item. Version represents a specific numbered version of the VSSItem. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetVersionComment(VSSItem : WideString; Version: Integer; var Error : WideString) : WideString** - gives you the comment left by the user who took the action for the version. VSSItem specifies the Visual SourceSafe item. Version represents a specific numbered version of the VSSItem. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetVersionDate(VSSItem : WideString; Version: Integer; var Error : WideString) : TDateTime** - gives you the date/time that that this version was created. VSSItem specifies the Visual SourceSafe item. Version represents a specific numbered version of the VSSItem. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetVersionLabel(VSSItem : WideString; Version: Integer; var Error : WideString) : WideString** - gets the label of a specific version of an item. VSSItem specifies the Visual SourceSafe item. Version represents a specific numbered version of the VSSItem. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetVersionNumber(VSSItem : WideString; var Error : WideString) : Integer** - returns the current version number of the file or project. VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure GetVersion(VSSItem : WideString; Version: Integer; var LocalFile : WideString; var Error : WideString)** - gets a specific version of a file or project. VSSItem specifies the Visual SourceSafe item. Version represents a specific numbered version of the VSSItem. LocalFile is the local file name to get it to. Error is a string that contains an error message if the method failed, otherwise is empty.

**function GetVersionUsername (VSSItem : WideString; Version: Integer; var Error : WideString) : WideString** - gets the name of the user who took the action for the vVersion. VSSItem specifies the Visual SourceSafe item. Version represents a specific numbered version of the VSSItem. Error is a string that contains an error message if the method failed, otherwise is empty.

**function IsDifferent(VSSItem : WideString; LocalFile : WideString; var Error : WideString) : Boolean** - returns a Boolean value indicating whether the two files are different or not. If LocalFile is empty compares the Visual SourceSafe file with its local equivalent, otherwise compares the Visual SourceSafe file with the file specified in LocalFile parameter. VSSItem specifies the Visual SourceSafe item. LocalFile specifies a file to compare with the VSSItem. Error is a string that contains an error message if the method failed, otherwise is empty.

**function IsShared(VSSItem : WideString; var Error : WideString) : Boolean** - returns a boolean value indicating whether the item is shared. VSSItem specifies the Visual SourceSafe item. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure NewSubproject(VSSItem : WideString; Name : WideString; Comment : WideString; var Error : WideString)** - creates a Visual SourceSafe subproject in the project that is specified by the VSSItem parameter. VSSItem specifies the Visual SourceSafe item. Name is the name only (not the full spec!) of the project to create. Comment represents a comment to add. Error is a string that contains an error message if the method failed, otherwise is empty.

**function OpenDatabase(SrcSafeIni : WideString; User : WideString; Password : WideString; var Error : WideString) : Boolean** - With this function you log into Visual SourceSafe. SrcSafeIni specifies the path and the name of a SRCSAFE.INI file to use, User specifies the user name, Password specifies the password. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure SetLocalSpec(VSSItem : WideString; LocalSpec : WideString; var Error :**

**WideString**) - sets the local item (Working folder) that corresponds to the specified Visual SourceSafe item. Valid for projects only. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure SetVersionLabel(VSSItem : WideString; Version: Integer; Label\_ : WideString; Comment : WideString; var Error : WideString)** - sets the label for a specific version of an item. VSSItem specifies the Visual SourceSafe item. Version represents a specific numbered version of the VSSItem. Label\_ specifies the label to add. Comment is the comment to add. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure Share(VSSItem : WideString; VSSProject : WideString; Comment : WideString; var Error : WideString)** - shares the specified item (file or project) into the VSSProject. VSSItem specifies the Visual SourceSafe item to share. VSSProject specifies the Visual SourceSafe project to share to. Comment is the comment to add. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure SetLabel(VSSItem : WideString; Label\_ : WideString; Comment : WideString; var Error : WideString)** - sets the label for an item. VSSItem specifies the Visual SourceSafe item. Label\_ specifies the label to add. Comment is the comment to add. Error is a string that contains an error message if the method failed, otherwise is empty.

**procedure UndoCheckOut(VSSItem : WideString; LocalFile : WideString; bDelete : boolean; var Error : WideString)** - this method is used to undo checkout of a checked-out file. VSSItem specifies the Visual SourceSafe item. LocalFile is a string that contains the complete path and file name from which you want to undo the checkout. bDelete indicates whether the local file should be deleted or not. Error is a string that contains an error message if the method failed, otherwise is empty.

[Buy Now!](#)

Copyright by Zyl Soft 2003 - 2016

<http://www.zylsoft.com>

[info@zylsoft.com](mailto:info@zylsoft.com)

