SIEMENS

Linking of EasyCODE with CM-Systems

This paper describes the possibility of linking EasyCODE with Configuration Management Systems.

EasyCODE and SourceSafe

EasyCODE

SourceSafe is a project oriented version control system from Microsoft company.

- SourceSafe stores changes of the source files and enables the user to create any old version.

Linking with CM Systems

- SourceSafe combines the files in projects and follows the changes of whole projects as well as singular files.

- SourceSafe controls re-usable code; this means, a source file can be used from more than one projects at the same time.

- SourceSafe controls users, preventing that one user in the network destroys the changes made by other users.

Techn. SW-Documentation

- SourceSafe has a command interface to DOS.

As the Windows version of SourceSafe has no interface (DLL, DDE), the linking with EasyCODE must be made with the Program linking interface of EasyCODE. In that case the PVCS-DOS commands are called from entries in the Program settings dialog window.

The following SourceSafe Commands are available:

SSINIT

This command is the beginn of every SourceSafe session under DOS and usually is written into the AUTOEXEC.BAT file. A simple memory-resident programm saves the SourceSafe user-key. This step is not necessary under Novell Netware.

Change Projekt CP

Comparable with the DOS command CD (change directory), CP changes the actual project in SourceSafe: in the example the sub-project FTDLL in the project EasyCODE is chosen.

Syntax: SS CP <project>

Example: SS CP $/EasyCODE/FTDLL

Create Project

This command creates a new project (subproject). SourceSafe asks for a comment concerning the project. Starting from the assigned project (or the root-project ($/)) similar to the DOS command MD (make directory) a project is created: in the example the project EasyCODE.

Syntax: SS Create <project>

Example: SS Create EasyCODE

List Projekt

This command shows all files and projects under the actual project.

EasyCODE Version 6.xE, 06-20-1996

Syntax: SS Dir

Add File

This command adds files to the actual project.

Syntax: SS Add <file>

Example: SS Add \USR\WORK\JETFILE.C

Get File

This command copies a read-only version of the file into the actual DOS-directory. The copy cannot be changed.

Syntax: SS Get <file>

Example: SS Get JETFILE.C

Checkout File

This command is necessary if you want to change the source file. It makes a copy of your file in the DOS-directory. You can change this copy and then write it back to SourceSafe with the **Update** command.

Syntax: SS Checkout <file>

Example: SS Checkout JETFILE.C

Status

This command creates a list of all checked-out files of your project.

Syntax: SS Status

Diff File

This commando shows the actual changes.

Syntax: SS Diff <file>

Example: SS Diff JETFILE.C

Update File

This commando adds the newest version of the file into the SourceSafe archive. the copy in the DOS-directory will be deleted.

Syntax: SS Update <file>

Example: SS Update JETFILE.C

History File

This command shows all informations about previous versions of the file.

Syntax: SS History <file>

Example: SS History JETFILE.C
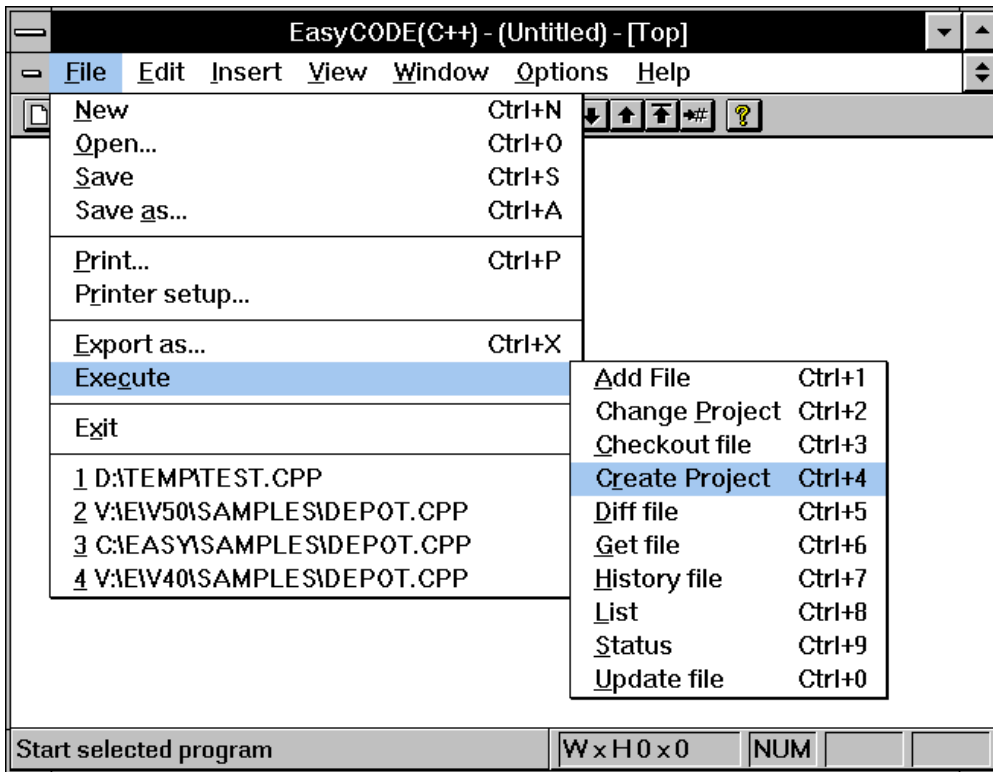
Help

This command starts the SourceSafe help system.
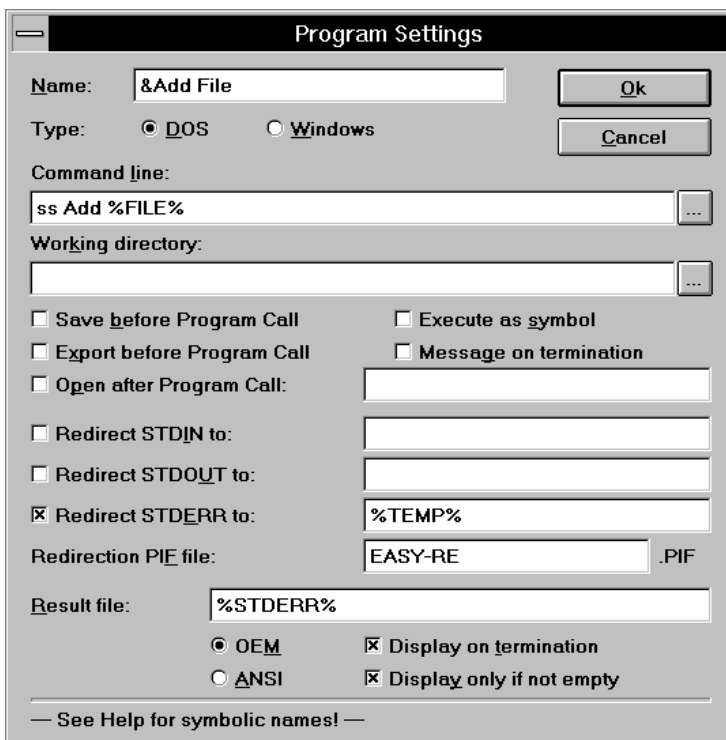
Syntax: SS Help

Example for linking EasyCODE and SourceSafe

The following SourceSafe commands shall be available in the Execute command of the EasyCODE File menu:

```
┌────────────────────────────────────────────────────────────────┐
│ ─          EasyCODE(C++) - (Untitled) - [Top]           ▼ ▲     │
├────────────────────────────────────────────────────────────────┤
│ ─  File  Edit  Insert  View  Window  Options  Help        ♦     │
├──────────────────────────────────────┬─────────────────────────┤
│ ☐  New                     Ctrl+N    │ ▼ ▲ ▲ *# ?              │
│    Open...                 Ctrl+O    │                         │
│    Save                    Ctrl+S    │                         │
│    Save as...              Ctrl+A    │                         │
│                                      │                         │
│    Print...                Ctrl+P    │                         │
│    Printer setup...                  │                         │
│                                      │                         │
│    Export as...            Ctrl+X    ┌───────────────────────┐ │
│    Execute                           │ Add File      Ctrl+1  │ │
│                                      │ Change Project Ctrl+2 │ │
│    Exit                              │ Checkout file  Ctrl+3 │ │
│                                      │ Create Project Ctrl+4 │ │
│    1 D:\TEMP\TEST.CPP                 │ Diff file      Ctrl+5 │ │
│    2 V:\E\V50\SAMPLES\DEPOT.CPP       │ Get file       Ctrl+6 │ │
│    3 C:\EASY\SAMPLES\DEPOT.CPP        │ History file   Ctrl+7 │ │
│    4 V:\E\V40\SAMPLES\DEPOT.CPP       │ List           Ctrl+8 │ │
│                                      │ Status         Ctrl+9 │ │
│                                      │ Update file    Ctrl+0 │ │
│                                      └───────────────────────┘ │
├──────────────────────────────────────┬──────────┬─────────────┤
│ Start selected program               │ W x H 0 x 0 │ NUM │     │
└──────────────────────────────────────┴──────────┴─────────────┘
```

Entries in the Program settings dialog window:

Add File command

```
┌────────────────────────────────────────────────────────────┐
│ ─              Program Settings                              │
├────────────────────────────────────────────────────────────┤
│ Name:      [ &Add File              ]        [    Ok    ]    │
│                                                              │
│ Type:      ⦿ DOS      ○ Windows              [  Cancel  ]    │
│                                                              │
│ Command line:                                                │
│ [ ss Add %FILE%                              ]  [...]        │
│ Working directory:                                           │
│ [                                            ]  [...]        │
│                                                              │
│ ☐ Save before Program Call    ☐ Execute as symbol           │
│ ☐ Export before Program Call  ☐ Message on termination      │
│ ☐ Open after Program Call:    [                 ]           │
│                                                              │
│ ☐ Redirect STDIN to:          [                 ]           │
│                                                              │
│ ☐ Redirect STDOUT to:         [                 ]           │
│                                                              │
│ ☒ Redirect STDERR to:         [ %TEMP%          ]           │
│                                                              │
│ Redirection PIF file:         [ EASY-RE         ] .PIF      │
│                                                              │
│ Result file:     [ %STDERR%                     ]           │
│                                                              │
│              ⦿ OEM    ☒ Display on termination              │
│              ○ ANSI   ☒ Display only if not empty           │
│                                                              │
│ ─ See Help for symbolic names! ─                            │
└────────────────────────────────────────────────────────────┘
```

Change Project command

Entries in the Program settings dialog window:

Name:	Change &Project
Type:	DOS
Command line:	SS CP %Prompt: Projectname ?%
Working directory:	%OPENPATH%
Redirect STDERR to:	%TEMP%
Redirection PIF file:	EASY-RE.PIF
Result file:	%STDERR%
	OEM
	☒ Display on termination
	☒ Display only if not empty

Hint: Using the %Prompt:<text>% variable in the command line you can give the filename in a separate dialog window after calling the command.

Checkout File command

Entries in the Program Settings dialog window

Name:	&Checkout File
Type:	DOS
Command line:	SS CHECKOUT %Prompt:Filename ?%
Working directory:	%OPENPATH%
☒ Execute as symbol	
Open after Program call:	%PROMPTVAL%
Redirect STDERR to:	%TEMP%
Redirection PIF file:	EASY-RE.PIF
Result file:	%STDERR%
	OEM
	☒ Display on termination
	☒ Display only if not empty

Hint: Using the %Prompt:<text>% variable in the command line you can give the filename in a separate dialog window after calling the command. This filename is stored in a variable named %PROMPTVAL%, which you can use in the edit fields. After getting the file from SourceSafe it is automatically opened in EasyCODE.

Create Projekt command

Entries in the Program Settings dialog window

Name:	C&reate Project
Type:	DOS
Command line:	SS CREATE %Prompt:Project ?%
Redirect STDERR to:	%TEMP%
Redirection PIF file:	EASY-RE.PIF
Result file:	%STDERR%
	OEM
	☒ Display on termination
	☒ Display only if not empty

Hint: Using the %Prompt:<text>% variable in the command line you can give the filename in a separate dialog window after calling the command. This filename is stored in a variable named %PROMPTVAL%, which you can use in the edit fields. After getting the file from SourceSafe it is automatically opened in EasyCODE.

EasyCODE and SourceSafe

Diff File command

Entries in the Program Settings dialog window

Name:	&Diff File
Type:	DOS
Command line:	SS DIFF %NAME%%EXT%
Working directory:	%OPENPATH%
☒ Execute as symbol	
Redirect STDOUT to:	%TEMP%
Redirection PIF file:	EASY-RE.PIF
Result file:	%STDERR%
	OEM
	☒ Display on termination

Get File command

Entries in the Program Settings dialog window

Name:	&Get File
Type:	DOS
Command line:	SS GET %Prompt:Filename ?%
Working directory:	%OPENPATH%
☒ Execute as symbol	
Open after Programm call:	%PROMPTVAL%
Redirect STDERR to:	%TEMP%
Redirection PIF file:	EASY-RE.PIF
Result file:	%STDERR%
	OEM
	☒ Display on termination
	☒ Display only if not empty

Hint: Using the %Prompt:<text>% variable in the command line you can give the filename in a separate dialog window after calling the command. This filename is stored in a variable named %PROMPTVAL%, which you can use in the edit fields. After getting the file from SourceSafe it is automatically opened in EasyCODE.

History File command

Entries in the Program Settings dialog window

Name:	&History File
Type:	DOS
Command line:	SS HISTORY %NAME%%EXT%
Working directory:	%OPENPATH%
☒ Execute as symbol	
Redirect STDOUT to:	%TEMP%
Redirection PIF file:	EASY-RE.PIF
Result file:	%STDERR%
	OEM
	☒ Display on termination

List command

Entries in the Program Settings dialog window

Name:	&List
Type:	DOS
Command line:	SS DIR
☒ Execute as symbol	
Redirect STDOUT to:	%TEMP%

Redirection PIF file: EASY-RE.PIF
Result file: %STDERR%
         OEM
         ☒ Display on termination

Status command

Entries in the Program Settings dialog window

Name: &Status
Type: DOS
Command line: SS STATUS
☒ execute as symbol
Redirect STDOUT to: %TEMP%
Redirection PIF file: EASY-RE.PIF
Result file: %STDERR%
         OEM
         ☒ Display on termination

Update File command

Entries in the Program Settings dialog window

Name: &Update File
Type: DOS
Command line: SS UPDATE %NAME%%EXT%
Working directory: %OPENPATH%
Redirect STDERR to: %TEMP%
Redirection PIFfile: EASY-RE.PIF
Result file: %STDERR%
         OEM
         ☒ Display on termination
         ☒ Display only if not empty

EasyCODE and PVCS

PVCS is one member of a family of a comprehensive Distributed Configuration Management system. The core product, PVCS, serves as a powerful Version Management System. PVCS stores the "workfiles" in a "logfile", which consists of the following information:

- the complete text of the latest revision of the source module.

- Change data (deltas) enabling PVCS to reconstruct any prior revision of the module.

- A narrative historyfor each revision that is entered by the user to describe the changes that were made to the revision.

- Information about each revision detailing who made the change and when ist was made.

- Other information that allows PVCS to manage the logfile.

EasyCODE and PVCS work together with the Program Linking Interface of EasyCODE. In this connection- similar to the linkage with SourceSafe - the PVCS DOS commands are called from Entries in the Program settings dialog window.

The following example shows how the most important PVCS commands can be integrated into EasyCODE. If you need further commands proceed analogously to the described example.
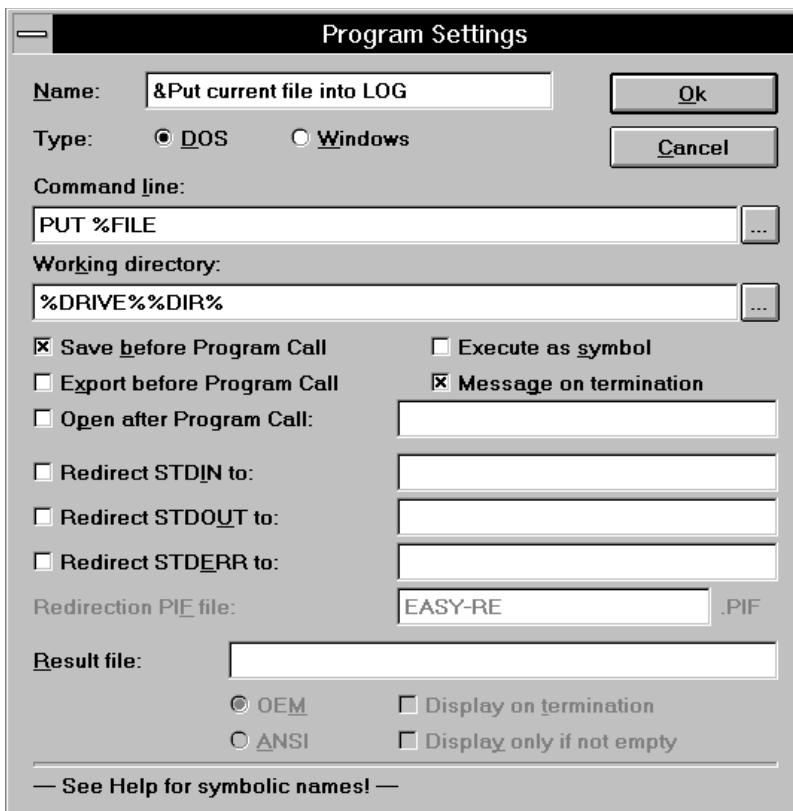
Example for linking EasyCODE and PVCS

The commands PUT, GET, VDIFF and VLOG shall be available in the Execute command of the File menu:
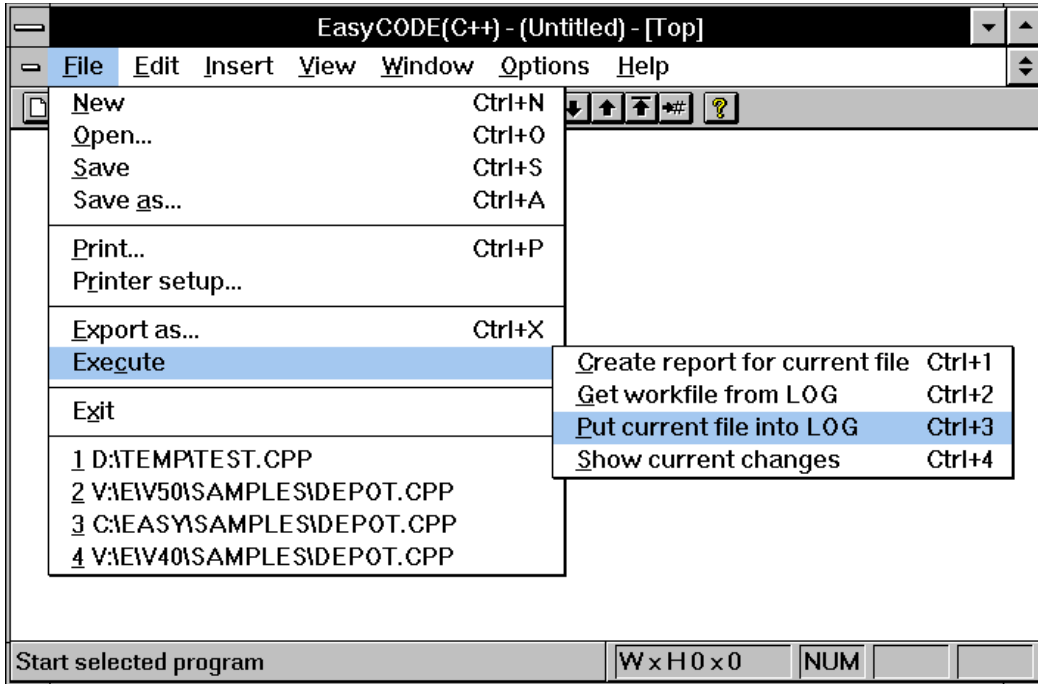
Entries in the Program settings dialog window:

PUT command

GET command

Entries in the Program Settings dialog window

Name:	&Get workfile from LOG
Type:	DOS
Command line:	GET -L %Prompt:File name ?%
Working directory:	%SAVEPATH%
Open after Programm Call:	%PROMPTVAL%

Hint: Using the %Prompt:<text>% variable in the command line you can give the filename in a separate dialog window after executing the command. This filename is stored in a variable named %PROMPTVAL%, which you can use in the edit fields. After getting the file from PVCS it is automatically opened in EasyCODE.

VLOG command

Entries in the Program Settings dialog window

Name:	&Create report for current file
Type:	DOS
Command line:	VLOG %FILE%
Working directory:	%DRIVE%%DIR%
Redirect STDOUT to:	%TEMP%
RedirectionPIF-File:	EASY-RE.PIF
Result file:	%STDOUT%
	OEM
	☒ Display on termination
	☒ Display only if not empty

VDIFF command

Entries in the Program Settings dialog window

Name:	&Show current changes
Type:	DOS
Command line:	VDIFF -R %FILE%
Working directory:	%DRIVE%%DIR%
☒ Save before Program Call	
Redirect STDOUT to:	%TEMP%
Redirection PIF file:	EASY-RE.PIF
Result file:	%STDOUT%
	OEM
	☒ Display on termination
	☒ Display only if not empty