

# GENSCRNX 1.6b

## INTRODUCTION

GENSCRNX.PRG extends the control of code generated from FoxPro's screen builder. After Generate... is selected when using the Screen Builder, GENSCRNX first copies the .SCX database and then updates it based on comments in the snippets and setup code. Also, GENSCRNX places the .SPR into a memo field after its created to make possible code changes and/or replacements after GENSCRN. The ability to define each object into a global database called FOXSCX.DBF is performed when a define object directive is placed in an object's comment code. The FOXSCX.DBF contains the same structure as FoxPro's .SCX files except has added fields for object name, field, library, and other objects that it bases from. New screens can be created without snippet code by simply placing a base object directive in the comment snippet with the appropriate name. GENSCRNX updates the .SCX before passing it to GENSCRN. Drivers can be defined in the CONFIG.FP and screen Setup snippet code. Every driver is called once for each record in the .SCX before GENSCRN generates code. The driver may update the .SCX database with no limitations. Objects can be manipulated or replaced by pure FoxPro code using a driver procedure. GENSCRNX handles the code replacements to the .SPR. A driver may use pre-made functions contained in GENSCRNX which handle the .SCX record update for code replacement, template insertion, and other .SCX update functions. A DEFINE WINDOW command can be inserted in the .SPR between the GET/SAY fields in the Screen Layout section.. Multiple drivers may also be selected for functions such as 3D effects or auto insertion of menu push buttons (Next, Previous, Append, Delete, etc.). GENSCRNX is entirely written in FoxPro and fully compatible with FoxPro 2.0 and FoxPro 2.5 (all platforms). The FOXSCX library database can be updated when referenced by FoxPro 2.0 and/or FoxPro 2.5 for MS-DOS without any conversion. FoxPro 2.5 for MS-DOS MS-DOS applications can be built by referencing objects created with FoxPro 2.0 and vice versa. The FOXSCX.DBF database can contain records for FoxPro 2.5 (all platform) while GENSCRNX automatically handles the record relation between platforms.

### Notes:

CONFIG.FP relates to FoxPro 2.0 and FoxPro 2.5 DOS.

CONFIG.FPW relates to FoxPro 2.5 Windows.

## FEATURES

- Extended control over FoxPro's Screen Builder without changing GENSCRN.
- Option for compiling the output file when generating from the Screen Builder.

- Option for displaying the .SPR and .ERR files if an .ERR file is generated after compiling the output file when generating from the Screen Builder.

- Ability to store screen objects into a database library.

- Ability to retrieve screen objects from a database library with support of multiple inheritance (expressions are separated by .AND. while procedures are appended).

- Option to set Read level settings (OpenFiles, CloseFiles, Modal, OutFile, etc.) from with the screens setup that override the Generate dialog checkboxes. This allows settings to be saved with the screen without using a project.

- Ability to insert records contained in a separate .SCX file at compile time. All records row and column information is automatically adjusted. This allows subforms to be inserted without copy and paste. If the inserted screen is updated, the screen importing it can be re-generated without change.

- Ability to insert FoxPro code in place of any screen object. This allows a line or multiple lines of FoxPro code to be generated between GET commands in the Screen Layout section.

- Ability to block a GET command with any IF/ENDIF statement.

- Ability to specify any SIZE clause and override the SIZE setting defaulted by the Screen Builder.

- Ability to remove the SIZE clause from any GET command.

- Ability to create .PRG drivers that update the .SCX database at compile time before GENSCRN is called. This allows external programs to be created that automatically add, update, or remove code of any screen snippet. Drivers can make function calls to many of GENSCRNX's built in function library for parsing or insertion of .SCX information.

- Support for any expression to be evaluated at compile time using {{<expC>}} in any snippet or field. GENSCRNX will evaluate <expC> at compile time and replace {{<expC>}} with its result. If <expC> was an external function and the command was placed in the Setup snippet, the external function could act like a #INCLUDE function by returning multiple lines of code.

Example:

If the following command was in the Setup snippet and assuming the current date was 06/01/93:

```
WAIT '{{DATE()}}' WINDOW NOWAIT
```

the following code would be placed in the .SPR:

```
WAIT '06/01/93' WINDOW NOWAIT
```

## INSTALLATION

After unzipping GENSCRNX.ZIP, copy GENSCRNX.PRG to all existing FoxPro 2.x directories

Change all CONFIG.FP and CONFIG.FPW files to:

```
_GENSCRN="<path>GENSCRNX.PRG"
```

```
MVCOUNT=512
```

Notes:

If MVCOUNT is already set to a number greater than 512, then do not change it. If MVCOUNT is set to a number less than 512, then change the number to 512. If a line containing MVCOUNT does not exist, then create one as above.

Notes:

For further information about GENSCRNX and related files, refer to VERSION.TXT supplied with GENSCRNX.ZIP.

## **CONFIG.FP/CONFIG.FPW OPTION SETTINGS**

**\_GENSCRNX**

Specifies program to generate .SPR file from .SCX database.

Default:

```
_GENSCRNX="<path>GENSCRN.PRG" in FoxPro start directory
```

Example:

```
_GENSCRNX="C:\MYDIR\MYGENSCN.PRG"
```

Notes:

When `_GENSCRN="<pathname>\GENSCRNX.PRG"`, then `_GENSCRNX` is used to specify which program is used to generate screen code. If `_GENSCRNX` is not specified in the CONFIG.FP/CONFIG.FPW, the default setting is GENSCRN.PRG located in FoxPro's start-up directory.

**\_FOXSCX**

Specifies database used for object library records.

Default:

```
_FOXSCX="FOXSCX.DBF" in FoxPro start directory
```

Example: `_FOXSCX="C:\MYDIR\FOXSCX.DBF"`

Notes:

It is recommended that all FoxPro 2.x CONFIG.FP/CONFIG.FPW

contain the same \_FOXSCX setting.

#### `_SCXDRV1`

Specifies global driver program.

Default:

`_SCXDRV1=""`

#### `_SCXDRV1` to `_SCXDRV8`

Specifies global driver program. The numbers 1-8 represent various driver hooks throughout GENSCRNX while the .SCX databases is being generated.

Example:

`_SCXDRV2="C:\3DFOX\3D"`

#### `_SPRDRV1`

Specifies global driver program.

Default:

`_SPRDRV1=""`

#### `_SPRDRV1` to `_SPRDRV6`

Specifies global driver program. The numbers 1-6 represent various driver hooks throughout GENSCRNX while the .SPR file is being updated.

Example:

`_SPRDRV1="C:\MYDIR\SPRUPD1"`

#### `GENSCRNX`

Specifies GENSCRNX functions enabled (ON) or disabled (OFF).

Default:

`GENSCRNX=ON`

#### `COMPSPR`

Specifies auto-compilation of .SPR file. A public variable called `_COMPSPR` to override the COMPSPR setting.

Default:

`COMPSPR=OFF`

Important:

This setting is ignored during screen building from projects.

## DISPSPR

Specifies auto-display of .SPR and .ERR files if an .ERR file exists. A public variable called \_DISPSPR to override the DISPSPR setting.

Default:  
DISPSPR=OFF

Important:  
DISPSPR=ON may cause a file sharing error when SHARE.EXE is installed.

## SETUP SNIPPET DIRECTIVE REFERENCE

\*:SET OPENFILES ON | OFF

Open files.

Example:  
\*:SET OPENFILES ON

\*:SET CLOSEFILES ON | OFF

Close files.

\*:SET DEFWINDS ON | OFF

Define windows.

\*:SET RELWINDS ON | OFF

Release windows.

\*:SET READCYCLE ON | OFF

Read cycle

\*:SET MULTREADS ON | OFF

Multiple READs.

\*:SET NOLOCK ON | OFF

READ nlock.

\*.SET MODAL ON | OFF

Modal.

\*.SET PLATONLY ON | OFF

Current platform objects only. If this setting is ON, GENSCRN will not generate code for other platform code but GENSCRNX will still process all platform records. Setting PLATONLY='ON' in the CONFIG.FP/CONFIG.FPW files will cause GENSCRNX to not pre or post process other platform records. See ADDITIONAL INFORMATION section below for controlling this setting using a public variable.

\*.SET BORDERGETS ON | OFF

Border for GETs.

\*.SET ASSOCWINDS TO <window title list>

Assoc. windows list. The <window title list> is appended to Assoc. windows list from screen or project.

Example:

\*.SET ASSOCWINDS TO Calculator,Calendar

\*.OUTFILE <file>

Output file name.

Example:

\*.OUTFILE TEST.PRG

\*.DEFLIB <library name>

Defines library name.

\*.INCLIB <library name>

Includes library in base object search path.

\*.BASLIB <library name>

Base library objects for field name match.

\*.SCXDRV1 <file>

Specifies screen driver program.

\*:SCXDRV1 to \*:SCXDRV8

Specifies screen driver program. The numbers 1-8 represent various driver hooks throughout GENSCRNX while the .SCX databases is being generated.

\*:SPRDRV1 <file>

Specifies screen driver program.

\*:SPRDRV1 to \*:SPRDRV6

Specifies screen driver program. The numbers 1-6 represent various driver hooks throughout GENSCRNX while the .SPR file is being updated.

\*:MEMVAR

Replaces all aliases in GET name from alias.variable to m.variable. All alias.variable names referenced in the WHEN, VALID, ERROR, MESSAGE, RANGE LO, and RANGE HIGH snippets will be replaced with m.variable.

\*:NAME

The following example demonstrates how \*:NAME affects FoxPro 2.5's #NAME directive.

```
#NAME v_show
```

is changed to

```
#NAME v_showd && _DOS=.T.  
#NAME v_showw && _WINDOWS=.T  
#NAME v_showm && _MAC=.T.  
#NAME v_showu && _UNIX=.T.
```

The above changes will occur before GENSCRN is called.  
Then, a function is appended to the Cleanup snippet as follows:

```
FUNCTION V_SHOW
```

```
DO CASE  
CASE _DOS  
RETURN V_SHOWD()  
CASE _WINDOWS  
RETURN V_SHOWW()  
CASE _MAC  
RETURN V_SHOWM()  
RETURN
```

```
CASE _UNIX  
  RETURN V_SHOWU()  
ENDCASE  
RETURN .F.
```

```
FUNCTION V_SHOWM  
RETURN .F.
```

```
FUNCTION V_SHOWU  
RETURN .F.
```

This will result in the exact same code execution as if a CASE \_DOS, CASE \_WINDOWS, etc. was generated in the snippet. The only rule is that \*:NAME uses only the first 9 characters of the snippet name specified. The 10th character is used for the platform character. Also, any PARAMETER statement that follows the #NAME in the snippet will be properly handled in the cross-platform function that is generated. The only rule here is that the PARAMETER statements must be identical for all platforms having the same #NAME definition.

\*:NOGEN

Prevents GENSCRN from being called so that no .SPR file is generated. \*:NOGEN should be used with templates since templates do not need code to be generated.

\*:NOXGEN

Prevents GENSCRNX from updating .SCX database and .SPR file.

\*:NOCOMPSPR

Overrides COMPSPR=ON in CONFIG.FP/CONFIG.FPW.

\*:NODISPSPR

Overrides DISPSPR=ON in CONFIG.FP/CONFIG.FPW.

## **COMMENT SNIPPET DIRECTIVE REFERENCE**

\*:DEFOBJ <object name>

Defines object name.

\*:BASOBJ [<library name.>]<object name>

Specify base object.



\*:INSOBJ [<library name.><object name>

Insert object from FOXSCX.DBF in place of screen object.

\*:INSSCX <file>

Insert screen from template in place of screen object.

\*:INSTXT

Insert all preceding text in place of screen object.

\*:TRNTXT <expC1> || <expC2> [[[ <expN1> ] [[[ <expN2>]]]

Transform text of \*all\* memo fields. The search is \*not\* case-sensitive.

<expC1>

The character expression that's searched for.

<expC2>

The search character expression <expC1> is replaced by the character expression <expC2>. If <expC2> is omitted, <expC1> is replaced with the null string.

<expN1>

The optional numeric expression <expN1> specifies which occurrence of <expC1> is the first to be replaced. For example, if <expN1> is 4, replacement begins with the fourth occurrence, counting from the left, and the first three occurrences remain unchanged. The occurrence where replacement begins defaults to 1 if <expN1> is omitted.

<expN2>

<expN2> specifies the number of occurrences of <expC1> to replace. If <expN2> is omitted, all occurrences of <expC1>, starting with the occurrence specified in <expN1>, are replaced.

Note:

\*:TRNTXT is mainly used with the \*:BASOBJ command for data translation of code being referenced from a library object.

\*:IF <expL>

Blocks object with IF ... ENDIF statements.

\*:SIZE <expC>

Replaces object SIZE clause with <expC>. <expC> can be any

character expression, including variable names or FoxPro functions.

**\*:NOSIZE**

Removes SIZE clause from object.

**\*:DEFAULT <expC>**

Replaces object DEFAULT clause with <expC>. <expC> can be any character expression, including variable names or FoxPro functions. Push buttons, Radio buttons, and Check boxes use the value of <expC>. Lists, invisible buttons, and spinners cannot use the \*:DEFAULT directive. All other objects use <expC> with a direct replacement. If a character default is desired, be sure to include the quotes in the expression. If the current object's color is set to default, then a COLOR SCHEME <expN> or COLOR <color pair list> may be included in <expC>.

**\*:DELETE**

Delete screen object at compile time. Use \*:DELETE for objects that need to appear while using the Screen Builder but not in the .SPR file at run-time.

## PROCEDURE SNIPPET DIRECTIVE REFERENCE

**#:INSERT <file>**

Screen generator directive inserts the contents of <file> into generated screen code. Not only does GENSCRNX support the #INSERT directive for FoxPro 2.0, but the #:INSERT directive performs the same operation as FoxPro 2.5's #INSERT except it is much faster when inserting large files.

FILE SIZE	GENSCRN #INSERT	GENSCRNX #:INSERT
2K	3.215	2.938
135K	178.717	3.475
330K	970.478	6.630

Time is in seconds using 486-50DX

When using GENSCRNX, use #:INSERT instead of #INSERT for better performance.

## PROCEDURE SNIPPET COMMAND REFERENCE

{{<expC>}}

Text surrounded by double braces performs the EVALUATION of <expC> at compile time and returns the value in string form. {{<expC>}} is replaced with the string of EVALUATE(<expC>). <expC> can be any type (character, numeric, date, logical, etc.) and {{<expC>}} will always return the result in character form.

Example:

If the following command was in the Setup snippet and assuming the current date was 06/01/93:

```
WAIT '{{DATE()}}' WINDOW NOWAIT
```

the following code would be placed in the .SPR:

```
WAIT '06/01/93' WINDOW NOWAIT
```

If the following command was in the Valid snippet:

```
DEFINE POPUP pop_test FROM {{VPOS+HEIGHT}},{{HPOS-1}};  
TO {{VPOS+HEIGHT+7}},{{HPOS+WIDTH}};  
PROMPT FIELD items.item
```

and VPOS=5, HPOS=10, WIDTH=8, HEIGHT=1 in the .SCX database, then the following code would result in the Valid snippet of that object in the .SPR:

```
DEFINE POPUP pop_test FROM 6,9;  
TO 13,18;  
PROMPT FIELD items.item
```

{{&.<expC>}}

Text surrounded by double braces with a &. immediately after the open braces performs the macro substitution of <expC> at compile time and returns a null value in string form. {{<expC>}} is replaced with a null string. <expC> can be any FoxPro command that can be executed within a macro substitution string.

Example:

If the following command was in the Setup snippet and assuming the current date was 06/01/93:

```
{{& WAIT '{{DATE()}}' WINDOW NOWAIT}}
```

the following WAIT window would appear at compile time of the screen:

06/01/93

If the following command was in the Setup snippet:

{{&.DO MYPROG}}

then a program called MYPROG would be executed as a subroutine at compile time of the screen. If the program was to return a character string for code insertion, then {{MYPROG()}} would have been used.

## DRIVER INFORMATION

Driver programs are specified either in the CONFIG.FP/CONFIG.FPW files by defining:

```
_SCXDRV2="<pathname>\[<file>"]
```

Driver programs can also be specified a screen Setup snippet by defining:

```
*:SCXDRV2 <pathname>\[<file>].
```

\*:SCXDRV1 is used before any GENSCRNX compilation. It can be used as a #INCLUDE to add any GENSCRN or GENSCRNX directives. Another method of obtaining a #INCLUDE type function is the use the braces ({{<expC>}}) when <expC> contains an external function. The character string returned from the function will replace the {{<expC>}} directly. For example, if the Setup snippet contained the following line:

```
{{inc_test()}}
```

and the external function inc\_test() return a character string of #NOREAD, then the {{inc\_test()}} line would be directly replaced by the #NOREAD command. Also, the returned character string may contain a carriage return and line feeds (CHR(13)+CHR(10)) to separate lines when multiple lines are needed for insertion. Refer to the {{<expC>}} definition supplied with GENSCRNX for further information.

Notes:

If the <file> parameter of a driver directive does not include a file extension, the following extensions are checked in this order:

.EXE, .APP, .PRG, .FXP

The n in SCXDRVn represents the hook number from GENSCRNX. GENSCRNX has 8 different places during the compiling loop that can call out to drivers. The most common one to use is #2. You can have infinite drivers for #2:

Example:

```
*:SCXDRV2 <driver1>
```

```
*:SCXDRV2 <driver2>
```

The order they are listed is the order they are called. Hook #1 is before compilation (like #INCLUDE), hook #2 is the first in each compile loop, and #8 is after compiling is complete.

Important:

Only one driver can be specified in the CONFIG.FP/CONFIG.FPW files. If more than one driver is specified in the Setup snippet, the drivers are called in the order they are listed. Drivers specified in the CONFIG.FP/CONFIG.FPW are called before the drivers specified in the Setup snippet.

## **ADDITIONAL INFORMATION**

All \*: directives used for GENSCRNX must be specified starting in column one of the snippet. Do not indent the \*: directives with spaces or tabs.

When using the Standard version of FoxPro for MS-DOS, the .SPR file size must be less than 64K.

If a public variable called \_GENSCRNX is set to OFF, GENSCRNX will pass the .SCX directly to GENSCRN and all GENSCRNX directives and commands will be ignored. GENSCRNX can also be specified in the CONFIG.FP/CONFIG.FPW files and changed without re-entering FoxPro.

If a public variable called \_PLATONLY is set to ON, GENSCRNX and GENSCRN will only generate code for the current running platform. PLATONLY can also be specified in the CONFIG.FP/CONFIG.FPW files and changed without re-entering FoxPro. \_PLATONLY='ON' is useful during development when cross-platform code generation is not required for screens until development is complete.

## **COPYRIGHT NOTICE**

Compressed file: GENSCRNX.ZIP  
System: GenScrnX  
Author: Ken R. Levy  
Company: Jet Propulsion Laboratory  
Copyright: None (Public Domain)

All source code and documentation contained in GENSCRNX.ZIP was developed at the Jet Propulsion Laboratory in Pasadena, Calif. and has been placed into the public domain. You may use, modify, copy, distribute, and demonstrate any source code, example programs, or documentation contained in GENSCRNX.ZIP freely without copyright protection. You may not resell any source code, example programs, or documentation contained in GENSCRNX.ZIP. All files contained in GENSCRNX.ZIP are provided 'as is' without warranty of any kind. In no event shall its authors, contributors, or distributors be liable for any damages.

## **COMMENTS/SUGGESTIONS/PROBLEMS/QUESTIONS**

Please use CompuServe's FoxForum (section 3rd Party Products)  
directed to:

Ken Levy 76350,2610

-----