

CST-Frame Model

This section describes the CST-Frame model, which is used to create the sample and generation subprograms for a model. Sample subprograms provide a sample of user exit code and generation subprograms supply code frame parameters.

This section covers the following topics:

- Sample Subprograms
 - Generation Subprograms
 - Parameters for the CST-Frame Model
 - User Exits for the CST-Frame Model
-

Sample Subprograms

Sample subprograms are invoked from a user exit and provide a starting sample to help the developer create user exit code. They can be simple or complicated, depending on the model.

Before invoking the sample subprograms, Natural Construct invokes all maintenance subprograms and the pre-generation subprogram. This ensures that the current specification parameters are valid and the conditions are set.

When creating a sample subprogram, you can include additional parameters to give the developer more control over what is generated into the user exit.

Note:

To pass additional information to the subprogram, use the CU—PDA.#PDAX-FRAME-PARM variable.

To define a sample subprogram:

1. Type ".E" at the beginning of a user exit line in the Code Frame editor.
2. Press Enter.

For more information about defining a sample subprogram, see Use Parameters Supplied by User Exits.

Generation Subprograms

Generation subprograms are invoked from a code frame and supply code frame parameters. Because the lengths and contents of some parameters change based on user-supplied input values or information in Predict, these parameters must be supplied by the generation subprograms. The subprograms write statements to the Natural edit buffer, based on the user-supplied input parameters or other calculated values.

To write to the edit buffer, include a `DEFINE PRINTER(SRC=1) OUTPUT 'SOURCE'` statement in the subprogram that routes the output to the source work area. To allow models to be ported to multiple platforms, use the `CU--DFPR` copycode member to define the SRC printer.

All `WRITE`, `DISPLAY`, and `PRINT` statement output for your print file is written to the edit buffer. Use the `NOTITLE` option on each of these statements. If a `DISPLAY` statement is used in the subprogram, also use the `NOHDR` option. When trailing blanks should be suppressed in variable names, the `PRINT` statement can be a useful alternative to the `WRITE` statement. However, you may want to increase the line length of the edit buffer when using the `PRINT` statement, so variable names are not split at the hyphen (-).

Because generation logic can be highly complex, these subprograms allow ultimate flexibility. However, they are less maintainable than code frame statements because you must change Natural programs to modify the generated code.

Generation subprograms can also accept the `#PDA-FRAME-PARM` constant code frame parameter from the `CU—PDA` common parameter data area. This parameter allows a subprogram to be invoked several times within the generation process. Each time the generation subprogram is invoked, it can use the value of this parameter to determine what to generate.

 **To invoke a generation subprogram:**

1. Specify line type "N" at the > prompt in the Code Frame editor.
2. Optionally, specify the constant parameter value at this prompt.

References

- For more information about generation subprograms, see *Parameters Supplied by Generation Subprograms*.
- For an example of a generated generation subprogram, refer to `CUMNGGL` in the `SYSCST` library.

Parameters for the CST-Frame Model

Use the CST-Frame model to create the generation or sample subprogram. This model has one specification panel, Standard Parameters.

Standard Parameters Panel

CUGFMA Mar 30	CST-Frame Subprogram Standard Parameters	CUG-MA0 1 of 1
Module name	CXMNGGL_	
Parameter data area	CXMNPDA_ *	
Title	Frame ..._____	
Description	This generation/sample subprogram .._____	

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---		
help retrn quit		userX main

The input fields on the Standard Parameters panel are:

Field	Description
Module name	<p>Name specified on the Generation main menu. The name of the subprogram must be alphanumeric and no more than eight characters in length. Use the following naming conventions:</p> <ul style="list-style-type: none"> <p><i>CXxxGyyy</i></p> <p>where <i>xx</i> uniquely identifies your model and <i>yyy</i> identifies your generation subprogram</p> <p><i>CXxxSyyy</i></p> <p>where <i>xx</i> uniquely identifies your model and <i>yyy</i> identifies your sample subprogram</p>
Parameter data area	<p>Name of the parameter data area (PDA) for your model. Natural Construct determines the PDA name based on the Module name specified on the Generation main menu. For example, if you enter "CXMNGAAA", Natural Construct assumes the PDA name is CXMNPDA.</p> <p>Use the following naming convention:</p> <p><i>CXxxPDA</i></p> <p>where <i>xx</i> uniquely identifies your model.</p>
Title	Title for the generated subprogram. The title identifies the subprogram for the List Generated Modules function on the Generation main menu and is used internally for program documentation.
Description	Brief description of the subprogram. The description is inserted in the banner at the beginning of the subprogram and is used internally for program documentation.

User Exits for the CST-Frame Model

CSGSAMPL		CST-Frame Subprogram		CSGSM0	
Mar 30		User Exits		1 of 1	
User Exits	Exists	Sample	Required	Conditional	

— CHANGE-HISTORY		Subprogram			
— PARAMETER-DATA					
— LOCAL-DATA					
— START-OF-PROGRAM					
— GENERATE-CODE					
— BEFORE-CHECK-ERROR		Example			
— ADDITIONAL-INITIALIZATIONS		Example			
— END-OF-PROGRAM					

For information about these user exits, see Supplied User Exits. For information about using the User Exit editor, see *User Exit Editor, Natural Construct Generation*.