

CST-Clear Model

This section describes how to use the CST-Clear model to generate the clear subprogram for your model. The clear subprogram resets variables in the model PDA.

This section covers the following topics:

- Introduction
 - Parameters for the CST-Clear Model
 - User Exits for the CST-Clear Model
-

Introduction

After defining the model PDA, use the CST-Clear model to generate the clear subprogram for your model. The clear subprogram resets the #PDA-USER-AREA variables in the model PDA. If the #PDA-USER-AREA alphanumeric field is redefined into a non-alphanumeric field that does not contain data according to the specified format, an abnormal termination may occur when it is used. To avoid this, the clear subprogram can reset redefined non-alphanumeric fields. Only non-alphanumeric variables are reset. The clear subprogram can also assign initial default values for user parameters.

The CST-Clear model assumes that your model PDA has the RESET-STRUCTURE group level name. For example:

```

*
*   User defined parameter area
  2 #PDA-USER-AREA           A  100 (1:40)
R  2 #PDA-USER-AREA           /* REDEF. BEGIN : #PDA-USER-AREA
  3 RESET-STRUCTURE
*
```

Note:

A model PDA generated by the CST-PDA model contains the RESET-STRUCTURE field.

If you do not specify a clear subprogram, the Clear Edit Buffer function on the Generation main menu sets the #PDA-USER-AREA field to blanks. The edit buffer is always cleared, regardless of whether the model uses a clear subprogram.

The nucleus invokes the clear subprogram in the following situations:

- When a user invokes the Clear Edit Buffer function on the Generation main menu.
- When a user changes the model name and the new model uses a different PDA.
- Immediately before the Read Specifications function is invoked on the Generation main menu.

Note:

For an example of a generated clear subprogram, refer to CUMNC in the SYSCST library.

Parameters for the CST-Clear Model

Use the CST-Clear model to generate the clear subprogram. This model has one specification panel, Standard Parameters.

Standard Parameters Panel

CUGCMA	CST-Clear Subprogram	CUG-MA0
Aug 17	Standard Parameters	1 of 1
Module name	CXMNC____	
Parameter data area	CXMNPDA_ *	
Title	Clear ..._____	
Description	Clear specification Parameters ..._____	

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

The input fields on the Standard Parameters panel are:

Field	Description
Module name	Name specified on the Generation main menu. The name of the clear subprogram must be alphanumeric and no more than eight characters in length. Use the following naming convention: CXxxC where <i>xx</i> uniquely identifies your model.
Parameter data area	Name of the parameter data area (PDA) for your model. Natural Construct determines the name of the PDA based on the Module name specified on the Generation main menu. For example, if you entered CXMNC as the name of the clear subprogram, Natural Construct assumes the name of the PDA is CXMNPDA. Use the following naming convention: CXxxPDA where <i>xx</i> uniquely identifies your model.
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-Clear Model

CSGSAMPL Aug 17	CST-Clear Subprogram User Exits	CSGSM0 1 of 1																																													
	<table border="0"> <thead> <tr> <th>User Exits</th> <th>Exists</th> <th>Sample</th> <th>Required</th> <th>Conditional</th> </tr> </thead> <tbody> <tr> <td>-----</td> <td>-----</td> <td>-----</td> <td>-----</td> <td>-----</td> </tr> <tr> <td>_ CHANGE-HISTORY</td> <td></td> <td>Subprogram</td> <td></td> <td></td> </tr> <tr> <td>_ PARAMETER-DATA</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>_ LOCAL-DATA</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>_ PROVIDE-DEFAULT-VALUES</td> <td></td> <td>Subprogram</td> <td></td> <td></td> </tr> <tr> <td>_ BEFORE-CHECK-ERROR</td> <td></td> <td>Example</td> <td></td> <td></td> </tr> <tr> <td>_ ADDITIONAL-INITIALIZATIONS</td> <td></td> <td>Example</td> <td></td> <td></td> </tr> <tr> <td>_ END-OF-PROGRAM</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	User Exits	Exists	Sample	Required	Conditional	-----	-----	-----	-----	-----	_ CHANGE-HISTORY		Subprogram			_ PARAMETER-DATA					_ LOCAL-DATA					_ PROVIDE-DEFAULT-VALUES		Subprogram			_ BEFORE-CHECK-ERROR		Example			_ ADDITIONAL-INITIALIZATIONS		Example			_ END-OF-PROGRAM					
User Exits	Exists	Sample	Required	Conditional																																											
-----	-----	-----	-----	-----																																											
_ CHANGE-HISTORY		Subprogram																																													
_ PARAMETER-DATA																																															
_ LOCAL-DATA																																															
_ PROVIDE-DEFAULT-VALUES		Subprogram																																													
_ 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*.