

Technical Information

The Transform-Browse model retrieves the specifications and user exits for a browse module and transforms this information into the specifications for the object browse modules within the program editor. The model takes advantage of the multi-model feature of Natural Construct to:

- Parse the contents of the editor to determine the specifications for each model
- Save the specifications to individual Natural modules
- Copy the information to the NCST-WORK view of the Natural Construct LFILE
- Read the NCST-WORK file to determine which Natural modules to read into the editor
- Generate and stow the modules
- Move the stowed modules to the transformed module library (using the reporting feature)

Note:

The Transform-Browse model does not support user exit processing because it only generates text specifications for the object browse modules. The model copies the user exit specifications for the browse module into user exits for the appropriate object browse module.

Tip:

If the browse module is a program, run the program to determine how many lines are used for the field headings. By default, the Transform-Browse model assumes that field headings use two lines. If this is not accurate, it will affect the number of rows displayed on the screen after transformation. For more information, see Display Correct Number of Rows.

This section covers the following topics:

- Naming Conventions for Generated Object Browse Modules
- Coding Conventions for Browse Keys
- PF-Key Styles
- Action Styles

Naming Conventions for Generated Object Browse Modules

To name the generated object browse modules, the Transform-Browse model first determines whether the module names have been specified on the Standard Parameters panel. If these names are blank, the model uses the first five characters in the name of the original browse module + a suffix + a number from 1 to 99 to make the name unique in both the current library and transformed module library (for example, if BINSN already exists in the transformed module library then BINSN1 is used).

The following table lists each module generated by the Transform-Browse model and the default suffix used in the generated module name:

Module Name	Suffix
Object browse subprogram	N
Object browse dialog	W
Object browse dialog driver	X
Object PDA	D
Key PDA	K
Restricted PDA	P

To see the naming conventions for your site, logon to the SYSCST library and execute the CTENAME driver program.

Note:

For information on changing these conventions, see *Modify/Test the Naming Conventions for Natural Objects*.

Coding Conventions for Browse Keys

To implement Browse model functionality in the Object-Browse models, the Transform-Browse model creates internal specifications that are only accessible if the browse module has been transformed. This functionality includes:

- Ascending and Descending Sort Order
- Key Prefixes
- Minimum and Maximum Key Values

Ascending and Descending Sort Order

To handle ascending and descending sort order, the object browse subprogram is generated with two logical keys: *A-keyname* and *D-keyname*. Next, an object browse dialog is generated to use either the *A-keyname* or *D-keyname* key, depending on which option a user requests. From the user's perspective, the functionality is the same as was in the browse module.

Key Prefixes

To handle the prefix concept for browse key components, the object browse subprogram specifies the leading components in the CDBRPDA parameter data area. In addition, it handles the prefix bytes by specifying a starting with value. All of this is automatically handled by the object browse dialog based on the information passed in from the browse specifications.

Minimum and Maximum Key Values

To establish a logical start and/or end of file for a browse subprogram, you must specify minimum and/or maximum key values. The minimum (starting value for the browse) and maximum (ending value for the browse) key values create a subset of records within the file. The program will not browse before or after these values.

The only way to use minimum and/or maximum key values in an object browse subprogram is via the Transform-Browse model as this option is only available for the first two keys (where the first one is a physical key in logical ascending order and the second one is a physical key in logical descending order). In addition, the object browse subprogram only generates code for the input criteria for the first and second key when minimum and/or maximum key values are specified in the original browse module.

Note:

If these options were not specified in the original browse module, the object browse subprogram generates the input criteria for all keys.

To allow greater flexibility when using minimum and/or maximum key values for additional keys, the READ-INPUT-CRITERIA user exit is available for the Object-Browse-Subprogram model. You can use this exit to write code similar to the first and second key for any additional keys. For information, see *READ-INPUT-CRITERIA, Natural Construct Generation*.

PF-Key Styles

By default, a generated object browse dialog module and browse module use different styles of PF-keys. To solve this problem, the Transform-Browse model generates the object browse dialog module with the browse style of PF-keys.

During transformation, the Transform-Browse model creates the following specification for the dialog:

```
**SAG USE-BROWSE-PFKEYS: X
```

Note:

If the object browse dialog module uses the browse-style PF-keys, the PF6 (pfkey) option is removed from the Standard Parameters panel (accessed using the NCSTG command) for the dialog. If the module does not use browse-style PF-keys, PF6 (pfkey) is available on the panel.

For information on changing the default style of PF-keys for an object browse dialog module, see the *Change the Default PF-key Style* section in Object-Browse-Dialog Model.

Action Styles

By default, a transformed object browse dialog module has one of the following action styles:

- no actions

If a browse module was transformed, the dialog has no actions.

- browse-select style actions

If a browse-select module was transformed, the dialog uses the browse-select style actions (instead of the object browse dialog-style actions).