Customizing Natural Business Services

This section describes how to customize the default behavior of Natural Business Services. The following topics are covered:

- Modify/Test the Naming Conventions for Natural Objects
- Define Default Specification Parameters
- Modify Logon Security
- Modify the BACKOUT or END TRANSACTION Processing
- Modify the Supplied Generation Models

Modify/Test the Naming Conventions for Natural Objects

This section describes how the Natural Business Services wizards create names for the Natural objects they generate and how to include your own naming conventions.

Since one Natural Business Services wizard may generate several different Natural objects, each type of object must be uniquely identified. For example, the Business Service wizard creates at least one subprogram that handles logic and one subprogram proxy.

The different styles and their unique identifiers are stored in an LDA called WTRNAMEL. Natural Business Services names the modules as follows:

- One character for the object style and one character for the category

  To change these characters, modify WTRNAMCL and then recompile CTENAMEL and WTRNAME.

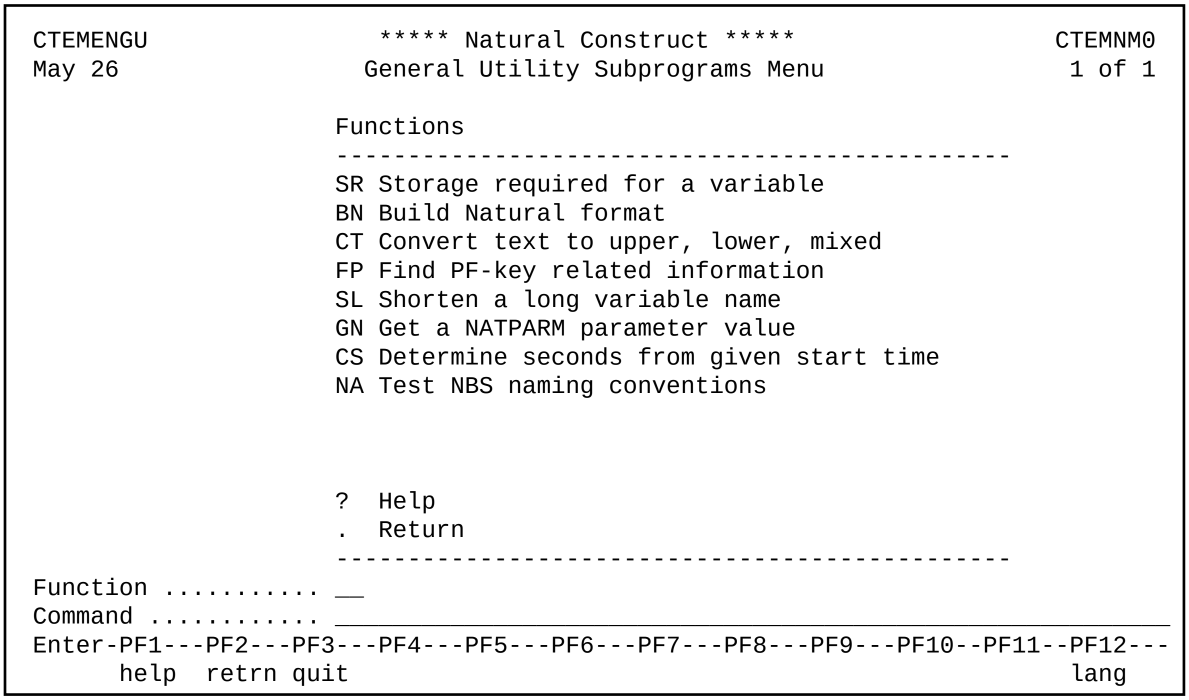
- Five characters supplied by you
- Optionally, a number to help retain uniqueness within a library

How the unique identifier is used, where it is used, and how unique Natural object names are derived is handled in WTRNAME. To change the location of the category, style, and/or five characters, modify WTRNAME and then test the generation of Natural module names.

To test the generation of Natural module names:

1. Enter the "menu D A GU" direct command.

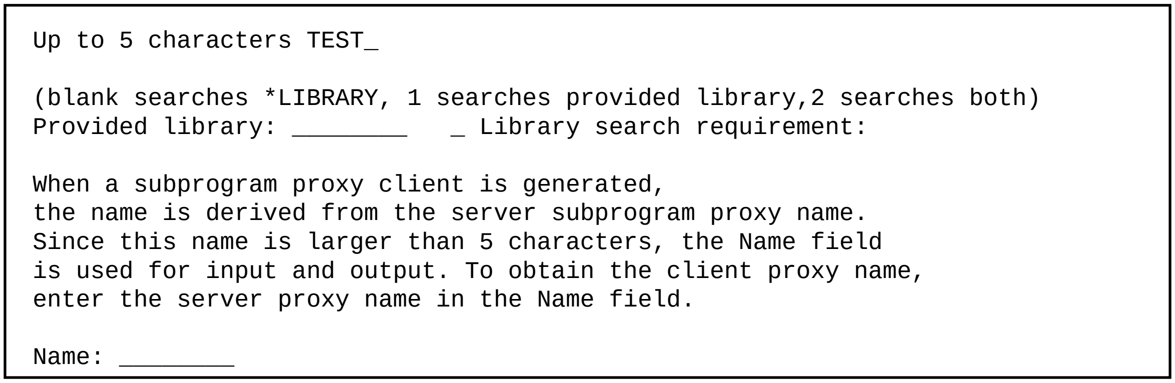
   The General Utility Subprograms menu is displayed. For example:



```
CTEMENGU                ***** Natural Construct *****                  CTEMNM0
May 26                 General Utility Subprograms Menu                 1 of 1

                     Functions
                     -----------------------------------------------
                     SR Storage required for a variable
                     BN Build Natural format
                     CT Convert text to upper, lower, mixed
                     FP Find PF-key related information
                     SL Shorten a long variable name
                     GN Get a NATPARM parameter value
                     CS Determine seconds from given start time
                     NA Test NBS naming conventions



                     ?  Help
                     .  Return
                     -----------------------------------------------
Function ........... __
Command ............ __________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit                                                  lang
```

2. Enter "NA" in Function on the General Utility Subprograms menu.

   The following screen is displayed. For example:



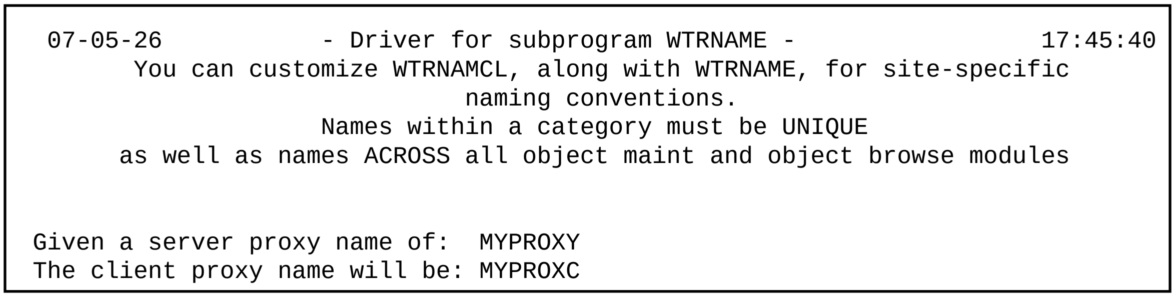
```
Up to 5 characters TEST_

(blank searches *LIBRARY, 1 searches provided library,2 searches both)
Provided library: ________   _ Library search requirement:

When a subprogram proxy client is generated,
the name is derived from the server subprogram proxy name.
Since this name is larger than 5 characters, the Name field
is used for input and output. To obtain the client proxy name,
enter the server proxy name in the Name field.

Name: ________
```

   By default, TEST is used for the example. Change the default characters and specify the name of the library (if not the current library).

   - To search the current library, leave the Provided library and Library search requirement fields blank.
   - To search the Provided library, type the library name in Provided library and type "1" in the Library search requirement field.
   - To search both libraries, type the library name in Provided library and type "2" in the Library search requirement field.

3. Enter the name of the server proxy in Name.

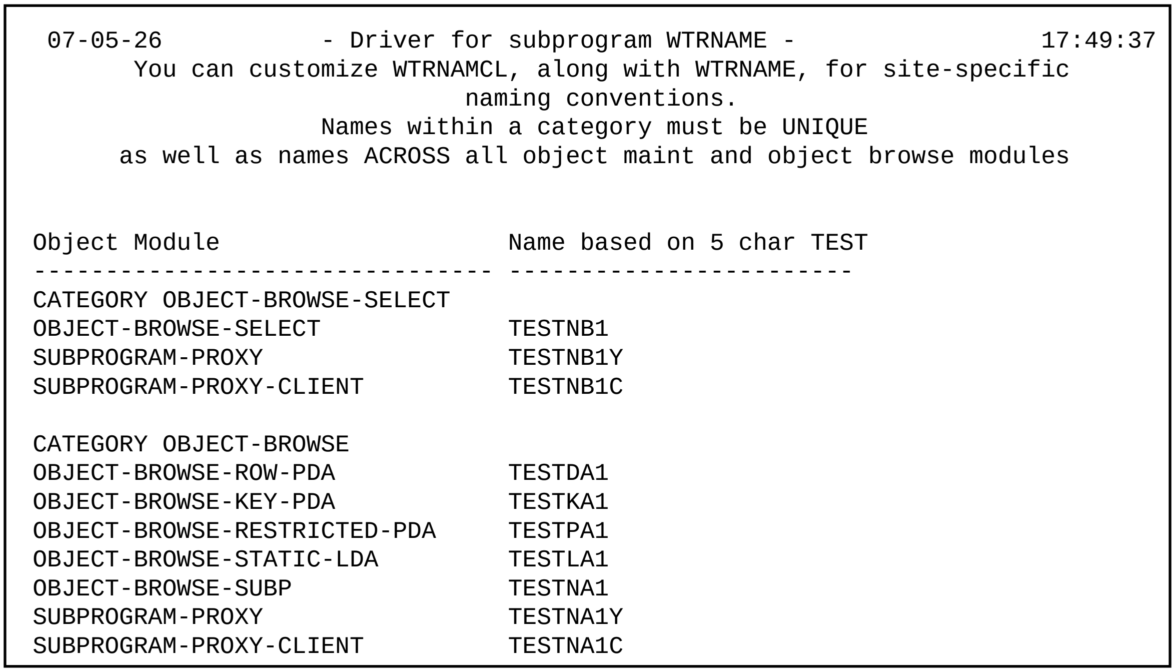
Natural Business Services displays the default name of the client proxy. For example, if you entered "MYPROXY" in Name, the following is displayed:



```
 07-05-26           - Driver for subprogram WTRNAME -                 17:45:40
       You can customize WTRNAMCL, along with WTRNAME, for site-specific
                              naming conventions.
                    Names within a category must be UNIQUE
      as well as names ACROSS all object maint and object browse modules


Given a server proxy name of:  MYPROXY
The client proxy name will be: MYPROXC
```

4. Press Enter.

   The following screen is displayed:

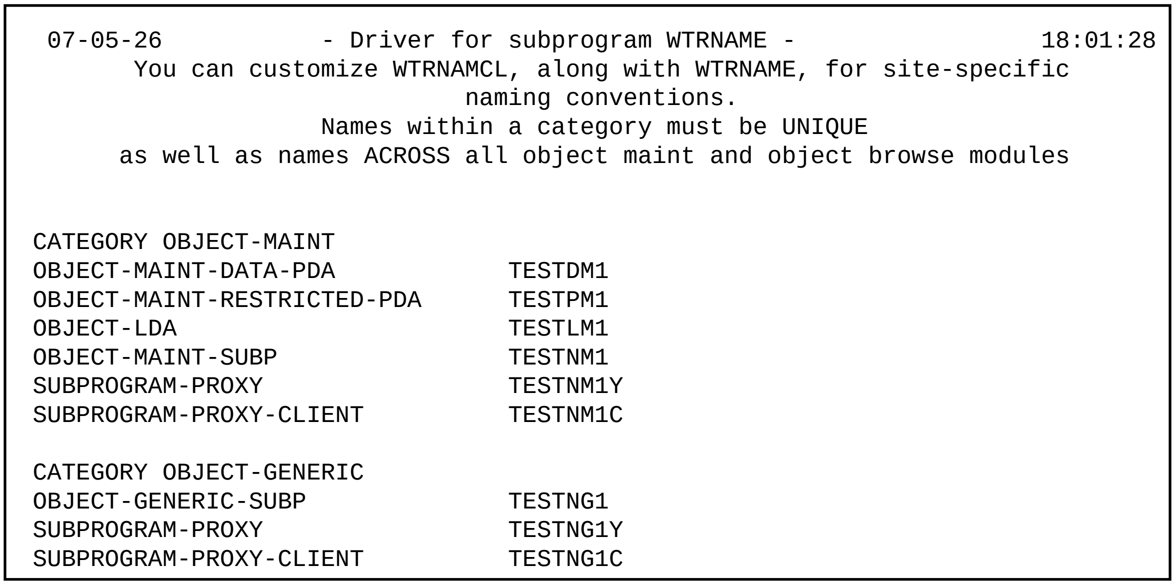


```
 07-05-26           - Driver for subprogram WTRNAME -                 17:49:37
       You can customize WTRNAMCL, along with WTRNAME, for site-specific
                              naming conventions.
                    Names within a category must be UNIQUE
      as well as names ACROSS all object maint and object browse modules


Object Module                    Name based on 5 char TEST
-------------------------------- ------------------------
CATEGORY OBJECT-BROWSE-SELECT
OBJECT-BROWSE-SELECT             TESTNB1
SUBPROGRAM-PROXY                 TESTNB1Y
SUBPROGRAM-PROXY-CLIENT          TESTNB1C

CATEGORY OBJECT-BROWSE
OBJECT-BROWSE-ROW-PDA            TESTDA1
OBJECT-BROWSE-KEY-PDA            TESTKA1
OBJECT-BROWSE-RESTRICTED-PDA     TESTPA1
OBJECT-BROWSE-STATIC-LDA         TESTLA1
OBJECT-BROWSE-SUBP               TESTNA1
SUBPROGRAM-PROXY                 TESTNA1Y
SUBPROGRAM-PROXY-CLIENT          TESTNA1C
```

   This screen displays the default names for the different models.

5. Press Enter again.

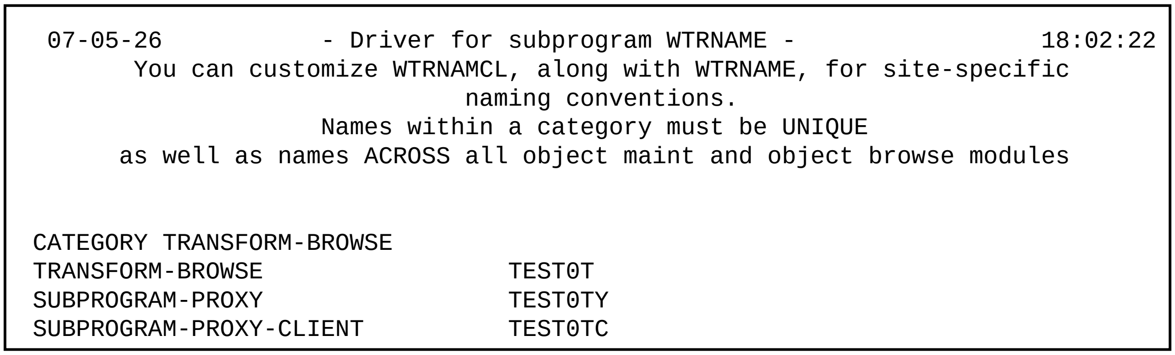
   The following screen is displayed:



```
 07-05-26           - Driver for subprogram WTRNAME -                 18:01:28
       You can customize WTRNAMCL, along with WTRNAME, for site-specific
                             naming conventions.
                   Names within a category must be UNIQUE
      as well as names ACROSS all object maint and object browse modules


CATEGORY OBJECT-MAINT
OBJECT-MAINT-DATA-PDA            TESTDM1
OBJECT-MAINT-RESTRICTED-PDA      TESTPM1
OBJECT-LDA                       TESTLM1
OBJECT-MAINT-SUBP                TESTNM1
SUBPROGRAM-PROXY                 TESTNM1Y
SUBPROGRAM-PROXY-CLIENT          TESTNM1C

CATEGORY OBJECT-GENERIC
OBJECT-GENERIC-SUBP              TESTNG1
SUBPROGRAM-PROXY                 TESTNG1Y
SUBPROGRAM-PROXY-CLIENT          TESTNG1C
```

6. Press Enter again.

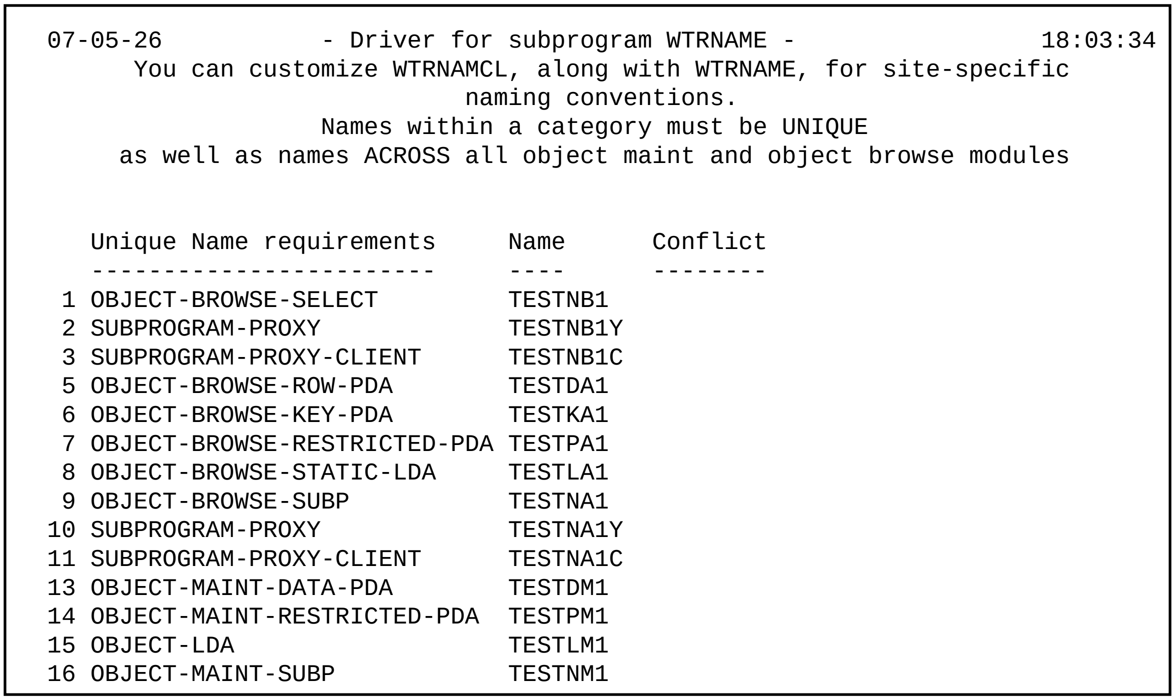
   The following screen is displayed:



```
 07-05-26           - Driver for subprogram WTRNAME -                 18:02:22
       You can customize WTRNAMCL, along with WTRNAME, for site-specific
                             naming conventions.
                   Names within a category must be UNIQUE
      as well as names ACROSS all object maint and object browse modules


CATEGORY TRANSFORM-BROWSE
TRANSFORM-BROWSE                 TEST0T
SUBPROGRAM-PROXY                 TEST0TY
SUBPROGRAM-PROXY-CLIENT          TEST0TC
```

7. Press Enter again.

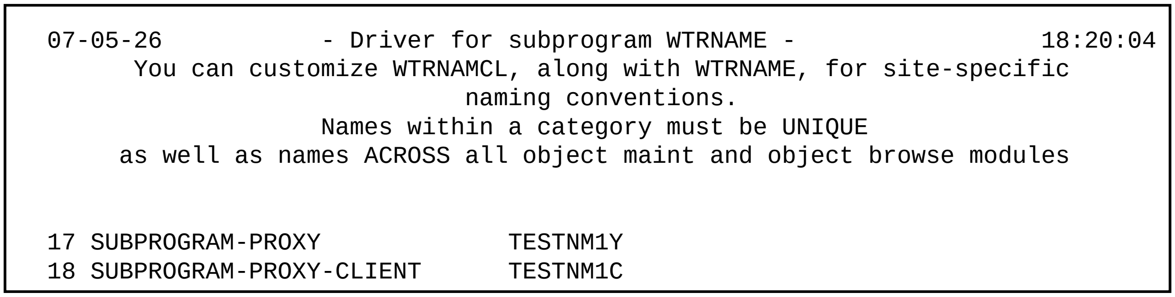
   The following screen is displayed, showing any conflicts in the naming conventions:



```
07-05-26           - Driver for subprogram WTRNAME -                 18:03:34
      You can customize WTRNAMCL, along with WTRNAME, for site-specific
                              naming conventions.
                    Names within a category must be UNIQUE
     as well as names ACROSS all object maint and object browse modules


   Unique Name requirements     Name      Conflict
   ------------------------     ----      --------
 1 OBJECT-BROWSE-SELECT         TESTNB1
 2 SUBPROGRAM-PROXY             TESTNB1Y
 3 SUBPROGRAM-PROXY-CLIENT      TESTNB1C
 5 OBJECT-BROWSE-ROW-PDA        TESTDA1
 6 OBJECT-BROWSE-KEY-PDA        TESTKA1
 7 OBJECT-BROWSE-RESTRICTED-PDA TESTPA1
 8 OBJECT-BROWSE-STATIC-LDA     TESTLA1
 9 OBJECT-BROWSE-SUBP           TESTNA1
10 SUBPROGRAM-PROXY             TESTNA1Y
11 SUBPROGRAM-PROXY-CLIENT      TESTNA1C
13 OBJECT-MAINT-DATA-PDA        TESTDM1
14 OBJECT-MAINT-RESTRICTED-PDA  TESTPM1
15 OBJECT-LDA                   TESTLM1
16 OBJECT-MAINT-SUBP            TESTNM1
```

8. Press Enter again.

   The last screen is displayed:



```
07-05-26           - Driver for subprogram WTRNAME -                 18:20:04
      You can customize WTRNAMCL, along with WTRNAME, for site-specific
                              naming conventions.
                    Names within a category must be UNIQUE
     as well as names ACROSS all object maint and object browse modules


17 SUBPROGRAM-PROXY             TESTNM1Y
18 SUBPROGRAM-PROXY-CLIENT      TESTNM1C
```

Define Default Specification Parameters

You can define default parameter values for the model specification panels in Natural Construct. The supplied models retrieve many of the default values by issuing a CALLNAT to the CSUDEFLT subprogram. Prior to returning the defaults, CSUDEFLT checks to see whether the values have been overridden by the user-defined CSXDEFLT sample exit subprogram. If so, the overridden values are returned to the model. When the Business Service wizard generates the models to create business services, your default values will be used.

The CSXDEFLT sample exit subprogram provides user-defined default values for model parameters that can be overridden on the specification panels, as well as internal model parameters that are not displayed on the panels.

Note:
For more information, see Define Default Specifications.

Modify Logon Security

The CSXSECX sample exit subprogram allows you to take advantage of Natural Security LOGON user exit. If the security routines in CSXSECX are defined and the object code is in the SYSLIBS library, they are processed before the standard Natural Business Services security routines are processed.

The parameters for CSXSECX are:

```
1 STEPLIB-DATA   (A144)
1 REDEFINE STEPLIB-DATA
  2 STEPLIB-DATA-STRUCT  (1:8)
   3 LIB-NAME (A8)
   3 LIB-DBID (N5)
   3 LIB-FNR (N5)
1 PROXY-SUBPROGRAM (A8)
1 USER-ID (A32)
1 REDEFINE USER-ID
 2 USER-ID-A8 (A8)
1 DOMAIN (A8)   /* Domain name
1 BUSINESS-SERVICE (A32) /* business service name
1 BUSINESS-SERVICE-VERSION (N6) /* business service version
1 BUSINESS-METHOD (A32) /* business service version
```

Note:
Although the parameters above are accessible within this exit, you cannot change the parameter values here; you can only add additional security checks.

To customize CSXSECX:

1. Copy the subprogram from the SYSCSTX library to the SYSCST library.
2. Add your own security settings.
3. Stow the subprogram.
4. Move the object code to the SYSLIBS library to make the routine active.

When the dispatcher is started, it checks to see if the object code for CSXSECX is in the SYSLIBS library. If the object code is not in the SYSLIBS library, the dispatcher assumes there is no customized security checking and will process the standard Natural Business Service security routines. If the object code is in the SYSLIBS library, the dispatcher will execute this code first and then process the standard security routines.

For information on the sample exit subprograms, see Access and Use the Sample Exit Subprograms.

Note:
You can also enhance the standard Natural Business Services security routines. For information, see Setting Business Service Security Options.

Modify the BACKOUT or END TRANSACTION Processing

The CSXTRANS sample exit subprogram allows you to issue BACKOUT or END TRANSACTION statements. If the object code for CSXTRANS is in the SYSLIBS library, it is executed after the call to a business service has been made.

Note:
An Object Maint-generated subprogram issues an END TRANSACTION statement when CDAOBJ2. #ET-IF-SUCCESSFUL is True. Therefore, a BACKOUT TRANSACTION statement may not be issued when expected.

The parameters for CSXTRANS are:

```
1 STEPLIB-DATA (A144)
1 REDEFINE STEPLIB-DATA
        2 STEPLIB-DATA-STRUCT (1:8)
                3 LIB-NAME (A8)
                3 LIB-DBID (N5)
                3 LIB-FNR (N5)
1 PROXY-SUBPROGRAM (A8)
1 USER-ID (A32)
1 REDEFINE USER-ID
        2 USER-ID-A8 (A8)
1 DOMAIN (A8) /* Domain name
1 BUSINESS-SERVICE (A32) /* business service name
1 BUSINESS-SERVICE-VERSION (N6) /* business service version
1 BUSINESS-METHOD (A32) /* business service version
1 START-OF-PROXY-TIMESTAMP (B8) /* time CALLNAT to business service was issued
**SAG END-EXIT
PARAMETER USING CDPDA-M
```

Note:
Changing any value except the values in CDPDA-M will have no effect; the values are reset to the values they were before entering this exit.

To customize CSXTRANS:

1. Copy the subprogram from the SYSCSTX library to the SYSCST library.
2. Add your own BACKOUT or END TRANSACTION logic.
3. Stow the subprogram.
4. Move the object code to the SYSLIBS library to make the routine active.

When the dispatcher is started, it checks to see if the object code for CSXTRANS is in the SYSLIBS library. If the object code is not in the SYSLIBS library, the dispatcher assumes there is no customized END or BACKOUT TRANSACTION processing and will process the standard Natural Business Service security routines. If the object code is in the SYSLIBS library, the dispatcher will execute this code first and then process the standard security routines.

For information on the sample exit subprograms, see Access and Use the Sample Exit Subprograms.

Note:
You can also enhance the standard Natural Business Services security routines. For information, see Setting Business Service Security Options.

Modify the Supplied Generation Models

You can customize the business services generated by the Business Service wizard by modifying the Natural Construct generation models. You can modify the code frames, subprograms, and copycode used by the supplied models, as well as the external data areas and subprograms.

Note:
For more information, see Modifying the Supplied Models.