Using SYSMAIN with Subprogram

The MAINUSER subprogram is an Application Programming Interface, which allows you to perform SYSMAIN utility functions from any user-written object (subroutine, program or subprogram) as an alternative to using SYSMAIN utility menus. Upon completion of the SYSMAIN function, the utility is terminated and control is returned to the object from which the request was issued. MAINUSER can be used in either online or batch mode. An example of a callable routine is the MAINCALL program, which is supplied in the SYSMAIN system library.

This section provides instructions for using MAINUSER and the syntax that applies when specifying commands for executing SYSMAIN utility functions.

- Invoking and Executing MAINUSER

- Using Commands

- LIST and FIND Command Syntax

- COPY and MOVE Command Syntax

- DELETE Command Syntax

- RENAME Command Syntax

- IMPORT Command Syntax

- where-clause

- with-clause

- Keywords and Variables in Commands

Invoking and Executing MAINUSER

▶ **To invoke and execute MAINUSER**

- Issue a `CALLNAT` statement that contains the following syntax elements:

  `CALLNAT 'MAINUSER' command error message library`

  where the variable values denote the following parameters:

Parameter	Natural Data Format/Length	Explanation
command	A250	The command string to be executed by SYSMAIN: see *Using Commands*.
error	N4	The return code issued by SYSMAIN at the end of processing to indicate a normal end of processing or an error.
message	A72	The message corresponding to the error given online.
library	A8	The name of the library containing the utility SYSMAIN; by default, this is the library SYSMAIN. (This parameter is provided for compatibility reasons only.)

Using Commands

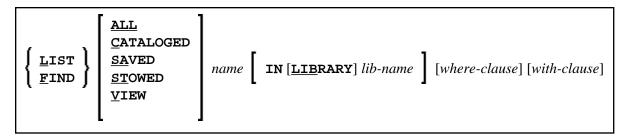SYSMAIN functions can be executed by using commands issued as a parameter of the MAINUSER subprogram.

A *command* consists of keywords and variable values. For each SYSMAIN function to be performed, the keywords and variable values are shown in the corresponding syntax diagrams below and explained in the section *Keywords and Variables in Commands*. The symbols in the syntax diagrams correspond to the syntax symbols used for system commands. These symbols are explained in *System Command Syntax* in the *System Commands* documentation.

The sequence of the command syntax is not completely fixed. The following rules apply:

- SYSMAIN function, object type and object name must be the first three parameters of the command string.

- A period (.) indicates the end of a command. If this character is detected anywhere within a command string, all subsequent data is ignored.

- In the syntax diagrams, FM or IN is shown instead of the FROM keyword to make the diagrams easier to read; however, FROM can always be used as a synonym for FM or IN and vice versa.

- The syntax of the *where-clause* and the *with-clause* is identical for each command.

LIST and FIND Command Syntax

The following command syntax applies to the list and find functions:

```
            ┌ ALL       ┐
            │ CATALOGED │
 ┌ LIST ┐   │ SAVED     │          ┌              ┐
 │      │   │ STOWED    │ name     │ IN [LIBRARY] lib-name │ [where-clause] [with-clause]
 └ FIND ┘   │ VIEW      │          └              ┘
            └           ┘
```

Examples of LIST and FIND

```
LIST VIEW * IN TESTLIB
```

```
L SAVED TEST* IN TESTLIB TYPE PNS FNR 6
```

```
L SA TEST* IN TESTLIB FNR 6 DBID 2 TYPE PM FMDATE 2007-01-01
```

```
FIND PROG1 IN * DBID 1 FNR 6
```

```
F STOWED MAINMENU IN SYS* WHERE DBID 1 FNR 5
```

```
FIND ALL PROG2 IN PROD* FNR 27 DBID 1
```

COPY and MOVE Command Syntax

The following command syntax applies to the copy and move functions:

```
                  ┌─────────────┐
                  │ ALL         │
                  │ CATALOGED   │
{ COPY }          │ SAVED       │  name  ┌ FM [LIBRARY] lib-name ┐ [where-clause]
{ MOVE }          │ STOWED      │        └                      ┘
                  │ VIEW        │
                  │ RESOURCE    │          TO [LIBRARY] lib-name   [where-clause] [with-clause]
                  └─────────────┘
```

Examples of COPY and MOVE

```
COPY PROG1 FM TESTORD TO ORDERS DBID 1 FNR 6 REP
```

```
C PGM* FM TESTLIB TO PRODLIB WITH REP TYPE PNS
```

```
C VIEW PERS FM OLDLIB FNR 10 TO NEWLIB FNR 16 REPLACE
```

```
MOVE VIEW PERSONNEL FM OLDLIB FNR 20 TO NEWLIB FNR 24
```

```
M PROG1 TO NEWLIB
```

```
M STOWED * FM OLDLIB TO NEWLIB WHERE DBID 100 FNR 160 WITH XREF Y
```

DELETE Command Syntax

The following command syntax applies to the delete function:

```
          ┌                ┐
          │ ALL            │
          │ CATALOGED      │
          │ SAVED          │                      ┌                          ┐
DELETE    │ STOWED         │  name                │ IN [LIBRARY] lib-name    │  [where-clause] [with-clause]
          │ VIEW           │                      └                          ┘
          │ RESOURCE       │
          └                ┘
```

Examples of DELETE

```
DELETE SA * IN LIBTEST TYPE GLA
```

```
D * IN TESTORD TYPE PM
```

```
D VIEW FINANCE IN TESTLIB DBID 12 FNR 27
```

RENAME Command Syntax

The following command syntax applies to the rename function:

```
          ┌                ┐
          │ ALL            │
          │ CATALOGED      │
          │ SAVED          │   name  AS new-name [with-clause]
RENAME    │ STOWED         │
          │ VIEW           │         FM [LIBRARY] lib-name [where-clause]
          │ RESOURCE       │
          └                ┘         TO [LIBRARY] lib-name [where-clause]
```

Examples of RENAME

```
RENAME PGM1 AS PROG1
```

```
R PGM1 AS PROG1 FM TESTLIB DBID 1 FNR 5 TO PRODLIB DBID 2 FNR 6
```

IMPORT Command Syntax

The command syntax that applies to the import function is shown in the following section.

For the points that must be considered before importing objects, see the description of the import function.

```
          ┌                ┐
          │ ALL            │
          │ CATALOGED      │
          │ SAVED          │
IMPORT    │ STOWED         │  name  FM [PATH] path-name
          │ VIEW           │
          │ RESOURCE       │       TO [LIBRARY] lib-name [where-clause] [with-clause]
          └                ┘
```

Examples of IMPORT

```
IMPORT ALL PGM* FM D:[NAT-PROGRAMS] TO IMP-LIB
```

```
I RES res1.bmp FM D:[RESOURCES] TO IMP-LIB
```

where-clause

```
[WHERE] [DBID dbid] [FNR fnr]

   [DIC (dbid,fnr,password,cipher)]

   [SEC (dbid,fnr,password,cipher)]
```
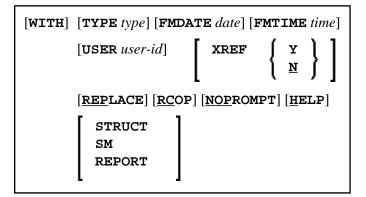
Separators

Commas must be used as separators between the values following the DIC and SEC keywords, or if a value is missing. For example: DIC (10,,secret,2a). If the ID session parameter (see also *ID - Input Delimiter Character* in the *Parameter Reference*) has been set to a comma, use a slash (/) as the separator between values.

with-clause

```
[WITH] [TYPE type] [FMDATE date] [FMTIME time]

      [USER user-id]   ⎡ XREF  ⎧ Y ⎫ ⎤
                       ⎣       ⎩ N ⎭ ⎦

      [REPLACE] [RCOP] [NOPROMPT] [HELP]
      ⎡ STRUCT  ⎤
      ⎢ SM      ⎥
      ⎣ REPORT  ⎦
```

Keywords and Variables in Commands

This section explains the keywords and corresponding variable values (if required) used in a command.

Keywords are listed alphabetically. Letters in italics represent variable values that must be supplied with a keyword. For each variable value, the Natural data format and length is indicated.

Keyword	Value	Natural Data Format/ Length	Explanation
ALL	*name*	A9	Only applies to programming objects. The name of the object to be processed or a range of names; see *Specifying a Range of Names*. Any saved (source) objects and/or cataloged objects are processed.
CATALOGED	*name*	A9	Only applies to programming objects. The name of the cataloged object to be processed or a range of names; see *Specifying a Range of Names*.
SAVED	*name*	A9	Only applies to programming objects. The name of the saved (source) object to be processed or a range of names; see *Specifying a Range of Names*.
STOWED	*name*	A9	Only applies to programming objects. The name of an object (or a range of names) for which the saved (source) *and* the cataloged object are to be processed (see also *Specifying a Range of Names*). Only an object that exists as both a saved (source) object *and* a cataloged object is processed. The exceptions to this are copycode and text, neither of which can be cataloged. However, they are included in processing when this option is specified.

Keyword	Value	Natural Data Format/ Length	Explanation
VIEW	*name*	A32	Only applies to DDMs. The name of the DDM to be processed or a range of names; see *Specifying a Range of Names*.
RESOURCE	*name*	A255	Only applies to shared resources. The name of the shared resource to be processed or a range of names; see *Specifying a Range of Names*.
FROM or FM or IN	*lib-name* or *path-name*	A8 or A253	Specifies a source library or a source path. The source library or path contains the object to be processed.
TO	*lib-name*	A8	Specifies a target library.
AS	*new-name*	A8 or A32 or A255	The new name to be given to an object when it is renamed with the RENAME command. Format/length A8 applies to programming objects, A32 to DDMs and A255 to shared resources.
LIBRARY	*lib-name*	A8	An optional keyword that indicates the name (*lib-name*) of a source or a target library. If you omit the keyword and respective value, the library where you logged on before you invoked SYSMAIN is used for processing. The source library contains the object to be processed. The target library is the library to which the object is to be copied or moved, or where the object is renamed. *lib-name* must be specified immediately after the FROM/FM/IN or TO keyword. If LIBRARY is used, it must be entered between FROM/FM/IN or TO and *lib-name*.
PATH	*path-name*	A253	Only applies to the IMPORT command. An optional keyword that indicates the name (*path-name*) of a source path. For a valid path name, see **PATH** in *Using the Fields in an Object-Specification Window*. *path-name* must be specified immediately after the FROM/FM/IN or TO keyword. If PATH is used, it must be entered between FROM/FM/IN or TO and *path-name*.

Keyword	Value	Natural Data Format/ Length	Explanation
WHERE	*where-clause*	-	An optional keyword that indicates the start of a *where-clause*. The *where-clause* must always follow the FROM/FM/IN or TO keyword and the library name (*lib-name*) or path name (*path-name*) if relevant; the sequence of the keywords and values within the clause can be specified in any order.
DBID	*dbid*	N5	The database ID (DBID) of a source or a target system file. The source system file contains the object to be processed. The target system file is the system file to which the object is to be copied or moved, or where the object is renamed if relevant. Valid DBIDs are 1 to 65535. If no DBID or FNR (file number) is specified, the following applies: The DBID and FNR of the system file where the current library resides are always used. For example: if you specify a library contained in the FUSER system file, the DBID and FNR of this file are used.
FNR	*fnr*	N5	The file number (FNR) of a source or a target system file. The source system file contains the object to be processed. The target system file is the system file to which the object is to be copied or moved, or where the object is renamed if relevant. Valid FNRs are 1 to 65535. If no DBID (database ID) or FNR is specified, the following applies: The DBID and FNR of the system file where the current library resides are always used. For example: if you specify a library contained in the FUSER system file, the DBID and FNR of this file are used.
DIC	*dbid* *fnr* *password* *cipher*	A80	Specifies the environment of the FDIC source and/or target system file: database ID (*dbid*), file number (*fnr*), Adabas password (*password*) and Adabas cipher code (*cipher*).
SEC	*dbid* *fnr* *password* *cipher*	A80	Specifies the environment of the FSEC source and/or target system file: database ID (*dbid*), file number (*fnr*), Adabas password (*password*) and Adabas cipher code (*cipher*).

Keyword	Value	Natural Data Format/ Length	Explanation
WITH	*with-clause*	-	An optional keyword that indicates the start of a *with-clause*. The keywords and values of the *with-clause* can be specified in any order, and the *with-clause* can be placed in any location within the command string, except in the first three positions.
TYPE	*type*	A20	The type(s) of object to be processed as listed in *TYPE Specification* below.
FMDATE	*date*	A10	The start date of a time period: All objects which were saved or cataloged on or after the specified date are processed. A date must be specified in a valid Natural date format. The default format is the international format *YYYY-MM-DD* (*YYYY* = year, *MM* = month, *DD* = day), for example, `2007-05-20`.
FMTIME	*time*	A5	Only applies if FMDATE is specified. Specifies a start time: All objects which were saved or cataloged at or after the specified time (and date) are processed. A time must be specified in the format *HH:II* (*HH* = hours, *II* = minutes), for example, `11:33`.
USER	*user-id*	A8	A user ID: All objects that were saved or cataloged by the specified user are processed.

Keyword	Value	Natural Data Format/ Length	Explanation
XREF	N or Y	A1	Only applies to programming objects and if Predict is installed. Indicates whether cross-reference (XRef) data stored on Predict system files is to be processed. You can specify one of the following values: N XRef data is not processed, except when using the `DELETE` command. If a cataloged object is deleted, SYSMAIN always deletes any existing XRef data for this object. Y All XRef data is processed. See also *XRef Considerations*.
REPLACE	-	-	Activates the replace option used in a *with-clause*. An object is automatically replaced. See also *Using the Replace Option*.
RCOP	-	-	Specifies that a copy of the object being renamed is to be made.
NOPROMPT	-	-	Not applicable in batch mode. Disables (`NOPROMPT`) the SYSMAIN prompts. With `NOPROMPT`, no confirmation screen is displayed. For example, before any deletion, SYSMAIN prompts you for confirmation.
HELP	-	-	This keyword is provided for compatibility reasons only.
STRUCT or SM			Only applies to the `IMPORT` command. Indicates structured mode described in *Natural Programming Modes* in the *Programming Guide*.

Keyword	Value	Natural Data Format/ Length	Explanation
REPORT			Only applies to the IMPORT command. Indicates reporting mode described in *Natural Programming Modes* in the *Programming Guide*.
.	-	-	A period (.) indicates the end of a command. If this character is detected anywhere within a command string, all subsequent data is ignored.

TYPE Specification

The following table lists all valid object-type codes for programming objects that can be used with the TYPE keyword:

Code	Object Type
P	Program
N	Subprogram
S	Subroutine
M	Map
H	Helproutine
3	Dialog
5	Processor
A	Parameter data area
G	Global data area
L	Local data area
C	Copycode
T	Text
4	Class
7	Function
V	View (DDM)
8	Adapter
*	All programming object types