

User Exit Considerations by Type of Exit

Each Com-plete user exit is classified as a batch, thread, or nucleus exit. Each type of exit has unique attributes and considerations. The conventions table for each exit specifies the type of exit.

This chapter covers the following topics:

- Batch Exits
 - Thread Exits
 - Nucleus Exits
-

Batch Exits

Batch exits are called from Com-plete batch utility programs. The exit can perform most operation system functions available to a batch program. Note that batch exits should not perform any input/output functions on files defined for that utility.

Note:

MVS linkage editor overlays are not permitted.

Abnormal termination within a batch exit will cause the Com-plete batch utility to terminate abnormally.

Thread Exits

Thread exits, called from Com-plete online utilities, execute as extensions of these utilities. The exit is restricted to using storage defined when the online utilities were cataloged to Com-plete. Thread exits are called in the PSW-protect key of the thread.

Note:

MVS linkage editor overlays are not permitted.

Thread exits can use Com-plete functions defined in the Com-plete Application Programmer's documentation. Storage can be acquired only from the region of the thread defined when the online utility was cataloged. GETMAIN/GETVIS requests are satisfied from the thread's region. The online utility may need to be recataloged with a larger region size.

If the exit is thread locked, its caller must also be thread locked.

A thread exit can also be used by specifying the RESIDENTPAGE sysparm.

Note that exits placed in either resident area must be reentrant.

When exits are specified as being RESIDENTPAGE, the exit is entered in the AMODE it was linked with for the operating systems. In this way, thread exits can be run in 31 bit mode and can also reside above the 16 MB line. This means they can be linked with RMODE=ANY. Exits must always return to the caller in the addressing mode they were called in. When the exit is called, register 14 will have the caller's AMODE set as standard.

Abnormal terminations within a thread exit cause the online utility to terminate abnormally.

Nucleus Exits

Nucleus exits, called from the Com-plete nucleus, execute as an extension of that nucleus. GETMAIN/GETVIS requests are satisfied from the Com-plete partition/address space, not from the thread. The nucleus exits execute in the PSW-protect key of Com-plete and in supervisor state. If the exit must refer to areas within a thread, those areas will be in the protect key of the thread during the time the exit has control.

Note that nucleus exits must be reentrant.

Note:

MVS linkage editor overlays are not permitted.

Nucleus exits must not use any Com-plete function defined in the Com-plete Application Programmer's documentation.

Nucleus exits are loaded dynamically. Refer to the description of each nucleus exit in the following sections for additional information.

As nucleus exits are now loaded using standard MVS / VSE loads, on XA and ESA systems the modules are loaded and entered according to their AMODE and RMODE specification. Please note that exits must return to the caller in the caller's AMODE. This is set as standard in the high-order byte of register 14 when the exit is initially called. Return is correct if the following instruction is used:

```
BSM R0,R14
```

This tells the machine to return to the address in R14 while setting AMODE 31 if the high-order bit is on, or AMODE 24 if the high-order bit is off. This will have been set by the caller of the exit.

Abnormal terminations within a nucleus exit cause the Com-plete nucleus to terminate abnormally.

Note that when a nucleus exit routine gets control, the storage key of the user program area of the thread may be in a different key to Com-plete's key. If the exit needs to change the contents of any fields in the user program area, they must do this bearing this fact in mind. The various nucleus exits which may be affected by this have appropriate notes in the Considerations sections of their individual description in this chapter.