

---

## STORAGE — Obtain and Release Storage

---

### Description

The STORAGE macro requests that the system obtain or release an area of virtual storage in the primary address space. The two functions of the macro are:

- STORAGE OBTAIN, which obtains virtual storage in an address space
- STORAGE RELEASE, which releases virtual storage in an address space.

If you use STORAGE OBTAIN to request real storage backing above 2 gigabytes, but your system does not support 64-bit storage, your request will be treated as a request for backing above 16 megabytes, even on earlier releases of OS/390 that do not support backing above 2 gigabytes. However, boundary requirements indicated by the CONTBDY and STARTBDY parameters will be ignored by earlier releases of OS/390.

### Environment

The requirements on the caller are:

<b>Minimum authorization:</b>	For subpools 0-127: problem state and PSW key 8-15. For subpools 131 and 132: a PSW key mask (PKM) that allows the calling program to switch its PSW key to match the key of the storage to be obtained or released.
<b>Dispatchable unit mode:</b>	Task
<b>Cross memory mode:</b>	Any PASN, any HASN, any SASN
<b>AMODE:</b>	24- or 31-bit
<b>ASC mode:</b>	Primary or AR
<b>Interrupt status:</b>	Enabled for I/O and external interrupts
<b>Locks:</b>	No locks held.
<b>Control parameters:</b>	No requirement.

### Programming Requirements

None.

### Restrictions

None.

### Register Information

Register usage varies depending on the type of STORAGE request. For specific information, see the descriptions of STORAGE OBTAIN and STORAGE RELEASE.

### Performance Implications

None.

---

### OBTAIN Option of STORAGE

The STORAGE macro with the OBTAIN parameter requests that the system allocate an area of virtual storage to the active task. Each virtual storage area begins on a doubleword or page boundary. The amount of storage you request must not exceed the amount available; the amount available depends on how much storage has already been allocated, and on your user region size. Valid subpools available for problem-state callers are 0 - 127, 131, and 132. When a task terminates, the system frees any storage in subpools 0 - 127 that has been allocated to the terminating task. The system does not free storage in subpools 131 and 132 until the job-step task terminates.

## STORAGE Macro

**Note:** When you obtain storage, the system clears the requested storage to zeros **only** if you obtain either:

- 8192 bytes or more from a pageable, private storage subpool
- 4096 bytes or more from a pageable, private storage subpool, with BNDRY=PAGE specified.

## Input Register Information

Before issuing the STORAGE macro with the OBTAIN parameter, the caller does not have to place any information into any register unless using it in register notation for a particular parameter, or using it as a base register.

## Output Register Information

When control returns to the caller, the general purpose registers (GPRs) contain:

Register	Contents
0	For a successful request in which maximum and minimum lengths were specified, contains the length of the storage obtained. Otherwise, used as a work register by the system.
1	When STORAGE OBTAIN is successful, contains the address of the allocated storage. Otherwise, used as a work register by the system.  <b>Note:</b> In an AMODE 64 routine, a successful STORAGE OBTAIN will return a 64-bit pointer to the obtained area (bits 0-32 will be zero).
2-13	Unchanged.
14	Used as a work register by the system.
15	For a conditional request, contains the return code. For an unconditional request, used as a work register by the system.

When control returns to the caller, the access registers (ARs) contain:

Register	Contents
0-1	Used as work registers by the system
2-13	Unchanged
14-15	Used as work registers by the system

Some callers depend on register contents remaining the same before and after issuing a service. If the system changes the contents of registers on which the caller depends, the caller must save them before issuing the service, and restore them after the service returns control.

## Syntax

The STORAGE macro with the OBTAIN parameter is written as follows:

---

<i>name</i>	<i>name</i> : Symbol. Begin <i>name</i> in column 1.
b	One or more blanks must precede STORAGE.
STORAGE	
b	One or more blanks must follow STORAGE.

---

OBTAIN	
,LENGTH= <i>length value</i>	<i>length value</i> : Symbol, decimal number, or register (0), (2) - (12).
,LENGTH=( <i>max amount</i> , <i>min amount</i> )	<i>max amount</i> : Symbol, decimal number, or register (0), (2) - (12).  <i>min amount</i> : Symbol, decimal number, or register (1) - (12).

	,ADDR= <i>stor addr</i>	<i>stor addr</i> : RX-type address or register (1) - (12). <b>Default:</b> ADDR=(1).
	,INADDR= <i>stor addr</i>	<i>stor addr</i> : RX-type address or register (1)-(12). <b>Note:</b> This parameter can only be specified with LOC=EXPLICIT.
	,SP= <i>subpool number</i>	<i>subpool number</i> : Symbol, decimal number 0-127, 131, 132, or register (2) - (12), (15). <b>Default:</b> SP=0.
	,BNDRY=DBLWD ,BNDRY=PAGE	<b>Default:</b> BNDRY=DBLWD
	,CONTBDY= <i>containing_bdy</i>	<i>containing_bdy</i> : Decimal number 3-31 or register (2) - (12).
	,STARTBDY= <i>starting_bdy</i>	<i>starting_bdy</i> : Decimal number 3-31 or register (2) - (12).
	,KEY= <i>key number</i>	<i>key number</i> : Decimal number 0-15 or register (2) - (12). <b>Note:</b> KEY is valid only when you also specify SP. You cannot specify both KEY and CALLRKY=YES.
	,CALLRKY=NO ,CALLRKY=YES	<b>Default:</b> CALLRKY=NO <b>Note:</b> You cannot specify both CALLRKY=YES and KEY.
	,LOC=24	
	,LOC=(24,31)	
	,LOC=(24,64)	
	,LOC=31	
	,LOC=(31,31)	
	,LOC=(31,64)	
	,LOC=RES	<b>Default:</b> LOC=RES
	,LOC=(RES,31)	
	,LOC=(RES,64)	
	,LOC=EXPLICIT	<b>Note:</b> You must specify the INADDR parameter with EXPLICIT.
	,LOC=(EXPLICIT,24)	
	,LOC=(EXPLICIT,31)	
	,LOC=(EXPLICIT,64)	
	,RTCD= <i>rtcd addr</i>	<i>rtcd addr</i> : RX-type address, register (15), or register (2) - (12). <b>Default:</b> RTCD=(15).
	,COND=YES ,COND=NO	<b>Default:</b> COND=NO
	,RELATED= <i>value</i>	<i>value</i> : Any valid macro parameter specification.

## Parameters

The parameters are explained as follows:

### OBTAIN

Requests that the system obtain virtual storage.

,LENGTH=*length value*

,LENGTH=(*max amount,min amount*)

Specifies the amount of storage the system is to obtain. *length value* specifies the length, in bytes, of the requested virtual storage. *max length* and *min length* specify the maximum and minimum amounts of storage. These numbers should be a multiple of 8; if they are not, the system uses the next higher multiple of 8.

If you specify LENGTH=(*max amount,min amount*), the system returns a value in general purpose register 0 to tell you the amount of storage it obtained.

,ADDR=*stor addr*

Specifies the location where the system returns the address of the storage it allocates.

**,INADDR=stor addr**

Specifies the desired virtual address for the storage to be obtained. When you specify INADDR, you must specify EXPLICIT on the LOC parameter.

**Notes:**

1. The address specified on INADDR must be on a doubleword boundary.
2. Make sure that the virtual storage address specified on INADDR and the central storage backing specified on the LOC=EXPLICIT parameter are a valid combination. For example, if the address specified on INADDR is for virtual storage above 16 megabytes, specify LOC=EXPLICIT or LOC=(EXPLICIT,ANY). Valid combinations include:
  - Virtual above, central any
  - Virtual any, central any
  - Virtual below, central below
  - Virtual below, central any

**,SP=subpool number**

Specifies the subpool number for the storage. Valid subpools for programs in problem state are 0 - 127, 131, and 132. See the discussion of subpool handling in *OS/390 MVS Programming: Assembler Services Guide* for information and requirements pertaining to specific subpools. If you specify a register, the subpool number must be in bits 24-31 of the register, with bits 0-23 set to zero. If you omit this parameter, the system uses subpool 0.

**,BNDRY=DBLWD****,BNDRY=PAGE**

Specifies whether the storage is to be aligned on a doubleword boundary (DBLWD) or a page boundary (PAGE). The default is BNDRY=DBLWD.

**,CONTBDY=containing\_bdy**

Specifies the boundary the obtained storage must be contained within. Specify a power of 2 that represents the containing boundary. Supported values are 3-31. For example, CONTBDY=10 means the containing boundary is  $2^{10}$ , or 1024 bytes. The containing boundary must be at least as large as the maximum requested boundary. The obtained storage will not cross an address that is a multiple of the requested boundary.

If a register is specified, the value must be in bits 24-31 of the register. Do not specify CONTBDY on a variable-length request.

CONTBDY is not valid with LOC=EXPLICIT or BNDRY=PAGE.

CONTBDY applies to all subpools.

If you omit this parameter, there is no containing boundary.

**,STARTBDY=starting\_bdy**

Specifies the boundary the obtained storage must start on. Specify a power of 2 that represents the start boundary. Supported values are 3-31. For example, STARTBDY=10 means the start boundary is  $2^{10}$ , or 1024 bytes. The obtained storage will begin on an address that is a multiple of the requested boundary.

If a register is specified, the value must be in bits 24-31 of the register. Do not specify STARTBDY on a variable-length request.

STARTBDY is not valid with LOC=EXPLICIT or BNDRY=PAGE.

STARTBDY applies to all subpools.

If you omit this parameter, the start boundary is 8 bytes (equivalent to specifying STARTBDY=3).

**,KEY=key number**

Indicates the storage key of the storage to be obtained. You may obtain storage in your storage key or in key 9. If you pass the storage key in a register, it must be in bits 24-27 in that register. KEY is valid only when SP is specified, and applies to subpools 131 and 132 only. See the discussion of subpool handling in *OS/390 MVS Programming*:

*Assembler Services Guide* for information on system-assigned defaults and authorization requirements pertaining to specific subpools.

**,CALLRKY=NO**

**,CALLRKY=YES**

Specifies how the system assigns the key for the storage to be obtained:

**CALLRKY=NO** The system assigns the value according to the specified subpool:

- For subpools 131 and 132, the system assigns the value specified on the KEY parameter (or 0, if the KEY parameter is omitted) as the storage key
- For subpools 0-127, the system assigns the value from the TCB key at the time of the first request to obtain storage. See the discussion of subpool handling in *OS/390 MVS Programming: Assembler Services Guide* for information on system-assigned defaults and authorization requirements pertaining to specific subpools.

**CALLRKY=YES** The system assigns the caller's current PSW key as the storage key. When you specify CALLRKY=YES, do not also specify KEY. Specify CALLRKY only when obtaining storage from subpools 131 and 132. For all other subpools, the system ignores the CALLRKY parameter.

The default is CALLRKY=NO.

```
|
| ,LOC=24
| ,LOC=(24,31)
| ,LOC=(24,64)
| ,LOC=31
| ,LOC=(31,31)
| ,LOC=(31,64)
| ,LOC=RES
| ,LOC=(RES,31)
| ,LOC=(RES,64)
| ,LOC=EXPLICIT
| ,LOC=(EXPLICIT,24)
| ,LOC=(EXPLICIT,31)
| ,LOC=(EXPLICIT,64)
```

Specifies the location of virtual storage and central (also called real) storage. This is especially helpful for callers with 24-bit dependencies. When LOC is specified, central storage is allocated anywhere until the storage is fixed (for example, using the PGSER macro). You can specify the location of central storage (after the storage is fixed) and virtual storage (whether or not the storage is fixed) using the following LOC parameter values:

```
|
| LOC=24 indicates that central and virtual storage are to be located below 16 megabytes.
| LOC=24 must not be used to allocate disabled reference (DREF) storage.
```

```
|
| Note: Specifying LOC=BELOW is the same as specifying LOC=24. LOC=BELOW is
| still supported, but IBM recommends using LOC=24 instead.
```

```
|
| LOC=(24,31) indicates that virtual storage is to be located below 16 megabytes and
| central storage can be located anywhere below 2 gigabytes.
```

```
|
| Note: Specifying LOC=(BELOW,ANY) is the same as specifying LOC=(24,31).
| LOC=(BELOW,ANY) is still supported, but IBM recommends using LOC=(24,31)
| instead.
```

```
|
| LOC=(24,64) indicates that virtual storage is to be located below 16 megabytes and
| central storage can be located anywhere in 64-bit storage.
```

```
|
| LOC=31 and LOC=(31,31) indicate that virtual and central storage can be located
| anywhere below 2 gigabytes.
```

**Note:** Specifying LOC=ANY or LOC=(ANY,ANY) is the same as specifying LOC =31 or LOC=(31,31). LOC=ANY and LOC=(ANY,ANY) are still supported, but IBM recommends using LOC=31 or LOC=(31,31) instead.

LOC=(31,64) indicates that virtual storage is to be located below 2 gigabytes and central storage can be located anywhere in 64-bit storage.

When you use LOC=RES to allocate storage that can reside either above or below 16 megabytes, LOC=RES indicates that the location of virtual and central storage depends on the location of the caller. If the caller resides below 16 megabytes, virtual and central storage are to be located below 16 megabytes. If the caller resides above 16 megabytes, virtual and central storage are to be located either above or below 16 megabytes.

LOC=(RES,31) indicates that the location of virtual storage depends upon the location of the caller. If the caller resides below 16 megabytes, virtual storage is to be located below 16 megabytes; if the caller resides above 16 megabytes, virtual storage can be located anywhere below 2 gigabytes. In either case, central storage can be located anywhere below 2 gigabytes.

**Note:** Specifying LOC=(RES,ANY) is the same as specifying LOC=(RES,31). LOC=(RES,ANY) is still supported, but IBM recommends using LOC=(RES,31) instead.

LOC=(RES,64) indicates that the location of virtual storage depends upon the location of the caller. If the caller resides below 16 megabytes, virtual storage is to be located below 16 megabytes; if the caller resides above 16 megabytes, virtual storage can be located anywhere in 64-bit storage. In either case, central storage can be located anywhere in 64-bit storage.

**Note:** If your program resides below 16 megabytes but runs with 31-bit addressing mode, you can specify LOC=RES (as a default or explicitly) or LOC=(RES,31) to obtain storage from a subpool supported only above 16 megabytes. Do not specify subpools supported only above 16 megabytes on requests using LOC=RES or LOC=(RES,31) if your program resides below 16 megabytes and runs with 24-bit addressing.

LOC=EXPLICIT, LOC=(EXPLICIT,24), LOC=(EXPLICIT,31), or LOC=(EXPLICIT,64) specify that the requested virtual storage is to be located at the address specified with the INADDR parameter, which is required with EXPLICIT. EXPLICIT is valid only for subpools 0-127, 131, and 132. You cannot specify the BNDRY or LENGTH=(*max amount,min amount*) parameter with EXPLICIT.

**Note:** Specifying LOC=(EXPLICIT,BELOW) is the same as specifying LOC=(EXPLICIT,24). Specifying LOC=(EXPLICIT,ANY) is the same as specifying LOC=(EXPLICIT,31). The older specifications are still supported, but IBM recommends using the newer specifications instead.

LOC=(EXPLICIT,31) indicates that virtual storage is to be located at the address specified on the INADDR parameter, and central storage can be located anywhere below 2 gigabytes.

LOC=(EXPLICIT,24) indicates that virtual storage is to be located at the address specified on the INADDR parameter, and central storage is to be located below 16 megabytes. The virtual storage address specified on the INADDR parameter must be below 16 megabytes.

LOC=EXPLICIT and LOC=(EXPLICIT,64) indicate that virtual storage is to be located at the address specified on the INADDR parameter, and central storage can be located anywhere in 64-bit storage.

When you specify EXPLICIT on a request for storage from the same virtual page as previously requested storage, you must request it in the same key, subpool, and central storage area as on the previous storage request. For example, if you request virtual storage backed with central storage below 16 megabytes, any subsequent requests for storage from that virtual page must be specified as LOC=(EXPLICIT,24).

**,RTCD=*rtcd addr***

Specifies the location where the system is to store the return code. This parameter is valid only with COND=YES. The return code is also in GPR 15.

**,COND=NO****,COND=YES**

COND=YES specifies that the active unit of work should not be abnormally terminated if there is insufficient contiguous virtual storage to satisfy the request, and instead should return to the caller with a non-zero return code. Use of COND=YES does not prevent all abnormal terminations. For example, if the request has incorrect or inconsistent parameters, the system abnormally terminates the active unit of work. If you specify COND=YES, you may also specify the RTCD parameter to define the location where the system is to store the return code.

COND=NO indicates that the request is unconditional. The system abnormally terminates the active unit of work if the STORAGE OBTAIN request cannot complete successfully. This situation occurs if the parameters passed on the request are incorrect or inconsistent, if the system encounters internal errors, or if there is not enough contiguous virtual storage to satisfy the request. COND=NO is the default.

**,RELATED=*value***

Specifies information used to self-document macro by “relating” functions or services to corresponding functions or services. The format and contents of the information specified are at the discretion of the user, and can be any valid coding values.

## ABEND Codes

STORAGE OBTAIN might issue the hexadecimal abend codes in the following list. For detailed abend code information, see *OS/390 MVS System Codes*.

178	278	378	478	778
878	978	A78	B78	D78

## Return and Reason Codes

When control returns from the STORAGE OBTAIN request and you specified a conditional request, GPR 15 (and *rtcd addr*, if you coded RTCD) contains one of the following hexadecimal return codes:

<i>Figure 64 (Page 1 of 2). Return Codes for STORAGE OBTAIN</i>	
Return Code	Meaning and Action
0	<p><b>Meaning:</b> Successful completion.</p> <p><b>Action:</b> None.</p>
4	<p><b>If you did not specify EXPLICIT on the LOC parameter:</b></p> <p><b>Meaning:</b> Environmental error. Virtual storage was not obtained because insufficient storage is available.</p> <p><b>Action:</b> Consult the system programmer to see if you have exceeded an installation-determined private storage limit.</p> <p><b>If you specified EXPLICIT on the LOC parameter:</b></p> <p><b>Meaning:</b> Program error. Virtual storage was not obtained because part of the requested storage area is outside the bounds of the user region.</p> <p><b>Action:</b> Determine why your program is mistakenly requesting storage outside the user region. If your region size is too small, consult the system programmer about increasing the region size.</p>
8	<p><b>Meaning:</b> System error. Virtual storage was not obtained because the system has insufficient central storage to back the request.</p> <p><b>Action:</b> Report the problem to the system programmer so the cause of the problem can be determined and corrected.</p>

Figure 64 (Page 2 of 2). Return Codes for STORAGE OBTAIN

Return Code	Meaning and Action
C	<p><b>Meaning:</b> System error. Virtual storage was not obtained because the system cannot page in the page table associated with the storage to be allocated.</p> <p><b>Action:</b> Report the problem to the system programmer so the cause of the problem can be determined and corrected.</p>
10	<p><b>Meaning:</b> Program error. Virtual storage was not obtained for one of the reasons listed below. This reason code applies only to STORAGE requests with LOC=EXPLICIT specified.</p> <ul style="list-style-type: none"> <li>• Part of the requested area is allocated already.</li> <li>• Virtual storage was already allocated in the same page as this request, but one of the following characteristics of the storage was different: <ul style="list-style-type: none"> <li>– The subpool</li> <li>– The key</li> <li>– Central storage backing</li> </ul> </li> </ul> <p><b>Action:</b> Determine why your program is attempting to obtain allocated storage or why your program is attempting to obtain virtual storage with different attributes from the same page of storage. Correct the coding error.</p>

## RELEASE Option of STORAGE

The STORAGE macro with the RELEASE parameter requests that the system release an area of virtual storage or an entire virtual storage subpool, previously allocated through the STORAGE or GETMAIN macro. The system abends the active task if the specified virtual storage does not start on a doubleword boundary or, for an unconditional request, if the specified area or subpool is not allocated to the task identified as the owning task.

## Input Register Information

Before issuing the STORAGE macro with the RELEASE parameter, the caller does not have to place any information into any register unless using it in register notation for a particular parameter, or using it as a base register.

## Output Register Information

When control returns to the caller, the general purpose registers (GPRs) contain:

Register	Contents
0-1	Used as work registers by the system.
2-13	Unchanged.
14	Used as a work register by the system.
15	For a conditional request, contains the return code. For an unconditional request, used as a work register by the system.

When control returns to the caller, the access registers (ARs) contain:

Register	Contents
0-1	Used as work registers by the system
2-13	Unchanged
14-15	Used as work registers by the system

Some callers depend on register contents remaining the same before and after issuing a service. If the system changes the contents of registers on which the caller depends, the caller must save them before issuing the service, and restore them after the service returns control.

## Syntax

The STORAGE macro with the RELEASE option is written as follows:

---

<i>name</i>	<i>name</i> : Symbol. Begin <i>name</i> in column 1.
b	One or more blanks must precede STORAGE.
STORAGE	
b	One or more blanks must follow STORAGE.

---

RELEASE

*,LENGTH=length value,ADDR=stor addr*  
*,LENGTH=length value,ADDR=stor addr,SP=subpool number*

*length value*: Symbol, decimal number, or register (0), (2) - (12).  
*stor addr*: RX-type address or register (1) - (12).  
*subpool number*: Symbol, decimal number 0-127, 131, 132, or register (2) - (12), (15).  
**Default**: SP=0.

*,KEY=key number*

*key number*: Decimal number 0-15 or register (2) - (12).  
**Note**: KEY is valid only when SP is specified.

*,RTCD=rtcd addr*

*rtcd addr*: RX-type address, register (15), or register (2) - (12). **Default**: RTCD=(15).

*,COND=YES*  
*,COND=NO*

**Default**: COND=NO

*,RELATED=value*

*value*: Any valid macro parameter specification.

---

## Parameters

The parameters are explained as follows:

### RELEASE

Requests that the system release virtual storage.

#### **,LENGTH=length value**

Specifies the number of bytes of storage that the system is to release. If you specify LENGTH, you must also specify ADDR. To free an entire subpool, use SP instead of LENGTH and ADDR. Do not specify a length value of 0 with an address of 0. This combination causes STORAGE RELEASE to free the subpool specified with the SP parameter, or subpool 0 if the SP parameter is omitted.

#### **,ADDR=stor addr**

Specifies the address of the storage to be released. If you specify ADDR, you must also specify LENGTH. To free an entire subpool, use SP instead of LENGTH and ADDR.

#### **,SP=subpool number**

Specifies the subpool number for the storage to be released. The valid subpool numbers are 0-127, 131, and 132. If you specify the subpool in a register, the subpool number must be in bits 24-31 of the register, with bits 0-23 set to zero. If you omit this parameter, the system uses subpool 0.

A request to release all the storage in a subpool is known as a **subpool release**. To issue a subpool release, use SP to indicate the subpool and do not specify LENGTH or ADDR. A caller in problem state can issue a subpool release for subpools 1-127, 131, and 132. A caller in problem state cannot issue a subpool release for subpool 0. See

## STORAGE Macro

the description of subpool handling in *OS/390 MVS Programming: Assembler Services Guide* for information and requirements pertaining to specific subpools.

### **,KEY=key number**

Indicates the storage key of the storage to be released. The valid storage keys are your program's storage key or key 9. If you pass the storage key in a register, it must be in bits 24-27 in that register. KEY is valid only when SP is specified and applies only to subpools 131 and 132. KEY allows you to release storage in the specified storage key. See the discussion of subpool handling in *OS/390 MVS Programming: Assembler Services Guide* for information on authorization requirements pertaining to specific subpools.

### **,RTCD=rtcd addr**

Specifies the location where the system is to store the return code. This parameter is valid only for conditional requests. The return code is also in GPR 15.

### **,COND=NO**

### **,COND=YES**

Specifies whether the request is unconditional or conditional.

COND=YES specifies that the task should not abend if the system cannot release the storage. However, the system cannot prevent some abends. The RTCD parameter specifies the location where the system is to store a return code.

COND=NO specifies that the system is to abend the active task if it cannot release the storage. COND=NO is the default.

### **,RELATED=value**

Specifies information used to self-document macro by "relating" functions or services to corresponding functions or services. The format and contents of the information specified are at the discretion of the user, and can be any valid coding values.

## ABEND Codes

STORAGE RELEASE might issue the hexadecimal abend codes in the following list. For detailed abend code information, see *OS/390 MVS System Codes*.

178	278	378	478	778
878	978	A78	B78	D78

## Return and Reason Codes

When the STORAGE macro returns control to your program and you specified a conditional request, GPR 15 (and *rtcd addr*, if you coded RTCD) contains one of the following hexadecimal return codes:

Return Code	Meaning and Action
0	<b>Meaning:</b> Successful completion. <b>Action:</b> None.
4	<b>Meaning:</b> Program error. Not all requested virtual storage was freed. <b>Action:</b> Check your program for the following kinds of errors: <ul style="list-style-type: none"><li>• The address of the storage area to be freed is not correct.</li><li>• The subpool you have specified does not match the subpool of the storage to be freed.</li><li>• The key you have specified does not match the key of the storage to be freed.</li></ul>

Figure 65 (Page 2 of 2). Return Codes for the STORAGE RELEASE

Return Code	Meaning and Action
8	<p><b>Meaning:</b> Program error. No virtual storage was freed because part of the storage area to be freed is fixed.</p> <p><b>Action:</b> Check for the following kinds of errors:</p> <ul style="list-style-type: none"> <li>You passed an incorrect storage area address to the STORAGE macro.</li> <li>You attempted to free storage that is fixed.</li> </ul>

## Examples of the OBTAIN and RELEASE Options

### Example 1

Request that the system obtain 1000 bytes of virtual storage from subpool 127 and return its address in register 3. If the request fails, the system is to abnormally end the caller.

```
LA      2,1000
STORAGE OBTAIN,LENGTH=(2),ADDR=(3),SP=127,LOC=ANY,COND=NO
```

```
.
* Release 1000 bytes from subpool 127 and abnormally end the
* caller if the request fails. Assume that the length of the storage
* is still in register 2 and the address of the storage is still in
* register 3.
```

```
.
STORAGE RELEASE,LENGTH=(2),ADDR=(3),SP=127,COND=NO
.
```

### Example 2

Request that the system obtain 4096 bytes from subpool 101 and return the address at the location defined by the RX-type address STRGA. If the request fails, the system is to save a return code at MY\_RC.

```
STORAGE OBTAIN,LENGTH=ONE_PAGE,ADDR=STRGA,SP=MY_SUBPOOL,      X
LOC=ANY,COND=YES,RTCD=MY_RC
```

```
.
* Release 4096 bytes from subpool 101.
```

```
.
STORAGE RELEASE,LENGTH=ONE_PAGE,ADDR=STRGA,SP=MY_SUBPOOL,      X
COND=YES,RTCD=MY_RC
```

```
.
MY_RC    DS F
STRGA    DS F
ONE_PAGE EQU 4096
MY_SUBPOOL EQU 101
```

### Example 3

Request that the system obtain 4096 bytes from subpool 101. If that much is not available, settle for a minimum of 1024 bytes. The system is to return the address of the storage at the RX-type address STRGA. If the request fails, the system is to store a return code at MY\_RC.

## STORAGE Macro

```
        STORAGE OBTAIN,LENGTH=(ONE_PAGE,ONE_K),ADDR=STRGA,      X
        SP=MY_SUBPOOL,LOC=ANY,COND=YES,RTCD=MY_RC
ST      0,STRG_LEN
        .
* Release the storage in subpool 101. The address of the
* storage is at the RX-type address 'STRGA'. Note that
* LENGTH=STRG_LEN is not valid.
        .
        L      3,STRG_LEN
        STORAGE RELEASE,LENGTH=(3),ADDR=STRGA,SP=MY_SUBPOOL,    X
        COND=YES,RTCD=MY_RC
        .
MY_RC   DS F
STRG_LEN DS F
STRGA   DS F
ONE_PAGE EQU 4096
ONE_K   EQU 1024
MY_SUBPOOL EQU 101
```

### Example 4

Code the instructions to set up an 18-word save area, such as one that a program in AR address space control (ASC) mode would obtain to call a program in primary mode. The program issuing the STORAGE macro is in 31-bit addressing mode, and the code is reentrant.

```
PGM    CSECT
PGM    AMODE 31
PGM    RMODE ANY
        BAKR 14,0          SAVE CALLER'S ARS, GPRS AND RETURN
*                               ADDRESS ON LINKAGE STACK
        SAC 512            SWITCH TO AR ASC MODE
        LAE 12,0(15,0)    SET UP PROGRAM BASE REGISTER AND AR
        USING PGM,12
        STORAGE OBTAIN,LENGTH=72 GET REENTRANT SAVEAREA
        LAE 13,0(1,0)    PUT SAVEAREA ADDRESS IN AR/GPR 13
        MVC 4(4,13),=C'F1SA' PUT ACRONYM INTO SAVEAREA TO
*                               INDICATE STATUS SAVED ON LINKAGE STACK
        .
* BEGIN PROGRAM CODE HERE
```

To release this save area, issue the following instructions:

```
        .
        LAE 1,0(13,0)    COPY SAVEAREA ADDRESS
        STORAGE RELEASE,ADDR=(1),LENGTH=72 FREE SAVEAREA
        .
        SLR 15,15       SET RETURN CODE OF ZERO
        PR              RETURN TO CALLER, RESTORE CALLERS STATUS
```