

Broker Resource Allocation

The EntireX Broker is a multithreaded application and communicates among multiple tasks in memory pools. If you do not need to restrict the memory expansion of EntireX Broker, we strongly recommend you enable the dynamic memory management in order to handle changing workload appropriately. See *Dynamic Memory Management* below. If dynamic memory management is disabled, non-expandable memory is allocated during startup to store all internal control blocks and the contents of messages.

This chapter covers the following topics:

- General Considerations
 - Specifying Global Resources
 - Restricting the Resources of Particular Services
 - Specifying Attributes for Privileged Services
 - Maximum Units of Work
 - Calculating Resources Automatically
 - Dynamic Memory Management
 - Dynamic Worker Management
 - Storage Report
 - Maximum TCP/IP Connections per Communicator
-

General Considerations

Resource considerations apply to both the global and service-specific levels:

- Dynamic assignment of global resources to services that need them prevents the return of a "Resource Shortage" code to an application when resources are available globally. It also enables the EntireX Broker to run with fewer total resources, although it does not guarantee the availability of a specific set of resources for a particular service.
- Flow control ensures that individual services do not influence the behavior of other services by accident, error, or simply overload. This means that you can restrict the resource consumption of particular services in order to shield the other services.

In order to satisfy both global and service-specific requirements, the EntireX Broker allows you to allocate resources for each individual service or define global resources which are then allocated dynamically to any service that needs them.

The resources in question are the number of conversations, number of servers, plus units of work and the message storage, separated in a long buffer of 4096 bytes and short buffer of 256 bytes. These resources are typically the bottleneck in a system, especially when you consider that non-conversational communication is treated as the special case of "conversations with a single message only" within the

EntireX Broker.

Global resources are defined by the parameters in the Broker section of the attribute file. The number of conversations allocated to each service is defined in the service-specific section of the attribute file. Because the conversations are shared by all servers that provide the service, a larger number of conversations should be allocated to services that are provided by more than one server. The number of conversations required is also affected by the number of clients accessing the service in parallel.

Specifying Global Resources

You can specify a set of global resources with no restrictions on which service allocates the resources:

- Specify the global attributes with the desired values.
- Do not specify any additional restrictions. That is, do not provide values for the following Broker-specific attributes:

```
LONG-BUFFER-DEFAULT
SHORT-BUFFER-DEFAULT
CONV-DEFAULT
SERVER-DEFAULT
```

- Also, do not provide values for the following server-specific attributes:

```
LONG-BUFFER-LIMIT
SERVER-LIMIT
SHORT-BUFFER-LIMIT
CONV-LIMIT
```

Example

The following example defines global resources. If no additional definitions are specified, resources are allocated and assigned to any server that needs them.

```
NUM-CONVERSATION=1000
NUM-LONG-BUFFER=200
NUM-SHORT-BUFFER=2000
NUM-SERVER=100
```

Restricting the Resources of Particular Services

You can restrict resource allocation for particular services in advance:

- Use `CONV-LIMIT` to limit the resource consumption for a specific service.
- Use `CONV-DEFAULT` to provide a default limit for services for which `CONV-LIMIT` is not defined.

Example

In the following example, attributes are used to restrict resource allocation:

```
DEFAULTS=BROKER
NUM-CONVERSATION=1000
CONV-DEFAULT=200
```

```
DEFAULTS=SERVICE
CLASS=A, SERVER=A, SERVICE=A, CONV-LIMIT=100
CLASS=B, SERVER=B, SERVICE=B, CONV-LIMIT=UNLIM
CLASS=C, SERVER=C, SERVICE=C
```

- Memory for a total of 1000 conversions is allocated (NUM-CONVERSATION=1000).
- Service A (CLASS A,SERVER A,SERVICE A) is limited to 100 conversation control blocks used simultaneously (CONV-LIMIT=100). The application that wants to start more conversations than specified by the limit policy will receive a "Resource shortage" return code. This return code should result in a retry of the desired operation a little later, when the resource situation may have changed.
- Service B (CLASS B,SERVER B,SERVICE B) is allowed to try to allocate as many resources as necessary, provided the resources are available and not occupied by other services. The number of conversations that may be used by this service is unlimited (CONV-LIMIT=UNLIM).
- Service C (CLASS C,SERVER C,SERVICE C) has no explicit value for the CONV-LIMIT attribute. The number of conversation control blocks that it is allowed to use is therefore limited to the default value which is defined by the CONV-DEFAULT Broker attribute.

The same scheme applies to the allocation of message buffers and servers:

- In the following example, long message buffers are allocated using the keywords NUM-LONG-BUFFER, LONG-BUFFER-DEFAULT and LONG-BUFFER-LIMIT:

```
DEFAULTS=BROKER
NUM-LONG-BUFFER=2000
LONG-BUFFER-DEFAULT=250
```

```
DEFAULTS=SERVICE
CLASS=A, SERVER=A, SERVICE=A, LONG-BUFFER-LIMIT=100
CLASS=B, SERVER=B, SERVICE=B, LONG-BUFFER-LIMIT=UNLIM
CLASS=C, SERVER=C, SERVICE=C
```

- In the following example, short message buffers are allocated using the keywords NUM-SHORT-BUFFER, SHORT-BUFFER-DEFAULT and SHORT-BUFFER-LIMIT:

```
DEFAULTS=BROKER
NUM-SHORT-BUFFER=2000
SHORT-BUFFER-DEFAULT=250
```

```
DEFAULTS=SERVICE
CLASS=A, SERVER=A, SERVICE=A, SHORT-BUFFER-LIMIT=100
CLASS=B, SERVER=B, SERVICE=B, SHORT-BUFFER-LIMIT=UNLIM
CLASS=C, SERVER=C, SERVICE=C
```

- In the following example, servers are allocated using the keywords NUM-SERVER, SERVER-DEFAULT and SERVER-LIMIT:

```

DEFAULTS=BROKER
NUM-SERVER=2000
SERVER-DEFAULT=250

```

```

DEFAULTS=SERVICE
CLASS=A, SERVER=A, SERVICE=A, SERVER-LIMIT=100
CLASS=B, SERVER=B, SERVICE=B, SERVER-LIMIT=UNLIM
CLASS=C, SERVER=C, SERVICE=C

```

Specifying Attributes for Privileged Services

If privileged services (services with access to unlimited resources) exist, specify UNLIMITED for the attributes CONV-LIMIT, SERVER-LIMIT, LONG-BUFFER-LIMIT and SHORT-BUFFER-LIMIT in the service-specific section of the attribute file.

For example:

```

DEFAULTS=SERVICE
CONV-LIMIT=UNLIM
LONG-BUFFER-LIMIT=UNLIM
SHORT-BUFFER-LIMIT=UNLIM
SERVER-LIMIT=UNLIM

```

To ensure a resource reservoir for peak load of privileged services, define more resources than would normally be expected by specifying larger numbers for the Broker attributes that control global resources:

```

NUM-SERVER
NUM-CONVERSATION
CONV-DEFAULT
LONG-BUFFER-DEFAULT
SHORT-BUFFER-DEFAULT
SERVER-DEFAULT

```

Maximum Units of Work

The maximum number of units of work (UOWs) that can be active concurrently is specified in the Broker attribute file. The MAX-UOWS attribute can be specified for the Broker globally as well as for individual services. It cannot be calculated automatically. If a service is intended to process UOWs, a MAX-UOWS value must be specified.

If message processing only is to be done, specify MAX-UOWS=0 (zero). The Broker (or the service) will not accept units of work, i.e., it will process only messages that are not part of a UOW. Zero is used as the default value for MAX-UOWS in order to prevent the sending of UOWs to services that are not intended to process them.

Calculating Resources Automatically

To ensure that each service runs without impacting other services, allow the EntireX Broker to calculate resource requirements automatically:

- Ensure that the attributes that define the default total for the Broker and the limit for each service are not set to UNLIM.

- Specify `AUTO` for the Broker attribute that defines the total number of the resource.
- Specify a suitable value for the Broker attribute that defines the default number of the resource.

The total number required will be calculated from the number defined for each service. The resources that can be calculated this way are Number of Conversations, Number of Servers, Long Message Buffers and Short Message Buffers.

Avoid altering the service-specific definitions at runtime. Doing so could corrupt the conversation consistency. Applications might receive a message such as "NUM-CONVERSATIONS reached" although the addressed service does not serve as many conversations as defined. The same applies to the attributes that define the long and short buffer resources.

Automatic resource calculation has the additional advantage of limiting the amount of memory used to run the EntireX Broker. Over time, you should be able to determine which services need more resources by noting the occurrence of the return code "resource shortage, please retry". You can then increase the resources for these services. To avoid disruption to the user, you could instead allocate a relatively large set of resources initially and then decrease the values using information gained from the Administration Monitor application.

Number of Conversations

To calculate the total number of conversations automatically, ensure that the `CONV-DEFAULT` Broker attribute and the `CONV-LIMIT` service-specific attribute are not set to `UNLIM` anywhere in the attribute file. Specify `NUM-CONVERSATION=AUTO` and an appropriate value for the `CONV-DEFAULT` Broker attribute. The total number of conversations will be calculated using the value specified for each service.

For example:

```
DEFAULTS=BROKER
NUM-CONVERSATION=AUTO
CONV-DEFAULT=200
```

```
DEFAULTS=SERVICE
CLASS=A, SERVER=A, SERVICE=A
CLASS=B, SERVER=B, SERVICE=B, CONV-LIMIT=100
CLASS=C, SERVER=C, SERVICE=C
```

- Service A and Service C both need 200 conversations (the default value). Service B needs 100 conversations (`CONV-LIMIT=100`).
- Because `NUM-CONVERSATIONS` is defined as `AUTO`, the broker calculates a total of 500 conversations ($200 + 200 + 100$).
- `NUM-CONVERSATIONS=AUTO` allows the number of conversations to be flexible without requiring additional specifications. It also ensures that the broker is started with enough resources to meet all the demands of the individual services.
- `AUTO` and `UNLIM` are mutually exclusive. If `CONV-DEFAULT` or a single `CONV-LIMIT` is defined as `UNLIM`, the EntireX Broker cannot determine the number of conversations to use in the calculation, and the EntireX Broker cannot be started.

Number of Servers

To calculate the number of servers automatically, ensure that the `SERVER-DEFAULT` Broker attribute and the `SERVER-LIMIT` service-specific attribute are not set to `UNLIM` anywhere in the attribute file. Specify `NUM-SERVER=AUTO` and an appropriate value for the `SERVER-DEFAULT` Broker attribute. The total number of server buffers will be calculated using the value specified for each service.

For example:

```
DEFAULTS=BROKER
NUM-SERVER=AUTO
SERVER-DEFAULT=250
```

```
DEFAULTS=SERVICE
CLASS=A, SERVER=A, SERVICE=A, SERVER-LIMIT=100
CLASS=B, SERVER=B, SERVICE=B
CLASS=C, SERVER=C, SERVICE=C
```

Long Message Buffers

To calculate the number of long message buffers automatically, ensure that the `LONG-BUFFER-DEFAULT` Broker attribute and the `LONG-BUFFER-LIMIT` service-specific attribute are not set to `UNLIM` anywhere in the attribute file. Specify `NUM-LONG-BUFFER=AUTO` and an appropriate value for the `LONG-BUFFER-DEFAULT` Broker attribute. The total number of long message buffers will be calculated using the value specified for each service.

For example:

```
DEFAULTS=BROKER
NUM-LONG-BUFFER=AUTO
LONG-BUFFER-DEFAULT=250
```

```
DEFAULTS=SERVICE
CLASS=A, SERVER=A, SERVICE=A, LONG-BUFFER-LIMIT=100
CLASS=B, SERVER=B, SERVICE=B
CLASS=C, SERVER=C, SERVICE=C
```

Short Message Buffers

To calculate the number of short message buffers automatically, ensure that the `SHORT-BUFFER-DEFAULT` Broker attribute and the `SHORT-BUFFER-LIMIT` service-specific attribute are not set to `UNLIM` anywhere in the attribute file. Specify `NUM-SHORT-BUFFER=AUTO` and an appropriate value for the `SHORT-BUFFER-DEFAULT` Broker attribute. The total number of short message buffers will be calculated using the value specified for each service.

For example:

```
DEFAULTS=BROKER
NUM-SHORT-BUFFER=AUTO
SHORT-BUFFER-DEFAULT=250
```

```
DEFAULTS=SERVICE
CLASS=A, SERVER=A, SERVICE=A
CLASS=B, SERVER=B, SERVICE=B, SHORT-BUFFER-LIMIT=100
CLASS=C, SERVER=C, SERVICE=C
```

Dynamic Memory Management

Dynamic memory management is a feature to handle changing Broker workload without any restart of the Broker task. It increases the availability of the Broker by using various memory pools for various Broker resources and by being able to use a variable number of pools for the resources.

If more memory is needed than currently available, another memory pool is allocated for the specific type of resource. If a particular memory pool is no longer used, it will be deallocated.

The following Broker attributes can be omitted if `DYNAMIC-MEMORY-MANAGEMENT=YES` has been defined:

- `NUM-CLIENT`
- `NUM-CMDLOG-FILTER`
- `NUM-COMBUF`
- `NUM-CONV[ERSATION]`
- `NUM-LONG[-BUFFER]`
- `NUM-PUBLICATION`
- `NUM-PUBLISHER`
- `NUM-SERVER`
- `NUM-SERVICE`
- `NUM-SERVICE-EXTENSION`
- `NUM-SHORT[-BUFFER]`
- `NUM-SUBSCRIBER`
- `NUM-SUBSCRIBER-TOTAL`
- `NUM-TOPIC`
- `NUM-TOPIC-EXTENSION`
- `NUM-TOPIC-TOTAL`
- `NUM-UOW | MAX-UOWS | MUOW`
- `NUM-WQE`

If you want statistics on allocation and deallocation operations in Broker, you can configure Broker to create a storage report with the attribute `STORAGE-REPORT`. See *Storage Report* below.

Note:

To ensure a stable environment, some pools of Broker are not deallocated automatically. The first pools of type `COMMUNICATION`, `CONVERSATION`, `CONNECTION`, `HEAP`, `PARTICIPANT`, `PARTICIPANT EXTENSION`, `SERVICE ATTRIBUTES`, `SERVICE`, `SERVICE EXTENSION`, `TIMEOUT QUEUE`, `TRANSLATION`, `WORK QUEUE` are excluded from the automatic deallocation even when they have not been used for quite some time. Large pools cannot be reallocated under some circumstances if the level of fragmentation in the address space has been increased in the meantime.

Dynamic Worker Management

Dynamic worker management is a feature to handle the fluctuating broker workload without restarting the Broker task. It adjusts the number of running worker tasks according to current workload. The initial portion of worker tasks started at Broker startup is still determined by `NUM-WORKER`.

If more workers are needed than currently available, another worker task is started. If a worker task is no longer needed, it will be stopped.

The following Broker attributes are used for the configuration if `DYNAMIC-WORKER-MANAGEMENT=YES` has been defined:

- `WORKER-MAX`

- WORKER-MIN
- WORKER-NONACT
- WORKER-QUEUE-DEPTH
- WORKER-START-DELAY

The following two attributes are very performance-sensitive:

- Attribute WORKER-QUEUE-DEPTH defines the number of unassigned user requests in the input queue before a new worker task is started.
- Attribute WORKER-START-DELAY defines the time between the last worker task startup and the next check for another possible worker task startup. It is needed to consider the time for activating a worker task.

Both attributes depend on the environment, in particular the underlying operating system and the hardware. The goal is to achieve high-performance user request processing without starting too many worker tasks.

A good starting point to achieve high performance is not to change the attributes and to observe the performance of the application programs after activating the dynamic worker management.

If broker attribute DYNAMIC-WORKER-MANAGEMENT=YES is set, operator commands are available under z/OS to deactivate and subsequently reactivate dynamic worker management.

The following section illustrates the two different modes of dynamic worker management:

- **Scenario 1**

```
DYNAMIC-WORKER-MANAGEMENT=YES
NUM-WORKER = 5
WORKER-MIN = 1
WORKER-MAX = 32
```

Broker is started with 5 worker tasks and then dynamically varies the number of worker tasks within the range from WORKER-MIN=1 to WORKER-MAX=32 due to DYNAMIC-WORKER-MANAGEMENT=YES.

- **Scenario 2**

```
DYNAMIC-WORKER-MANAGEMENT=NO
NUM-WORKER = 5
WORKER-MIN = 1
WORKER-MAX = 32
```

Broker is started with 5 worker tasks. The WORKER-MIN/MAX attributes are ignored due to DYNAMIC-WORKER-MANAGEMENT=NO.

Storage Report

You can create an optional report file that provides details about all activities to allocate or to deallocate memory pools. This section details how to create the report and provides a sample report.

- Creating a Storage Report
- Platform-specific Rules
- Sample Storage Report

See also Broker-specific attribute `STORAGE-REPORT`.

Creating a Storage Report

Use Broker's global attribute `STORAGE-REPORT` with the value `YES`. If attribute value `YES` is supplied, all memory pool operations will be reported if the output mechanism is available. If the value `NO` is specified, no report will be created.

Platform-specific Rules

z/OS

`DDNAME ETBSREP` assigns the report file. Format `RECFM=FB`, `LRECL=121` is used.

UNIX and Windows

Broker creates a file with the name `STORAGE.REPORT` in the current working directory. If the environment variable `ETB_STORAGE_REPORT` is supplied, the file name specified in the environment variable will be used. If Broker receives the command-line argument `-r`, the token following argument `-r` will be used as the file name.

BS2000

`LINK-NAME ETBSREP` assigns the report file. Format `REC-FORM=V`, `REC-SIZE=0`, `FILE-TYPE ISAM` is used by default.

z/VSE

Logical unit `SYS015` and logical file name `ETBSREP` are used. Format `RECORD-FORMAT=FB`, `RECORD-LENGTH=121` is used.

Sample Storage Report

The following is an excerpt from a sample `STORAGE` report.

```
EntireX 8.1.0.0      STORAGE Report      2009-06-26 12:28:58      Page      1

Identifier          Address          Size            Total          Date          Time          Action
KERNEL POOL        0x25E48010      407184 bytes    407184 bytes    2009-06-26    12:28:58.768  Allocated
HEAP POOL          0x25EB4010      1050692 bytes   1457876 bytes   2009-06-26    12:28:58.769  Allocated
COMMUNICATION POOL 0x25FB5010      16781380 bytes  18239256 bytes  2009-06-26    12:28:58.769  Allocated
ACCOUNTING POOL    0x26FB7010      762052 bytes    19001308 bytes  2009-06-26    12:28:58.769  Allocated
BROKER POOL        0x27072010      61540 bytes     19062848 bytes  2009-06-26    12:28:58.775  Allocated
CONVERSATION POOL 0x27082010      368964 bytes    19431812 bytes  2009-06-26    12:28:58.775  Allocated
CONNECTION POOL    0x270DD010      233668 bytes    19665480 bytes  2009-06-26    12:28:58.779  Allocated
```

Broker Resource Allocation

Maximum TCP/IP Connections per Communicator

| | | | | | |
|----------------------------|------------|----------------|----------------|-------------------------|-------------|
| LONG MESSAGES POOL | 0x27117010 | 4395204 bytes | 24060684 bytes | 2009-06-26 12:28:58.782 | Allocated |
| SHORT MESSAGES POOL | 0x27549010 | 3703876 bytes | 27764560 bytes | 2009-06-26 12:28:58.806 | Allocated |
| PARTICIPANT POOL | 0x278D2010 | 134244 bytes | 27898804 bytes | 2009-06-26 12:28:58.827 | Allocated |
| PARTICIPANT EXTENSION POOL | 0x278F3010 | 36996 bytes | 27935800 bytes | 2009-06-26 12:28:58.829 | Allocated |
| PROXY QUEUE POOL | 0x278FD010 | 26724 bytes | 27962524 bytes | 2009-06-26 12:28:58.829 | Allocated |
| SERVICE ATTRIBUTES POOL | 0x27904010 | 131668 bytes | 28094192 bytes | 2009-06-26 12:28:58.829 | Allocated |
| SERVICE POOL | 0x27925010 | 54372 bytes | 28148564 bytes | 2009-06-26 12:28:58.830 | Allocated |
| SERVICE EXTENSION POOL | 0x27933010 | 32900 bytes | 28181464 bytes | 2009-06-26 12:28:58.831 | Allocated |
| TIMEOUT QUEUE POOL | 0x2793C010 | 87268 bytes | 28268732 bytes | 2009-06-26 12:28:58.831 | Allocated |
| TRANSLATION POOL | 0x27952010 | 179300 bytes | 28448032 bytes | 2009-06-26 12:28:58.832 | Allocated |
| UNIT OF WORK POOL | 0x2797E010 | 176324 bytes | 28624356 bytes | 2009-06-26 12:28:58.834 | Allocated |
| WORK QUEUE POOL | 0x279AA010 | 391268 bytes | 29015624 bytes | 2009-06-26 12:28:58.835 | Allocated |
| BLACKLIST POOL | 0x27A0A010 | 42084 bytes | 29057708 bytes | 2009-06-26 12:28:58.838 | Allocated |
| SUBSCRIPTION POOL | 0x27A15010 | 344148 bytes | 29401856 bytes | 2009-06-26 12:28:58.839 | Allocated |
| TOPIC ATTRIBUTES POOL | 0x27A6A010 | 129620 bytes | 29531476 bytes | 2009-06-26 12:28:58.841 | Allocated |
| TOPIC POOL | 0x26FB6068 | 2952 bytes | 29534428 bytes | 2009-06-26 12:28:58.842 | Allocated |
| TOPIC EXTENSION POOL | 0x27A8A010 | 30852 bytes | 29565280 bytes | 2009-06-26 12:28:58.842 | Allocated |
| PSTORE SUBSCRIBER POOL | 0x27A92010 | 33892 bytes | 29599172 bytes | 2009-06-26 12:28:58.843 | Allocated |
| PSTORE TOPIC POOL | 0x27A9B010 | 19540 bytes | 29618712 bytes | 2009-06-26 12:28:58.843 | Allocated |
| COMMUNICATION POOL | 0x25FB5010 | 16781380 bytes | 12837332 bytes | 2009-06-26 12:30:58.514 | Deallocated |
| ACCOUNTING POOL | 0x26FB7010 | 762052 bytes | 12075280 bytes | 2009-06-26 12:30:58.515 | Deallocated |
| BROKER POOL | 0x27072010 | 61540 bytes | 12013740 bytes | 2009-06-26 12:30:58.516 | Deallocated |
| CONVERSATION POOL | 0x27082010 | 368964 bytes | 11644776 bytes | 2009-06-26 12:30:58.518 | Deallocated |
| CONNECTION POOL | 0x270DD010 | 233668 bytes | 11411108 bytes | 2009-06-26 12:30:58.519 | Deallocated |
| LONG MESSAGES POOL | 0x27117010 | 4395204 bytes | 7015904 bytes | 2009-06-26 12:30:58.520 | Deallocated |
| SHORT MESSAGES POOL | 0x27549010 | 3703876 bytes | 3312028 bytes | 2009-06-26 12:30:58.526 | Deallocated |
| PROXY QUEUE POOL | 0x278FD010 | 26724 bytes | 3285304 bytes | 2009-06-26 12:30:58.530 | Deallocated |
| SUBSCRIPTION POOL | 0x27A15010 | 344148 bytes | 2941156 bytes | 2009-06-26 12:30:58.530 | Deallocated |
| TOPIC ATTRIBUTES POOL | 0x27A6A010 | 129620 bytes | 2811536 bytes | 2009-06-26 12:30:58.531 | Deallocated |
| TOPIC POOL | 0x26FB6068 | 2952 bytes | 2808584 bytes | 2009-06-26 12:30:58.531 | Deallocated |
| TOPIC EXTENSION POOL | 0x27A8A010 | 30852 bytes | 2777732 bytes | 2009-06-26 12:30:58.531 | Deallocated |
| TIMEOUT QUEUE POOL | 0x2793C010 | 87268 bytes | 2690464 bytes | 2009-06-26 12:30:58.532 | Deallocated |
| UNIT OF WORK POOL | 0x2797E010 | 176324 bytes | 2514140 bytes | 2009-06-26 12:30:58.533 | Deallocated |
| WORK QUEUE POOL | 0x279AA010 | 391268 bytes | 2122872 bytes | 2009-06-26 12:30:58.533 | Deallocated |
| BLACKLIST POOL | 0x27A0A010 | 42084 bytes | 2080788 bytes | 2009-06-26 12:30:58.534 | Deallocated |
| PSTORE SUBSCRIBER POOL | 0x27A92010 | 33892 bytes | 2046896 bytes | 2009-06-26 12:30:58.534 | Deallocated |
| PSTORE TOPIC POOL | 0x27A9B010 | 19540 bytes | 2027356 bytes | 2009-06-26 12:30:58.534 | Deallocated |
| PARTICIPANT POOL | 0x278D2010 | 134244 bytes | 1893112 bytes | 2009-06-26 12:49:25.817 | Deallocated |
| PARTICIPANT EXTENSION POOL | 0x278F3010 | 36996 bytes | 1856116 bytes | 2009-06-26 12:49:25.818 | Deallocated |
| SERVICE ATTRIBUTES POOL | 0x27904010 | 131668 bytes | 1724448 bytes | 2009-06-26 12:49:25.818 | Deallocated |
| SERVICE POOL | 0x27925010 | 54372 bytes | 1670076 bytes | 2009-06-26 12:49:25.818 | Deallocated |
| SERVICE EXTENSION POOL | 0x27933010 | 32900 bytes | 1637176 bytes | 2009-06-26 12:49:25.819 | Deallocated |
| TRANSLATION POOL | 0x27952010 | 179300 bytes | 1457876 bytes | 2009-06-26 12:49:25.819 | Deallocated |
| HEAP POOL | 0x25EB4010 | 1050692 bytes | 407184 bytes | 2009-06-26 12:49:25.820 | Deallocated |
| KERNEL POOL | 0x25E48010 | 407184 bytes | 0 bytes | 2009-06-26 12:49:25.820 | Deallocated |

| Header | Description |
|------------|---|
| Identifier | Name of the memory pool. |
| Address | Start address of the memory pool. |
| Size | Size of the memory pool. |
| Total | Total size of all obtained memory pools. |
| Date, Time | Date and time of the action. |
| Action | The action of Broker. The following actions are currently supported: Allocated: memory pool is allocated . Deallocated: memory pool is deallocated. |

Maximum TCP/IP Connections per Communicator

This table shows the maximum number of TCP/IP connections per communicator:

| Platform | Maximum Number of TCP/IP Connections per Communicator |
|------------|---|
| AIX | 2,048 |
| BS2000/OSD | 2,048 |
| HP-UX | 2,048 |
| Linux | 4,096 |
| Solaris | 65,356 |
| Windows | 4,096 |
| z/OS | 16,384 |
| z/VSE | 2,048 |

With the Broker-specific attribute `POLL`, these restrictions can be lifted under z/OS, UNIX and z/VSE. See `POLL`.

See also `MAX-CONNECTIONS` under `TCP-OBJECT` (*Struct* `INFO_TCP`) under *Information Reply Structures* in the Broker CIS documentation.

Note for UNIX

Under UNIX, you can use the following command to display the maximum number of open files in the operating system shell.

```
ulimit -n
```

This value should be greater than the expected number of TCP/IP connections.