- GLOBAL — Queues created with the GLOBAL option set are visible and accessible from any node in the system. When a message is sent to a queue in another node, the message is physically copied to that other node. In non-shared memory systems, it is not guaranteed that a message has arrived in the destination node before the operation returns a successful completion status.

- FIFO — With this option set, the tasks waiting for messages from the queue will be queued first in first out. The tasks are by default queued in order of task priority.

7.1. QUEUE_CREATE

Create a message queue.

Synopsis

```
   queue_create( name, max_buff, length, options, qid )
```

Input Parameters

name	: string	user defined queue name
max_buff	: integer	maximum number of buffers allowed in queue
length	: integer	length of message buffers in bytes
options	: bit_field	queue create options

Output Parameters

qid	: queue_id	kernel defined queue identifier

Literal Values

options	+ GLOBAL	the new queue will be visible throughout the system
	+ FIFO	tasks waiting on a message will be queued first in first out

Completion Status

OK	queue_create operation successful
ILLEGAL_USE	operation not callable from XSR or ISR
INVALID_PARAMETER	a parameter refers to an illegal address
INVALID_LENGTH	buffer length not supported
INVALID_OPTIONS	invalid options value
TOO_MANY_QUEUES	too many queues on node
NO_MORE_MEMORY	not enough memory to allocate message buffer(s)

Description

This operation creates a new queue in the kernel data structure. The given number of buffers of the given length are allocated by the kernel. If the kernel cannot find sufficient memory it returns the NO_MORE_MEMORY completion status.

The maximum possible length of messages is implementation dependent, but an ORKID compliant kernel is required to support message lengths of up to 32 bytes.

7.2. QUEUE_DELETE

Delete an existing queue.

Synopsis

```
   queue_delete( qid )
```

Input Parameters

qid	: queue_id	kernel defined queue identifier

Output Parameters

<none>

Completion Status

OK	queue_delete operation successful
ILLEGAL_USE	operation not callable from ISR
INVALID_PARAMETER	a parameter refers to an illegal address
INVALID_ID	queue does not exist
OBJECT_DELETED	queue specified has been deleted
NODE_NOT_REACHABLE	node on which semaphore resides is not reachable

Description

This option deletes the given queue from the kernel data structure. If any tasks were waiting for a message from the queue, they are unblocked and returned the QUEUE_DELETED completion status. If there were any messages in the queue, they are lost and the buffers deallocated.

7.3. QUEUE_IDENT

Obtain the identifier of a queue on a given node with a given name.

Synopsis

```
queue_ident( name, nid, qid )
```

Input Parameters

name	: string	user defined queue name
nid	: node_id	node identifier

Output Parameters

qid	: queue_id	kernel defined queue identifier

Literal Values

nid	= LOCAL_NODE	the node containing the calling task
	= OTHER_NODES	all nodes in the system except the local node.

Completion Status

OK	queue_ident operation successful
ILLEGAL_USE	operation not callable from XSR or ISR
INVALID_PARAMETER	a parameter refers to an illegal address
INVALID_NODE	node does not exist
NAME_NOT_FOUND	name does not exist on node
NODE_NOT_REACHABLE	node on which semaphore resides is not reachable

Description

This operation searches the kernel data structure in the node(s) specified for a queue with the given name, and returns its identifier if found. If OTHER_NODES is specified, the node search order is implementation dependent. If there is more than one queue with the same name in the node(s) specified, then the qid of the first one found is returned.

7.4. QUEUE_SEND

Send a message to a given queue.

Synopsis

```
queue_send( qid, message, length )
```

Input Parameters

qid	: queue_id	kernel defined queue identifier
message	: address	message starting address
length	: integer	length of message in bytes

Output Parameters

<none>

Completion Status

OK	queue_send operation successful
INVALID_PARAMETER	a parameter refers to an illegal address
INVALID_ID	queue does not exist
OBJECT_DELETED	queue specified has been deleted
INVALID_LENGTH	message length greater than queue's buffer length
QUEUE_FULL	no more buffers available
NODE_NOT_REACHABLE	node on which semaphore resides is not reachable

Description

This operations sends a message to a queue. If the queue's wait queue contains a number of tasks waiting on messages, then the message is delivered to the task at the head of the wait queue. This task is then removed from the wait queue, unblocked and will be returned a successful completion status along with the message. Otherwise the message is put on the queue.

If the maximum queue length has been reached, then the QUEUE_FULL completion status is returned.