

queue-handler

COLLABORATORS

	TITLE : queue-handler		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 23, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	queue-handler	1
1.1	queue-handler.doc	1
1.2	queue-handler/queue_handler	1

Chapter 1

queue-handler

1.1 queue-handler.doc

```
queue_handler()
```

1.2 queue-handler/queue_handler

NAME

queue_handler -- an 'infinite' pipe (FIFO) handler. (V37)

SYNOPSIS

```
Open("PIPE:[channel_name][[/[buf_size]][/max_buffers]]");
```

FUNCTION

Queue-handler takes data from the output channel, buffers it up, allocating more memory as needed, and sends it to the input channel, as requested by the applications. Queue-handler never locks the output channel unless max_buffers was specified. Buffer size can be specified only on the first reference to a particular channel.

Queue-handler works by invoking a separate process every time an open packet is received, and uses a FIFO buffer pool approach to track the actual data.

INPUTS

channel_name - unique channel name to specify. Must begin with a non-numeric character. The channel_name is optional.

buf_size - Size in bytes of the buffers to allocate (default is 4096). The buf_size is optional.

max_buffers - maximum number of buffers allowed. Will suspend the output channel if exceeded. A max_buffers value of 0 indicates there should be no fixed limit on the number of buffers allocated.

PACKETS

```

ACTION_FINDINPUT
ACTION_FINDOUTPUT
ACTION_FINDUPDATE
ACTION_READ
ACTION_WRITE
ACTION_END
ACTION_IS_FILESYSTEM

```

MOUNTLIST ENTRIES

This is the default mountlist entry used for Queue-Handler:

```

PIPE:   Handler    = L:Queue-Handler
        Priority   = 5
        StackSize  = 3000
        GlobVec    = -1
#

```

Below is an extended mountlist entry which lets you specify a default number of buffers, and their size. There are some extra fields added at the end of the mountlist entry which aren't used for anything, but the Mount command requires that these fields be present. In this mountlist, the "Buffers" field indicates the default maximum number of buffers, and "SectorSize" indicates the size of these buffers.

```

PIPE:   FileSystem = L:Queue-Handler
        Priority   = 5
        StackSize  = 3000
        GlobVec    = -1
        SectorSize = 1024
        Buffers    = 2

        /* these are unused, but required by Mount */
        Surfaces   = 1
        SectorsPerTrack = 1
        LowCyl      = 0
        HighCyl     = 1
        Device      = ""
        Unit        = 0
#

```

EXAMPLE

```

From process 1:
list >pipe: work: all

```

```

From process 2:
type pipe:

```

```

To gather the results of several C compilations:
lc >pipe:ll foo
lc >pipe:ll bar
lc >pipe:ll road
lc >pipe:ll kill
type pipe:ll

```

```

To use channel names:
list >pipe:crazy
copy #?.c to >pipe:all_c/32000 ;Specifies a channel 'all_c' and

```

```
;a buffer size of 32000  
;bytes
```

To set a limit on the number of buffers to 5:

```
dir >pipe://5          ; create a channel without a channel-name, and  
                        ; only allow 5 buffer entries.
```

BUGS

Full checking is not done to ensure that the application can't read from a write channel or vice-versa. The results of this operation shall be undefined. Don't do it.