in

COLLABORATORS			
	TITLE : in		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		June 25, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

in

1.1 Autodocs

TABLE OF CONTENTS


```
    EPC_ general
        All commands are used in the same way. Ext_Proc_Msg is a standard
        exec message which is sent from external processes to SamEd and then
        returned. epm_Msg is an exec Message struct which should be
        initialised so the reply is sent to you. epm_Command should equal the
        command you wish to perform, and epm_Data is command specific.
        epm_Error is an error defined in extproc.h. epm_Error equals
        EPCERR_NOERR (or NULL) if no error occured.
        Commands with (LOCK) after the name require you to hold the LOCK to
        SamEd (see EPC_LOCK, EPC_UNLOCK).
```

```
    NEW
    Send messages using the functions EP_SendMsg(), and EP_SendIDMsg().

NAME
    EPC_VERSION

FUNCTION
    Find the current version of SamEd's message port handler. If the
    returned version number is less than the version you require then
    some commands are not supported and you may need to quit (and inform
    the user).

INPUTS
    epm_Command =   EPC_VERSION
    epm_Data    =   NULL

RESULT
    epm_Data will point to a ULONG containing the version number.
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h


NAME
    EPC_AUTOQUIT

FUNCTION
    Tell SamEd to signal you if it quits. You need to send a pointer to
    your task and a signal bit number in an Ext_Proc_ID struct. You can
    then wait on the signal bit to check if SamEd quits, and if it does
    you should also.

INPUTS
    epm_Command =   EPC_AUTOQUIT
    epm_Data    -   pointer to an Ext_Proc_ID struct:
        task    -   pointer to your task (ie. = FindTask(NULL);)
        number  -   a signal bit number (eg. = AllocSig(-1);)

RESULT
    You will be placed on SamEd's list of tasks to signal, until either
    you send an EPC_QUIT or SamEd quits.
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h

SEE ALSO
    EPC_QUIT.


NAME
    EPC_IDNUM
FUNCTION
    Return your id / button number, given a pointer to your task.
INPUTS
RESULT
BUGS
    Not tested, use the CLI argument.
SEE ALSO
```

```
NAME
    EPC_MULTI

FUNCTION
    Set multi load mode for your ext. proc. When SamEd launches an ext.
    proc. the button pressed then becomes inactive so only one copy of
    the process is running at any one time. To reactivate the button an
    EPC_QUIT needs to be send. Alternatively the button can be in MULTI
    LOAD mode, meaning a new process is loaded with each press of your
    button. When a button is put into MULTI LOAD mode it will stay that
    way until SamEd quits, it cannot be cancelled.

INPUTS
    epm_Command =   EPC_MULTI
    epm_Data    -   points to a ULONG holding your button number

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h

SEE ALSO
    EPC_QUIT.


NAME
    EPC_QUIT

FUNCTION
    Tell SamEd that you have / will shortly quit. The actual actions of
    this command are two fold. It both cancels EPC_AUTOQUIT for your
    task, and allows your button to be reused (if you haven't sent
    EPC_MULTI).

INPUTS
    epm_Command =   EPC_QUIT
    epm_Data    -   pointer to an Ext_Proc_ID struct:
        task    -   a pointer to your task, you can send this even if
                    EPC_AUTOQUIT was not allocated.
        number  -   should equal your button number.

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h, although you con ignore the error.

SEE ALSO
    EPC_AUTOQUIT, EPC_MULTI.


NAME
    EPC_TOTALNUM

FUNCTION
    Return the total number of samples available. SamEd holds the
    information for samples in an array of sam_info structs:
```

```
    struct sam_info sample[MAX_SAMPLES +1];
    sample[0]   =   copy buffer;
    sample[1 to MAX_SAMPLES] =   sound samples.

    EPC_TOTALNUM returns MAX_SAMPLES.

INPUTS
    epm_Command =   EPC_TOTALNUM
    epm_Data    =   NULL

RESULT
    epm_Data    -   points to a ULONG containing the max. number of
    samples.
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h

SEE ALSO
    EPC_GETSAMPLES, EPC_GETSAMNUM, EPC_SETSAMNUM.


NAME
    EPC_LOCK

FUNCTION
    Lock SamEd so that you can fiddle with its data. Some commands
    require you to hold the lock to SamEd. You must always lock SamEd
    before altering any data. You must remember to unlock SamEd.

INPUTS
    epm_Command =   EPC_LOCK
    epm_Data    =   NULL

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h. You MUST check this.

BUGS
    Semaphores should realy be used. They may be in future.

SEE ALSO
    EPC_UNLOCK


NAME
    EPC_UNLOCK      (LOCK)

FUNCTION
    Unlock the system previously locked. You MUST unlock the system if
    you locked it. You MUST NOT unlock the system if you did not obtain
    the lock.

INPUTS
    epm_Command =   EPC_UNLOCK
    epm_Data    =   NULL

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
```

```
    extproc.h.

SEE ALSO
    EPC_LOCK.


NAME
    EPC_GETSAMPLE   (LOCK)

FUNCTION
    Get a pointer to the current sam_info structure. You can then alter
    the data stored in it (carefully).

INPUTS
    epm_Command =   EPC_GETSAMPLE
    epm_Data    =   NULL

RESULT
    epm_Data    -   points to current sam_info structure if successful.
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_GETSAMPLES, EPC_GETBUFFER.


NAME
    EPC_GETSAMPLES  (LOCK)

FUNCTION
    Get a pointer to the first element of an array of sam_info structs.
    The first element is the copy buffer.

INPUTS
    epm_Command =   EPC_GETSAMPLES
    epm_Data    =   NULL

RESULT
    epm_Data    -   points to first element if sucessful.
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_GETSAMPLE, EPC_GETBUFFER.


NAME
    EPC_GETBUFFER   (LOCK)

FUNCTION
    Get a pointer to the sam_info struct of the copy buffer. Currently
    this is the same as EPC_GETSAMPLES, but this could change at any
    time.

INPUTS
    epm_Command =   EPC_GETBUFFER
    epm_Data    =   NULL
```

```
RESULT
    epm_Data    -   points to copy buffer if sucessful.
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_GETSAMPLE, EPC_GETSAMPLES.


NAME
    EPC_GETSAMNUM   (LOCK)

FUNCTION
    Get a pointer to a ULONG which holds the current sample number (ie.
    the sample which is currently displayed in the window and is being
    edited).

INPUTS
    epm_Command =   EPC_GETSAMNUM
    epm_Data    =   NULL

RESULT
    epm_Data    -   points to a ULONG containing the sample number if
    successful.
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_GETSAMPLE, EPC_SETSAMNUM.


NAME
    EPC_GETRSTART   (LOCK)

FUNCTION
    Get a pointer to a ULONG containing the position in BYTES (from the
    start of the sample) of the start of the ranged area. This will be
    smaller or equal to the range end.

INPUTS
    epm_Command =   EPC_GETRSTART
    epm_Data    =   NULL

RESULT
    epm_Data    -   points to a ULONG holding the position in BYTES.
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_GETREND, EPC_SETRSTART, EPC_SETREND.


NAME
    EPC_GETREND     (LOCK)

FUNCTION
```

```
    Get a pointer to a ULONG containing the position of the end of the
    ranged area, in BYTES. This will be greater or equal to the range
    start.

INPUTS
    epm_Command =   EPC_GETREND
    epm_Data    =   NULL

RESULT
    epm_Data    -   points to a ULONG holding the position in BYTES.
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_GETRSTART, EPC_SETRSTART, EPC_SETREND.


NAME
    EPC_SETSAMNUM   (LOCK)

FUNCTION
    Set the current sample number.

INPUTS
    epm_Command =   EPC_SETSAMNUM
    epm_Data    -   pointer to a ULONG holding the new sample number.

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

BUGS
    Has not been tested properly.

SEE ALSO
    EPC_GETSAMNUM.


NAME
    EPC_SETRSTART   (LOCK)

FUNCTION
    Set the start of the range.

INPUTS
    epm_Command =   EPC_SETRSTART
    epm_Data    -   pointer to a ULONG holding the start of the range in
    BYTES

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

BUGS
    Has not been tested properly.

SEE ALSO
```

EPC_GETRSTART, EPC_GETREND, EPC_SETREND.

NAME
EPC_SETREND (LOCK)

FUNCTION
Set the end of the range.

INPUTS
epm_Command = EPC_SETREND
epm_Data - pointer to a ULONG holding the position of the end of the range in BYTES from the begining of the sample.

RESULT
epm_Error will equal EPCERR_NOERR if sucessful or an error defined in extproc.h.

BUGS
Has not been tested properly.

SEE ALSO
EPC_GETRSTART, EPC_GETREND, EPC_SETRSTART.

NAME
EPC_FLUSH (LOCK)

FUNCTION
Flush current sample from memory. This is the same as finding the current sample with EPC_GETSAMPLE and freeing the memory where it's waveform is stored:
FreeMem (sample->waveform, sample->length);
sample->length = 0;

INPUTS
epm_Command = EPC_FLUSH
epm_Data = NULL

RESULT
epm_Error will equal EPCERR_NOERR if sucessful or an error defined in extproc.h.

SEE ALSO

NAME
EPC_PIOPEN (LOCK)

FUNCTION
Open SamEd's Progress indicator window. This provides a standard means of notifying the user that an operation is taking place. A message may be placed in the window and the progress bar updated to reflect the current status.

INPUTS
epm_Command = EPC_PIOPEN

```
    epm_Data    =   NULL

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_PICLOSE, EPC_PIMAX, EPC_PIVAL, EPC_PISTRING,


NAME
    EPC_PISTRING    (LOCK)

FUNCTION
    Set the message in the Progress indicator window. Can be set before
    the window is opened. The string you pass is copied.

INPUTS
    epm_Command =   EPC_PISTRING
    epm_Data    =   pointer to a NULL terminated string

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_PICLOSE, EPC_PIMAX, EPC_PIVAL, EPC_PIOPEN,


NAME
    EPC_PIMAX       (LOCK)

FUNCTION
    Set the maximum value of the bar in the PI window. So if you are eg.
    scanning a sound sample you could set PIMAX to the length, and
    periodically update the bars position using EPC_PIVAL.

INPUTS
    epm_Command =   EPC_PIMAX
    epm_Data    =   pointer to a ULONG of the maximum length

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_PICLOSE, EPC_PIOPEN, EPC_PIVAL, EPC_PISTRING,


NAME
    EPC_PIVAL       (LOCK)

FUNCTION
    Set the value of the progress bar in the PI window. Should not be
    greater than PIMAX.

INPUTS
```

```
    epm_Command =   EPC_PIVAL
    epm_Data    =   pointer to a ULONG of the current position of the bar

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_PICLOSE, EPC_PIMAX, EPC_PIOPEN, EPC_PISTRING,


NAME
    EPC_PICLOSE     (LOCK)

FUNCTION
    Close SamEd's Progress indicator window. This provides a standard
    means of notifying the user that an operation is taking place. A
    message may be placed in the window and the progress bar updated to
    reflect the current status.

INPUTS
    epm_Command =   EPC_PICLOSE
    epm_Data    =   NULL

RESULT
    epm_Error will equal EPCERR_NOERR if sucessful or an error defined in
    extproc.h.

SEE ALSO
    EPC_PIOPEN, EPC_PIMAX, EPC_PIVAL, EPC_PISTRING,
```