NAME

`XNvrMpegDisplay` - the MPEG display widget class

SYNOPSIS

```
#include <nvr/NvrMpegDisplay.h>
```

DESCRIPTION

The `XNvrMpegDisplay` is a motif widget that encapsulates all of the mechanism for MPEG display and playback. The control resources available allow the programmer to access to playback direction, quality, speed, colormap selection and others. If you don't want to provide your own control buttons, a subclass of this widget, `XNvrMpegControl`, should be used.

NEW RESOURCES

MpegDisplay defines the following resources. The fifth column lists a code consisting of one or more of the letters C, S, and G. This code indicates whether the resource can be set when the widget is created (C), whether it can be set with `XtSetValues()` (S), and whether it can be read with `XtGetValues()` (G). This convention is adopted from the Motif Programmer's Reference Manual.

Name	Class	Type	Default	Access
XnvrNAction	XnvrCAction	NvrPlayer Action (int)	NvrStop	CS
XnvrNAudio Channel	XnvrCAudio Channel	int	First Available Channel	CSG
XnvrNAudio Quality	XnvrCAudio Quality	int	dynamic	CSG
XnvrNAudio QualitySteps	XnvrCAudio QualitySteps	int	dynamic	G
XnvrNBrightness	XnvrCBrightness	int	128	CSG
XnvrNContrast	XnvrCContrast	int	128	CSG
XnvrNDepth	XnvrCDepth	NvrDepth (int)	dynamic	CSG
XnvrNDepths	XnvrCDepths	long	dynamic	G
XnvrNInputFile	XnvrCInputFile	NvrInputSpec Ptr	NULL	CSG
XnvrNSeekPoint	XnvrCSeekPoint	NvrSeekPtr	dynamic	CSG
XnvrNState	XnvrCState	NvrPlayer State (int)	NvrPlayer Stopped	G

Name	Class	Type	Default	Access
XnvrNSizeScale	XnvrCSizeScale	NvrSize Scale (int)	NvrSize1x	CSG
XnvrNVideo Channel	XnvrCVideo Channel	int	First Available Channel	CSG
XnvrNVisual	XnvrCVisual	Visual *	dynamic	G

XnvrNAction The desired action for the player. Most commands are queued internally and are processed in the order recieved. For example, three XtSetValues() calls specifying NvrStepFwd will show all three frames, no matter how fast the XtSetValues() was performed, and no matter how long the pictures take to display. The two exceptions to this are NvrStop and setting the XnvrNSeekPoint resource; both actions will flush the playback queue. Possible values:

- NvrStop — Stop current playback; no effect if state is stopped.
- NvrPlayFwd — Play forward at normal speed.
- NvrPlayAll — Special playback mode that plays every frame as fast as possible, without audio.
- NvrFastFwd — Play forward at 3x speed without audio.
- NvrPlayBak — Play backward at normal speed without audio.
- NvrFastBak — Play backward at 3x speed without audio.
- NvrStepFwd — Advance to the next frame.
- NvrStepBak — Rewind one frame.

XnvrNAudioChannel
XnvrNVideoChannel

When read with `XtGetValues()` it is a bitmask of the available audio/video input channels in the multiplexed MPEG stream. When written with `XtSetValues()`, exactly one bit must be set from those available. A special enum value of `FirstAvailableChannel` is the default value, and signifies that the widget should choose the first channel available (if any), starting from channel 0.

XnvrNAudioQuality

Specifies the playback quality of the MPEG audio. The valid values for this resource are in the range 0 to `XnvrNAudioQualitySteps`-1. Changing the quality may change the sample rate and/or the number of output channels.

XnvrNAudioQualitySteps

Read-only. Specifies the number of steps in the audio quality settings.

XnvrNBrightness
XnvrNContrast

Specifies the contrast and brightness for the video display. The possible values are 0 to 255 with the default set to 128. Values above or below 128 will increase or decrease brighness and contrast, respectively.

`XnvrNDepth` Specifies the desired playback depth and colormap style. All of the depths available can be found in the `XnvrNDepths` resource. Each value below has exactly one bit set, and specifying a value for the depth resource must have only one bit set. Possible values:

- `NvrRgb888` 24-bit RGB color, possibly double buffered.
- `NvrRgb444` 12-bit RGB color, possibly double buffered.
- `NvrRgb332` 8-bit RGB color index.
- `NvrRgb232` 7-bit RGB color index. This is useful on 8-bit systems that have only one hardware colormap.
- `NvrGray8` 8-bit gray scale. Grayscale playback it typically 40% faster than color.
- `NvrGray7` 7-bit gray scale. This is useful on 8-bit systems that have only one hardware colormap.
- `NvrRgbC555` 8-bit RGB color index using 125 colormap entries.
- `NvrRgbC666` 8-bit RGB color index using 216 colormap entries.

`XnvrNDepths` Read-only. Specifies the available depths in a bitmask of the depths listed above. For example, if `NvrRgb444` is available, and the value of `XnvrNDepths` is placed in the variable `depth`, the the expression `(depth & NvrRgb444)` will be true. All depths available can be found in the `XnvrNDepths` resource.

`XnvrNInputFile`

Specifies the MPEG input stream. This may be an elementary audio or video stream, a system stream, an open file descriptor, a pair of descriptors or a TCP port. The resource is set by specifying a pointer to the union, `NvrInputSpecPtr`. The pointer may be freed immediately because all data is copied internally. Each structure in the union includes a type as the first member so the correct structure may be referenced. See the section titled input file specification. Possible type values are:

- `NvrInputSinglePath`
  Specifies a single pathname.
- `NvrInputPathPair`
  Specifies a pair of pathnames: audio and video.
- `NvrInputDescriptor`
  Specifies an open file descriptor.
- `NvrInputDescriptorPair`
  Specifies a pair of descriptors: audio and video.
- NvrInputTcpPort
  tcp port number for listening

Note that specifying a new input file while another sequence is playing simply queues the sequence to play after the current one finishes. To proceed immediately to the next sequence, you must send it a stop action.

`XnvrNSeekPoint`

Specifies a desired seek point in the specifed input stream, measured in bytes. The resource is set by specifying a pointer to the union, `NvrSeekPtr`. The pointer may be freed immediately because all data is copied internally. This is so that two offsets can be described for `NvrInputPathPair` and `NvrInput-DescriptorPair`. For `NvrTcpPort`, the seek is ignored.

For seeks on single path of descriptors, the actual frame found will be the first I-frame (reference frame) following the seek point, and the audio will be set to the correct sample within the audio frame. For seeks on path pairs or descriptor pairs, The first frame will be as described above; the audio will seek to the first audio frame boundary following the seek point.

XnvrNSizeScale

Specifies the size of the output playback window. The widget insists on a fixed size based on the input material and this resource. The actual width and height can be queried via the `XmNwidth` and `XmNheight` resources. Possible values:

`NvrSize1x` The output playback window equals the input stream size.

`NvrSize4x` The output playback window is twice the width and twice the height of the input stream size.

`NvrSizeQuarter`

The output playback window is half the width and half the height of the input stream size.

`XnvrNState` Read-only. Current state of the playback engine. The state is asynchronous with respect to a specified action and may lag by an unspecified amount of time. Possible values:

`NvrPlaying` Playback in progress. This may mean any direction or any speed, including single stepping.

`NvrStopped` Stopped and waiting for commands.

`XnvrNVisual` Read-only. The visual of the selected depth.

CALLBACK RESOURCES

XmNexposeCallback

List of callbacks that are called when the widget recieves an exposure. The callback function is passed a pointer to the following structure as the third argument of the XtCallbackProc which has the same style as the expose callback for the Motif drawing area widget. Note that the intent of this callback is so that the programmer can know that the display window is visible and that playback can begin, if appropriate.

```
typedef struct {
      int reason;        /* XmCR_EXPOSE, XmCR_INPUT or XmCR_RESIZE */
      XEvent *event;     /* event structure that triggered */
                         /* the callback */
      Window window;     /* the widget's window */
} NvrMpegExposeCbInfo, *NvrMpegExposeCbInfoPtr;
```

XnvrNStateCallback

List of callbacks that are called when the playback engine changes state. The callback function is passed a pointer to the following structure as the third argument of the XtCallbackProc. All elements are valid for all callbacks; the reason element gives the reasons for the callback.

```
typedef struct {
      int cbMask;        /* mask of NvrMpegDisplayCbReason */
      NvrPlayerState playState;
      int frameNumber;   /* current frame number */
      long timeCode;     /* current time */
      long lastFrame;    /* last frame in sequence */
      Boolean lastApprox;/* true: 'lastFrame' is only approximate */
      int width;         /* input width */
      int height;        /* input height */
      float frameRate;   /* video input frame rate */
      int sampleRate;    /* audio sample rate */
      int channels;      /* 1=mono; 2=stereo */
} NvrMpegDisplayCbInfo, *NvrMpegDisplayCbInfoPtr;
```

The reason member is a bitmask of one or more of the following values:

NvrAudioChange

Specified when the input audio sampleRate or channels change. It is not called when XnvrNAudioQuality changes.

NvrEofReached

Specified when eof detected in sequence. This notification can be used to stream multiple sequences together during playback by immediately setting XnvrNInputFile.

NvrFrameNumber

Specified when frameNumber and timeCode changes. This occurs during playback; however , the frame numbers returned are not gururanteed to be consecutive.

`NvrNewPlayState`
: Specified when the `XnvrNState` resource has changed. The value is supplied in `playState`.

`NvrNewVideo`
: Specified when `XnvrNVideoChannels` changes or when a new sequence is selected with `XnvrNInputFile`. Note that width and height may be zero before a video channel has been selected for the first time.

`NvrNewAudio`
: Specified when `XnvrNAudioChannels` changes or when a new sequence is selected with `XnvrNInputFile`. Note that audio `sampleRate` and `channels` may be zero before an audio channel has been selected for the first time.

`NvrVideoChange`
: Specified when the input `width`, `height` or frameRate changes , such as when a new sequence is selected with `XnvrNInputFile`. The callback is expected to resize the enclosing widget to match the input size, scaled by the value specified in `XnvrNSizeScale`. Note: this is not called when `XnvrNSizeScale` changes.

INPUT FILE SPECIFICATION

```
typedef union _nvrInputSpec {
      NvrInputType type;
      struct {
            NvrInputType type;/* must be NvrInputSinglePath */
            char *path;       /* audio, video or multiplexed file */
      } single;
      struct {
            NvrInputType type;/* must be NvrInputPathPair */
            char *audio;      /* path to separate audio file */
            char *video;      /* path to separate video file */
      } pathPair;
      struct {
            NvrInputType type;/* must be NvrTcpPort */
            char *host;       /* if specified, this host only; */
                              /* otherwise, accept any host */
            int port;
      } net;
      struct {
            NvrInputTypetype; /* must be NvrInputDescriptor */
            int fd;           /* file descriptor of open file */
      } descriptor;
      struct {
            NvrInputTypetype; /* must be NvrInputDescriptorPair */
            int vfd;          /* file descriptor of video file */
            int afd;          /* file descriptor of audio file */
```

```
                } descriptor;
        } NvrInputSpec, *NvrInputSpecPtr;
```

SEEK SPECIFICATION

```
        typedef union _nvrSeek {
              NvrInputType type;
              struct {
                    NvrInputType type;/* must be NvrInput{SinglePath, *
                                      /* Descriptor} */
                    long offset;
              } single;
              struct {
                    NvrInputType type;/* must be NvrInput{Path, */
                                      /* Descriptor}Pair */
                    long vOffset;     /* offset for video file */
                    long aOffset;     /* offset for audio file */
              } single;
        } NvrSeek, *NvrSeekPtr;
```

SEE ALSO

`XNvrMpegControl`

COPYRIGHT