



IMoraeVideoDraw interface

Purpose of interface:

This interface allows a plug-in draw on the playback video in Manager. Drawing can take place while the user is viewing a recording in Manager. It can also be used to modify a highlight video as it is produced in Manager. At this time (Morae release 3.2.0), drawing is only allowed on the main video stream.

This interface can be implemented by a plug-in that also implements IMoraeEventStream to allow the user to visualize events from a non-standard event stream that is captured by a Recorder plug-in. This is not a requirement. Other potential uses for plug-ins of this type would be to provide mark-up capabilities, allowing the user to annotate the video or to provide visualization of existing event streams that are already captured as part of Morae's RRT data.

This interface derives from IUnknown.

Note: Methods that return BSTRs that will be displayed to the user all use a locale ID to indicate the language. At this time (Morae release 3.2.0), only Morae is only offered in an English version.

Note: To be loaded by Manager and used for event capture, the plug-in must register its CLSID with the operating system and register itself as implementing the CATID_ManagerVideoDraw category:

5A245B48-94A3-4A37-9468-57F3807348E6

IMoraeVideoDraw specifications:

Interface ID:

IID_ IMoraeVideoDraw = E9215EFD-F193-47EF-A2C7-B8C0D0505470

Enumerations

1. RenderingHints

- a. RenderingNotRealtime

This flag indicates that the plug-in does not need to draw in real time because the call is being made in the context of producing a highlight video rather than updating the video as it is being rendered on the screen.



Currently, there is only one flag in this enumeration. Any flags in this enumeration can be or'ed together and provided as the `uRenderingHints` input for the `GiveUpcomingFrame` method.

Methods

1. HRESULT GetName(

BSTR* pStr, [out]
long localeID) [in]

This requests the name that will be displayed to the user to identify this plug-in. The `localeID` should be used to determine the language. The plug-in must allocate the system resources for the BSTR. The caller will be responsible for de-allocation of resources.

2. HRESULT GetDescription (

BSTR* pStr, [out]
long localeID) [in]

This requests a description of what this plug-in will do. If this plug-in implements a setup dialog that allows the user to modify its behavior, the description can be used to indicate the current settings. The `localeID` should be used to determine the language. The plug-in must allocate the system resources for the BSTR. The caller will be responsible for de-allocation of resources.

3. HRESULT HasSetupDialog(

VARIANT_BOOL* pBool) [out]

Manager will call this method to determine whether the plug-in has a setup dialog to display. If the plug-in sets `pBool` to `VARIANT_TRUE`, it must be able to show a dialog in the `ShowSetupDialog` method call. Otherwise, `ShowSetupDialog` will not be called.

4. HRESULT ShowSetupDialog (

VARIANT varHwndParent, [in]
LCID localeID) [in]

This requests that the plug-in display a setup dialog to allow the user to configure draw parameters. The `localeID` input should determine the language that the dialog should employ. The `varHwndParent` input gives the `HWND` that the dialog should use as a parent window. The plug-in should be able to accept either a 32-bit or 64-bit number, either signed or unsigned. A return value of `S_FALSE` will indicate that the user has dismissed the dialog without changing the setup. If this method returns `S_OK`, Morae Manager will assume that the user has modified the draw criteria.

5. HRESULT SetRecording(

IMoraeRecording* pRecording) [in]

This provides an interface to the Recording that the current instance of the plug-in can draw on. See the documentation on the `IMoraeRecording` interface for information on the methods that can be called on this interface. If the plug-in stores this interface pointer, it should call `AddRef`



on it. Each AddRef call should be balanced by a Release call when the plug-in no longer needs this interface.

6. HRESULT SetProject(

IMoraeProject* pProject) [in]

This provides an interface to the project in which the recording is found. See the documentation on the IMoraeProject interface for information on the methods that can be called on this interface. If the plug-in stores this interface pointer, it should call AddRef on it. Each AddRef call should be balanced by a Release call when the plug-in no longer needs this interface. Note that the IMoraeProject interface corresponds to a study in the Morae Manager user interface.

7. HRESULT GiveUpcomingFrame(

BSTRBLOB* pBitmapInfo, [in]
BSTRBLOB* pBitmapBytes, [in, out]
unsigned hyper rtStart, [in]
unsigned hyper rtEnd, [in]
int nOriginX, [in]
int nOriginY, [in]
double dblScaleX, [in]
double dblScaleY, [in]
ULONG uRenderingHints) [in]

This call is made to provide a video frame (in the form of a bitmap) to the plug-in for drawing. The pData member of the *pBitmapInfo can be cast to a pointer to a windows BITMAPINFO structure to get information about the bitmap. The pData member of the *pBitmapBytes is a pointer to the actual bitmap data that the plug-in is allowed to modify. rtStart and rtEnd refer to the start and end time for the frame, respectively. These times are given in 100 nanosecond units from the start of the recording. nOriginX and nOriginY provide offsets (in pixels) to translate from the origin of the bitmap to the origin of the actual video. dblScaleX and dblScaleY provide scaling factors to translate from the original video to the bitmap. (A scaling factor of 1.0 indicates no scaling. Smaller values indicate that the bitmap is scaled down from the original video.) The uRenderingHints will be a bitwise combination of RenderingHints enum values. Note that if the RenderingNotRealtime flag is not set, the plug-in should perform its operations quickly enough to not fall behind the video playback. If drawing lags, the user will be prompted to turn off one or more draw plug-ins.

8. HRESULT GetDrawSetupParameters (

LPSAFEARRAY* ppParams) [out]

Manager will make this call to ask for draw setup information that can be stored and applied to the plug-in in the SetDrawSetupParameters call. The plug-in will allocate resources for the array. The caller will be responsible for de-allocation of resources. The datatype of this array



cannot be VT_DISPATCH, VT_UNKNOWN, or VT_VARIANT. All elements of the array must be of the same fixed size. VT_BSTR is an acceptable datatype.

9. HRESULT SetDrawSetupParameters (

LPSAFEARRAY pParams) [in]

This will set the criteria to be used to perform drawing. The criteria used in this call would have been acquired in a call to GetDrawSetupParameters. If this method is called before a call to ShowSetupDialog, the settings in the dialog should reflect these settings. Note that the caller is responsible for de-allocation of the SAFEARRAY.