



## IMoraeProject interface

### Purpose of interface:

This interface provides a plug-in with information about a group of Morae recordings with a common set of configuration data. In the Morae Manager UI, this grouping is referred to as a Study. It is implemented by Morae Manager. Several plug-in interfaces (IMoraeEventSearch, IMoraeEventStream, IMoraeVideoDraw) are given an IMoraeProject interface to provide access to details about the study and all of its recordings.

This interface derives from IUnknown.

### IMoraeProject specifications:

#### Interface ID:

IID\_IMoraeProject = 48000048-23DD-426C-843B-D6A888F521FD

#### Methods

1. HRESULT GetRecording (   
 REFIID uuidRec, [in]   
 IMoraeRecording\* pplRecording ) [out]   
 This call gets an interface to a specific recording. The caller is responsible for calling Release on \*pplRecording when it is no longer needed.
2. HRESULT GetAllAppNames (   
 VARIANT\* pVar ) [out]   
 This call outputs the “friendly” names of all applications that were running within all recordings in a study. The “friendly” names are the names displayed by Manager to the user rather than the actual executable file names. If successful, \*pVar will be set to a SAFEARRAY of BSTRs (type = VT\_ARRAY | VT\_BSTR). The caller is responsible for freeing all system resources.
3. HRESULT EnumRecordings (   
 IEnumMoraeRecording\*\* pplEnum) [out]   
 This call retrieves an enumerator that can be used to access all of the recordings in the study.
4. HRESULT GetMarkerStrings (   
 LCID localeID, [in]   
 int nMarkerID, [in]   
 BSTR\* pstrType, [out]   
 BSTR\* pstrDefn ) [out]



For a given marker ID, this outputs the type and definition strings. The definition is not affected by the localeID. The caller is responsible for freeing the strings.

5. HRESULT GetMarkerScores (

int nMarkerID, [in]  
LPSAFEARRAY\* ppArrayScores ) [out]

For a given marker ID, this call outputs the valid score IDs as a SAFEARRAY of VT\_I4 values. The caller is responsible for freeing the system resources allocated for this array.

6. HRESULT GetMarkerScoreString (

int nMarkerID, [in]  
int nScore, [in]  
BSTR\* pstr ) [out]

This looks up string that the user has assigned to describe a marker score ID. The caller is responsible for freeing the string.

7. HRESULT GetTaskIDs (

LPSAFEARRAY\* ppTaskIDs ) [out]

This method outputs an array of all ID values for the tasks that are configured for the study. The array data type will be VT\_UI4. The caller is responsible for freeing the array.

8. HRESULT GetTaskDefinition (

UINT nTaskID, [in]  
BSTR\* pstrName, [out]  
BSTR\* pstrDefn ) [out]

This outputs the name and definition that the user has assigned to a particular task. The caller is responsible for freeing the string resources.

9. HRESULT GetTaskScores (

UINT nTaskID, [in]  
LPSAFEARRAY\* ppArrayScores ) [out]

For a given task ID, this call outputs the valid score IDs as a SAFEARRAY of VT\_UI4 values. The caller is responsible for freeing the system resources allocated for this array.

10. HRESULT GetTaskScoreString (

UINT nTaskID, [in]  
int nScore, [in]  
BSTR\* pStr ) [out]

This looks up string that the user has assigned to describe a task score ID. The caller is responsible for freeing the string.



11. HRESULT GetSurveyQuestions (

REFGUID uuidSurvey, [in]  
LPSAFEARRAY\* ppQuestions) [out]

This outputs an array of questions for a particular survey. The output is a SAFEARRAY of BSTRs. The caller is responsible for freeing the resources.

12. HRESULT GetSurveyQuestionStyles (

REFGUID uuidSurvey, [in]  
LPSAFEARRAY\* ppStyles ) [out]

This outputs an array of styles for the questions in a particular survey. The output is a SAFEARRAY of VT\_I4 values. The caller is responsible for freeing the resources.

13. HRESULT GetSurveyQuestionChoices (

REFGUID uuidSurvey, [in]  
LPSAFEARRAY\* ppChoices ) [out]

This method gets the choices for all questions for a particular survey. It outputs an array of VARIANTS. Each of these VARIANTS corresponds to a question. Each VARIANT contains a SAFEARRAY of BSTRs. Each BSTR is a survey choice for the question. The caller is responsible for freeing all resources.

14. HRESULT GetSurveyQuestionCount (

REFGUID uuidSurvey, [in]  
unsigned long\* pnCount ) [out]

This method outputs the number of questions found in a particular survey.

15. HRESULT GetSurveyChoiceCounts (

REFGUID uuidSurvey, [in]  
LPSAFEARRAY\* ppChoiceCounts ) [out]

This call outputs the number of choices for each question in a particular survey.

\*ppChoiceCounts will be a SAFEARRAY of VT\_UI4 values. The caller is responsible for freeing all resources.

16. HRESULT GetSurveyStyle (

REFGUID uuidSurvey, [in]  
int\* pnStyle) [out]

This call outputs the style of a particular survey. \*pnStyle will be set to 1 for a SUS survey or 2 for a custom survey.

17. HRESULT GetSurveyName (

REFGUID uuidSurvey, [in]  
BSTR\* pstrName ) [out]

This call outputs the name of a particular survey. The caller is responsible for freeing the string.



18. HRESULT GetSurveyIDs (

LPSAFEARRAY\* ppSurveyIDs ) [out]

This call outputs the ID values of all surveys in the study. \*ppSurveyIDs will be set to a SAFEARRAY of BSTRs. Each BSTR will be the string form of a guid. The caller is responsible for freeing all resources.