sceye-Sphere Storage-API Interface documentation

TABLE OF CONTENTS

- 1. /ApiGetLicense
- 2. /ApiInit
- 3. /ApiRelease
- 4. /ArchiveCleanup
- 5. /ArchiveClose
- 6. /ArchiveCreate
- 7. /ArchiveDelete
- 8. /ArchiveGetCount
- 9. /ArchiveGetInfo
- 10. /ArchiveOpen
- 11. /ArchiveRename
- 12. /ArchiveUpdateAbort
- 13. /ArchiveUpdateAddPage
- 14. /ArchiveUpdateBegin
- 15. /ArchiveUpdateCommit
- 16. /ArchiveUpdateSetDocInfo
- 17. /ClipboardClear
- 18. /ClipboardClearDoc
- 19. /ClipboardCopyDocToClipboard
- 20. /ClipboardCopyPageToClipboard
- 21. /ClipboardGetDocCount
- 22. /ClipboardGetDocInfo
- 23. /ClipboardGetImageData
- 24. /ClipboardGetInformation
- 25. /ClipboardGetPageInfoByIdentifier
- 26. /ClipboardGetPageInfoByIndex
- 27. /ClipboardGetUsedDiskSpace
- 28. /ClipboardPasteDocument
- 29. /ClipboardPastePage
- 30. /DocumentGetCount
- 31. /DocumentGetDetails
- 32. /DocumentGetInfoByIdentifier
- 33. /DocumentGetInfoByIndex
- 34. /DocumentQueryExistence
- 35. /DocumentSetDetails
- 36. /DocumentSetPageFilter
- 37. /GlobalSettingRead
- 38. /GlobalSettingWrite
- 39. /ImageAddData
- 40. /ImageGetData
- 41. /PageGetDetails
- 42. /PageGetInfoByDocIdAndIndex
- 43. /PageGetInfoByIdentifier
- 44. /PageGetInfoByIndex
- 45. /PageQueryExistence
- 46. /PageSetDetails
- 47. /PageSetRotation
- 48. /TrashboxClear
- 49. /TrashboxClearDoc
- 50. /TrashboxCopyDocToTrash
- 51. /TrashboxCopyPageToTrash
- 52. /TrashboxGetDocCount
- 53. /TrashboxGetDocInfo
- 54. /TrashboxGetImageData
- 55. /TrashboxGetInformation
- 56. /TrashboxGetPageInfoByIdentifier
- 57. /TrashboxGetPageInfoByIndex
- 58. /TrashboxGetUsedDiskSpace
- 59. /TrashboxRestoreDocument
- 60. /TrashboxRestorePage
- 61. /WorkspaceGetDetails
- 62. /WorkspaceGetInfo
- 63. /WorkspaceSetDetails
- 64. /WorkspaceSetDocFilter
- 65. /WorkspaceSetFilter
- 66. /WorkspaceSettingRead
- 67. /WorkspaceSettingWrite

1. /ApiGetLicense [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ApiGetLicense   (wchar_t* strLicenseName, unsigned long iMaxBufferSize_LicName,
                                  wchar_t* strLicenseNumber, unsigned long iMaxBufferSize_LicNumber)
```

FUNCTION

Returns the LicenseName and the LicenseNumber (provided by Silvercreations) of the StorageAPI DLL

INPUTS

```
strLicenseName             -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_LicName     -- max. text length of the previous wchar_t buffer
strLicenseNumber           -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_LicNumber   -- max. text length of the previous wchar_t buffer
```

OUTPUT

```
LicenseName (via strLicenseName buffer)
LicenseNumber (via strLicenseNumber buffer)
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

2. /ApiInit [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ApiInit (void** pApiData,
                          const wchar_t* strApiSettings)
```

FUNCTION

Initializes the StorageAPI DLL

INPUTS

```
pApiData -- pointer to the internal data struct (Sphere SDK stores the pointer internally)
strApiSettings -- additional API settings (not used yet)
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ApiRelease

3. /ApiRelease [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ApiRelease (void** pApiData)
```

FUNCTION

Releases the ApiData structure

INPUTS

pApiData -- pointer to the internal data struct

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ApiInit

4. /ArchiveCleanup [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveCleanup(void* pApiData)
```

FUNCTION

Removes all objects (documents, images) which are not referenced anymore

INPUTS

pApiData -- pointer to the internal data struct

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ArchiveOpen ArchiveClose ArchiveCreate ArchiveDelete ArchiveGetCount ArchiveGetInfo ArchiveRename

5. /ArchiveClose [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveClose(void* pApiData)
```

FUNCTION

Closes a previously opened archive

INPUTS

pApiData -- pointer to the internal data struct

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ArchiveOpen ArchiveCleanup ArchiveCreate ArchiveDelete ArchiveGetCount ArchiveGetInfo ArchiveRename

6. /ArchiveCreate [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveCreate   (void* pApiData,
                                  const wchar_t* strName,
                                  bool bReadOnly,
                                  long* iCreatedArchiveIndex)
```

FUNCTION

Creates a new archive

INPUTS

pApiData -- pointer to the internal data struct
strName -- name of the archive

```
bReadOnly -- bool flag to indicate that the archive is read only
iCreatedArchiveIndex -- pointer to a long variable which gets filled by the function
```

OUTPUT

```
iCreatedArchiveIndex (index of the created archive (0-based))
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

ArchiveOpen ArchiveClose ArchiveCleanup ArchiveDelete ArchiveGetCount ArchiveGetInfo ArchiveRename

7. /ArchiveDelete [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveDelete   (void* pApiData,
                                  long iArchiveIndex)
```

FUNCTION

```
Deletes an archive
```

INPUTS

```
pApiData -- pointer to the internal data struct
iArchiveIndex -- index of the archive that will be deleted (0-based)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

ArchiveOpen ArchiveClose ArchiveCreate ArchiveCleanup ArchiveGetCount ArchiveGetInfo ArchiveRename

8. /ArchiveGetCount [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveGetCount (void* pApiData,
                                  long* iArchiveCount)
```

FUNCTION

```
Retrieves the archive count
```

INPUTS

```
pApiData -- pointer to the internal data struct
iArchiveCount -- pointer to a long variable which will be filled with the archive count
```

OUTPUT

```
iArchiveCount (count of the archives found in archive base path)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

ArchiveOpen ArchiveClose ArchiveCreate ArchiveDelete ArchiveCleanup ArchiveGetInfo ArchiveRename

9. /ArchiveGetInfo [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveGetInfo  (void* pApiData,
                                  long iArchiveIndex,
                                  wchar_t* strArchiveName, long iMaxBufferSize_Name,
                                  bool* bReadOnly,
                                  long* iDocCount,
                                  wchar_t* strArchiveCreated, long iMaxBufferSize_Created,
                                  wchar_t* strArchiveModified, long iMaxBufferSize_Modified)
```

FUNCTION

```
Retrieves archive metadata for the specified archive index
```

INPUTS

```
pApiData -- pointer to the internal data struct
iArchiveIndex -- index of the selected archive (0-based)
strArchiveName -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Name -- max. text length of the previous wchar_t buffer
bReadOnly -- pointer to a bool variable which indicates if the archive is read-only
iDocCount -- pointer to a long variable which stores the number of documents
             stored in the selected archive
strArchiveCreated -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Created -- max. text length of the previous wchar_t buffer
strArchiveModified -- pointer to a wchar_t buffer which gets filled by this function
iMaxBuffersize_Modified -- max. text length of the previous wchar_t buffer
```

OUTPUT

```
strArchiveName (name of the archive)
bReadOnly (read-only flag)
iDocCount (document count)
strArchiveCreated (archive creation timestamp)
strArchiveModified (archive modification timestamp)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

ArchiveOpen ArchiveClose ArchiveCreate ArchiveDelete ArchiveGetCount ArchiveCleanup ArchiveRename

10. /ArchiveOpen [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveOpen (void* pApiData,
                              long iArchiveIndex)
```

FUNCTION

```
Opens the selected archive
```

INPUTS

```
pApiData -- pointer to the internal data struct
iArchiveIndex -- index of the selected archive (0-based)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

ArchiveCleanup ArchiveClose ArchiveCreate ArchiveDelete ArchiveGetCount ArchiveGetInfo ArchiveRename

11. /ArchiveRename [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveRename   (void* pApiData,
                                  long iArchiveIndex,
                                  const wchar_t* strNewArchiveName)
```

FUNCTION

Changes the name of the selected archive

INPUTS

```
pApiData -- pointer to the internal data struct
iArchiveIndex -- index of the archive that will be renamed (0-based)
strNewArchiveName -- name of the new archive
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ArchiveOpen ArchiveClose ArchiveCreate ArchiveDelete ArchiveGetCount ArchiveGetInfo ArchiveCleanup

12. /ArchiveUpdateAbort [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveUpdateAbort (void* pApiData)
```

FUNCTION

Aborts an archive update

INPUTS

```
pApiData -- pointer to the internal data struct
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ArchiveUpdateAddPage ArchiveUpdateBegin ArchiveUpdateCommit ArchiveUpdateSetDocInfo

13. /ArchiveUpdateAddPage [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveUpdateAddPage    (void* pApiData,
                                          const wchar_t* strPageIdentifier,
                                          const wchar_t* strSource,
                                          const wchar_t* strCreatedTimestamp,
                                          const wchar_t* strModifiedTimestamp,
                                          long iRotationAngle)
```

FUNCTION

adds a page to an document during an update archive procedure

INPUTS

```
pApiData -- pointer to the internal data struct
strPageIdentifier -- GUID of the page that will be added
strSource -- source of the page
strCreatedTimestamp -- page creation timestamp
strModifiedTimestamp -- page modification timestamp
iRotationAngle -- rotation angle of the page (values between 0 and 359, no negative values)
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ArchiveUpdateAbort ArchiveUpdateBegin ArchiveUpdateCommit ArchiveUpdateSetDocInfo

14. /ArchiveUpdateBegin [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveUpdateBegin  (void* pApiData,
                                      long iUpdateType,
                                      long iUpdateIndex)
```

FUNCTION

start an update archive procedure

INPUTS

pApiData -- pointer to the internal data struct
iUpdateType -- the update type
iUpdateIndex -- the update index

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ArchiveUpdateAddPage ArchiveUpdateAbort ArchiveUpdateCommit ArchiveUpdateSetDocInfo

15. /ArchiveUpdateCommit [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveUpdateCommit (void* pApiData,
                                      wchar_t* strDocIdentifier,
                                      long iBufferSize_DocIdentifier)
```

FUNCTION

executes the update archive procedure

INPUTS

pApiData -- pointer to the internal data struct
strDocIdentifier -- pointer to a wchar_t buffer which gets filled by this function
iBufferSize_DocIdentifier -- max. text length of the previous wchar_t buffer

OUTPUT

strDocIdentifier (Identifier of the created/modified document)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ArchiveUpdateAddPage ArchiveUpdateBegin ArchiveUpdateAbort ArchiveUpdateSetDocInfo

16. /ArchiveUpdateSetDocInfo [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ArchiveUpdateSetDocInfo (void* pApiData,
                                          const wchar_t* strDocIdentifier,
```

```
                                        const wchar_t* strVersion,
                                        const wchar_t* strDocType,
                                        const wchar_t* strDocName,
                                        const wchar_t* strCreatedTimestamp,
                                        const wchar_t* strModifiedTimestamp)
```

FUNCTION

set document metadat during update archive procedure

INPUTS

```
pApiData -- pointer to the internal data struct
strDocIdentifier -- document identifier of the document
strVersion -- version of the document metadata information (currently "1")
strDocType -- type of the document
strDocName -- name of the document
strCreatedTimestamp -- document creation timestamp
strModifiedTimestamp -- document modification timestamp
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ArchiveUpdateAddPage ArchiveUpdateBegin ArchiveUpdateCommit ArchiveUpdateAbort

17. /ClipboardClear [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardClear (void* pApiData)
```

FUNCTION

clears the internal clipboard (remove all clipboard content)

INPUTS

```
pApiData -- pointer to the internal data struct
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

18. /ClipboardClearDoc [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardClearDoc   (void* pApiData,
                                      const wchar_t* strDocumentIdentifier)
```

FUNCTION

clears a document in the internal clipboard (remove a specific document from the clipboard)

INPUTS

```
pApiData -- pointer to the internal data struct
strDocumentIdentifier -- the identifier of the document that will be removed
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ClipboardClear ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

19. /ClipboardCopyDocToClipboard [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardCopyDocToClipboard (void* pApiData,
                                              const wchar_t* strDocIdentifier)
```

FUNCTION

copies a document (specified by identifier) from the current archive to the clipboard

INPUTS

pApiData -- pointer to the internal data struct
strDocIdentifier -- the identifier of the document that will be copied

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ClipboardClearDoc ClipboardClear ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

20. /ClipboardCopyPageToClipboard [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardCopyPageToClipboard    (void* pApiData,
                                                  const wchar_t* strPageIdentifier)
```

FUNCTION

copies a page (specified by identifier) from the current archive to the clipboard

INPUTS

pApiData -- pointer to the internal data struct
strPageIdentifier -- the identifier of the page that will be copied

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardClear ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

21. /ClipboardGetDocCount [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardGetDocCount    (void* pApiData,
                                          long* iDocCount)
```

FUNCTION

retrieves the document count from the clipboard

INPUTS

```
pApiData -- pointer to the internal data struct
iDocCount -- pointer to a long variable which will be filled during execution of this method
```

OUTPUT

```
iDocCount (number of documents currently stored in clipboard)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardClear
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

22. /ClipboardGetDocInfo [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardGetDocInfo (void* pApiData,
                                      long iDocIndex,
                                      wchar_t* strDocID, long iMaxBufferSize_DocID,
                                      wchar_t* strDocType, long iMaxBufferSize_Type,
                                      wchar_t* strDocName, long iMaxBufferSize_Name,
                                      wchar_t* strDocCreatedTimestamp, long iMaxBufferSize_Created,
                                      wchar_t* strDocModifiedTimestamp, long iMaxBufferSize_Modified,
                                      long* iPageCount)
```

FUNCTION

```
retrieves the document metadata from the clipboard
```

INPUTS

```
pApiData -- pointer to the internal data struct
iDocIndex -- index of the selected document
strDocID -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_DocID -- max. text length of the previous wchar_t buffer
strDocType -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Type -- max. text length of the previous wchar_t buffer
strDocName -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Name -- max. text length of the previous wchar_t buffer
strDocCreatedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Created -- max. text length of the previous wchar_t buffer
strDocModifiedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Modified -- max. text length of the previous wchar_t buffer
iPageCount -- pointer to a long variable which gets filled during method execution
```

OUTPUT

```
strDocID (identifier of the selected document)
strDocType (type of the selected document)
strDocName (name of the selected document)
strDocCreatedTimestamp (selected document creation timestamp)
strDocModifiedTimestamp (selected document modification timestamp)
iPageCount (selected document page count)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardClear ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

23. /ClipboardGetImageData [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardGetImageData   (void* pApiData,
                                          const wchar_t* strImageIdentifier,
                                          const wchar_t*  strImageType,
                                          unsigned char* pImageDataBuffer, long iMaxSize_ImageDataBuffer)
```

FUNCTION

retrieves the binary image data from the clipboard

INPUTS

pApiData -- pointer to the internal data struct
strImageIdentifier -- identifier of the selected image
strImageType -- type of the selected image
pImageDataBuffer -- pointer to a data buffer which gets filled with the binary image data
iMaxSize_ImageDataBuffer -- maximum size of the previous data buffer

OUTPUT

iImageDataBuffer (holds the binary image data)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

NOTES

strImageType can be one of the following
* FULL
* PREVIEW2
* PREVIEW4
* PREVIEW8

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardClear ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

24. /ClipboardGetInformation [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardGetInformation (void* pApiData,
                                          wchar_t* strVersion, long iMaxBuffer_Version,
                                          wchar_t* strCreatedTimestamp, long iMaxBuffer_Created,
                                          wchar_t* strModifiedTimestamp, long iMaxBuffer_Modified,
                                          long* iDocCount)
```

FUNCTION

retrieves the metadata information of the clipboard

INPUTS

pApiData -- pointer to the internal data struct
strVersion -- pointer to a wchar_t buffer which gets filled by this function
iMaxBuffer_Version -- max. text length of the previous wchar_t buffer
strCreatedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBuffer_Created -- max. text length of the previous wchar_t buffer
strModifiedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBuffer_Modified -- max. text length of the previous wchar_t buffer
iDocCount -- pointer to a long variable which gets filled by this function

OUTPUT

strVersion (version number of the clipboard (currently "1")
strCreatedTimestamp (clipboard creation timestamp)
strModifiedTimestamp (clipboard modification timestamp)
iDocCount (count of documents currently stored in clipboard)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardClear ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

25. /ClipboardGetPageInfoByIdentifier [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardGetPageInfoByIdentifier    (void* pApiData,
                                                      const wchar_t* strPageIdentifier,
                                                      const wchar_t* strInfoType,
                                                      long* iWidth,
                                                      long* iHeight,
                                                      long* iFileSize,
                                                      long* iResolution,
                                                      wchar_t* strColormode, long iMaxBufferSize_Colormode,
                                                      wchar_t* strSource, long iMaxBufferSize_Source,
                                                      wchar_t* strFileExtension, long iMaxBufferSize_FileExtension,
                                                      long* iPageRotation)
```

FUNCTION

retrieves the page metadata information of the clipboard
usually this method gets called prior to a call to ClipboardGetImageData
(to determine the right size of the image buffer to be allocated)

INPUTS

pApiData -- pointer to the internal data struct
strPageIdentifier -- the identifier of the selected page
strInfoType -- the selected information type you want to retrieve
iWidth -- pointer to a long variable which gets filled by this function
iHeight -- pointer to a long variable which gets filled by this function
iFileSize -- pointer to a long variable which gets filled by this function
iResolution -- pointer to a long variable which gets filled by this function
strColormode -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Colormode -- max. text length of the previous wchar_t buffer
strSource -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Source -- max. text length of the previous wchar_t buffer
strFileExtension -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_FileExtension -- max. text length of the previous wchar_t buffer
iPageRotation -- pointer to a long variable which gets filled by this function

OUTPUT

iWidth (the width of the selected image in pixel)
iHeight (the height of the selected image in pixel)
iFileSize (the filesize of the selected image)
iResolution (the resolution of the selected image in dpi)
strColormode (the colormode of the selected image)
strSource (the source where the selected page came from)
strFileExtension (the file extension of the selected image)
iPageRotation (the rotation of the page in degrees, 0 to 359 degrees, no negative values)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

NOTES

strInfoType can be one of the following
* FULL
* PREVIEW2
* PREVIEW4
* PREVIEW8

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardClear
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

26. /ClipboardGetPageInfoByIndex [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardGetPageInfoByIndex (void* pApiData,
                                              long iDocIndex,
                                              long iPageIndex,
                                              const wchar_t* strInfoType,
                                              wchar_t* strPageIdentifier, long iMaxBufferSize_PageID,
                                              long* iWidth,
                                              long* iHeight,
                                              long* iFileSize,
                                              long* iResolution,
                                              wchar_t* strColormode, long iMaxBufferSize_Colormode,
                                              wchar_t* strSource, long iMaxBufferSize_Source,
                                              wchar_t* strFileExtension, long iMaxBufferSize_FileExtension,
                                              long* iPageRotation)
```

FUNCTION

```
retrieves the page metadata information of the clipboard
(page selected by doc index and page index)
usually this method gets called prior to a call to ClipboardGetImageData
(to determine the right size of the image buffer to be allocated)
```

INPUTS

```
pApiData -- pointer to the internal data struct
iDocIndex -- index of the document containing the selected page
iPageIndex -- index of the selected page
strInfoType -- the selected information type you want to retrieve
strPageIdentifier -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_PageID -- max. text length of the previous wchar_t buffer
iWidth -- pointer to a long variable which gets filled by this function
iHeight -- pointer to a long variable which gets filled by this function
iFileSize -- pointer to a long variable which gets filled by this function
iResolution -- pointer to a long variable which gets filled by this function
strColormode --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Colormode -- max. text length of the previous wchar_t buffer
strSource --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Source -- max. text length of the previous wchar_t buffer
strFileExtension --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_FileExtension -- max. text length of the previous wchar_t buffer
iPageRotation -- pointer to a long variable which gets filled by this function
```

OUTPUT

```
strPageIdentifier -- page identifier of the selected page
iWidth (the width of the selected image in pixel)
iHeight (the height of the selected image in pixel)
iFileSize (the filesize of the selected image)
iResolution (the resolution of the selected image in dpi)
strColormode (the colormode of the selected image)
strSource (the source where the selected page came from)
strFileExtension (the file extension of the selected image)
iPageRotation (the rotation of the page in degrees, 0 to 359 degrees, no negative values)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

NOTES

```
strInfoType can be one of the following
   * FULL
   * PREVIEW2
   * PREVIEW4
   * PREVIEW8
```

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardClear ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardPastePage

27. /ClipboardGetUsedDiskSpace [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardGetUsedDiskSpace   (void* pApiData,
                                              long* iDiskSpace)
```

FUNCTION

calculates the disk space used by the clipboard

INPUTS

```
pApiData -- pointer to the internal data struct
iDiskSpace -- pointer to a long variable which gets filled by this function
```

OUTPUT

iDiskSpace (used disk space measured in MB)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardClear ClipboardPasteDocument ClipboardPastePage

28. /ClipboardPasteDocument [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardPasteDocument      (void* pApiData,
                                              const wchar_t* strDocIdentifier, wchar_t* strNewDocIdentifier,
                                              long iBufferSize_NewDocID)
```

FUNCTION

copies a document from the clipboard to the archive
(and creating a new document identifier in the archive)

INPUTS

```
pApiData               -- pointer to the internal data struct
strDocIdentifier       -- the document identifier you want to restore from the clipboard
strNewDocIdentifier    -- pointer to a wchar_t buffer which gets filled by this function
iBufferSize_NewDocID   -- max. text length of the previous wchar_t buffer
```

OUTPUT

strNewDocIdentifier (the new document identifier in the archive)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardClear ClipboardPastePage

29. /ClipboardPastePage [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ClipboardPastePage  (void* pApiData,
                                      const wchar_t* strPageIdentifier,
                                      wchar_t* strNewDocIdentifier, long iBufferSize_NewDocID)
```

FUNCTION

copies a page from the clipboard to the archive
(and creating a new document identifier for a single page document in the archive)

INPUTS

```
pApiData               -- pointer to the internal data struct
strPageIdentifier      -- the page identifier of the page you want to restore from the clipboard
strNewDocIdentifier    -- pointer to a wchar_t buffer which gets filled by this function
iBufferSize_NewDocID   -- max. text length of the previous wchar_t buffer
```

OUTPUT

```
strNewDocIdentifier (the new document identifier in the archive)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

ClipboardClearDoc ClipboardCopyDocToClipboard ClipboardCopyPageToClipboard ClipboardGetDocCount
ClipboardGetDocInfo ClipboardGetInformation ClipboardGetImageData ClipboardGetPageInfoByIdentifier
ClipboardGetPageInfoByIndex ClipboardGetUsedDiskSpace ClipboardPasteDocument ClipboardClear

30. /DocumentGetCount [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long DocumentGetCount    (void* pApiData,
                                      long* iDocCount)
```

FUNCTION

```
get the document count of the current archive
```

INPUTS

```
pApiData               -- pointer to the internal data struct
iDocCount              -- pointer to a long variable which gets filled by the function
```

OUTPUT

```
iDocCount (number of documents in the current archive)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

DocumentGetDetails DocumentGetInfoByIdentifier DocumentGetInfoByIndex DocumentQueryExistence
DocumentSetDetails DocumentSetPageFilter WorkspaceSetDocFilter

31. /DocumentGetDetails [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long DocumentGetDetails  (void* pApiData,
                                      const wchar_t* strDocIdentifier,
                                      long iDetailIndex,
                                      wchar_t* strDetailInformation, long iMaxBufferSize_DetailInformation)
```

FUNCTION

```
get additional metadata information for the selected document
```

INPUTS

```
pApiData                          -- pointer to the internal data struct
strDocIdentifier                  -- document identifier of the selected document
iDetailIndex                      -- index of the detail information (0-based)
strDetailInformation              -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_DetailInformation  -- max. text length of the previous wchar_t buffer
```

OUTPUT

strDetailInformation (detail information selected by document identifier and detail index)

NOTES

this function is not used in version 1.0 of SceyeSphere

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

DocumentGetCount DocumentGetInfoByIdentifier DocumentGetInfoByIndex DocumentQueryExistence
DocumentSetDetails DocumentSetPageFilter WorkspaceSetDocFilter

32. /DocumentGetInfoByIdentifier [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long DocumentGetInfoByIdentifier (void* pApiData,
                                              const wchar_t* strDocID,
                                              wchar_t* strDocType, long iMaxBufferSize_Type,
                                              wchar_t* strDocName, long iMaxBufferSize_Name,
                                              wchar_t* strDocCreatedTimestamp, long iMaxBufferSize_Created,
                                              wchar_t* strDocModifiedTimestamp, long iMaxBufferSize_Modified,
                                              long* iPageCount,
                                              bool* bReadOnly)
```

FUNCTION

get additional metadata information for the selected document

INPUTS

pApiData -- pointer to the internal data struct
strDocID -- identifier of the selected document
strDocType -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Type -- max. text length of the previous wchar_t buffer
strDocName -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Name -- max. text length of the previous wchar_t buffer
strDocCreatedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Created -- max. text length of the previous wchar_t buffer
strDocModifiedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Modified -- max. text length of the previous wchar_t buffer
iPageCount -- pointer to a long variable which gets filled during method execution
bReadOnly -- pointer to a bool variable which gets filled during method execution

OUTPUT

strDocType (type of the selected document)
strDocName (name of the selected document)
strDocCreatedTimestamp (selected document creation timestamp)
strDocModifiedTimestamp (selected document modification timestamp)
iPageCount (selected document page count)
bReadOnly (indicates if document is readonly)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

DocumentGetDetails DocumentGetCount DocumentGetInfoByIndex DocumentQueryExistence
DocumentSetDetails DocumentSetPageFilter WorkspaceSetDocFilter

33. /DocumentGetInfoByIndex [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long DocumentGetInfoByIndex  (void* pApiData,
                                          long iDocIndex,
                                          wchar_t* strDocID, long iMaxBufferSize_DocID,
```

```
                                        wchar_t* strDocType, long iMaxBufferSize_Type,
                                        wchar_t* strDocName, long iMaxBufferSize_Name,
                                        wchar_t* strDocCreatedTimestamp, long iMaxBufferSize_Created,
                                        wchar_t* strDocModifiedTimestamp, long iMaxBufferSize_Modified,
                                        long* iPageCount,
                                        bool* bReadOnly)
```

FUNCTION

get additional metadata information for the selected document

INPUTS

pApiData -- pointer to the internal data struct
iDocIndex -- index of the selected document
strDocID -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_DocID -- max. text length of the previous wchar_t buffer
strDocType -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Type -- max. text length of the previous wchar_t buffer
strDocName -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Name -- max. text length of the previous wchar_t buffer
strDocCreatedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Created -- max. text length of the previous wchar_t buffer
strDocModifiedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Modified -- max. text length of the previous wchar_t buffer
iPageCount -- pointer to a long variable which gets filled during method execution

OUTPUT

strDocID (identifier of the selected document)
strDocType (type of the selected document)
strDocName (name of the selected document)
strDocCreatedTimestamp (selected document creation timestamp)
strDocModifiedTimestamp (selected document modification timestamp)
iPageCount (selected document page count)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

DocumentGetDetails DocumentGetInfoByIdentifier DocumentGetCount DocumentQueryExistence
DocumentSetDetails DocumentSetPageFilter WorkspaceSetDocFilter

34. /DocumentQueryExistence [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long DocumentQueryExistence  (void* pApiData,
                                          const wchar_t* strDocumentIdentifier,
                                          bool* bDocumentExists)
```

FUNCTION

checks if a document (selected by document identifier) exists

INPUTS

pApiData -- pointer to the internal data struct
strDocumentIdentifier -- identifier of the selected document
bDocumentExists -- pointer to bool variable which gets filled during method execution

OUTPUT

bDocumentExists (indicates if the selected document exists)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

DocumentGetDetails DocumentGetInfoByIdentifier DocumentGetInfoByIndex DocumentGetCount
DocumentSetDetails DocumentSetPageFilter WorkspaceSetDocFilter

35. /DocumentSetDetails [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long DocumentSetDetails  (void* pApiData,
                                      const wchar_t* strDocIdentifier,
                                      long iDetailIndex,
                                      const wchar_t* strDetailInformation)
```

FUNCTION

sets additional detail information for selected document

INPUTS

pApiData -- pointer to the internal data struct
strDocIdentifier -- identifier of the selected document
iDetailIndex -- index of the detail information
strDetailInformation -- the detail information you want to set for the selected document

NOTES

this function is not used in version 1.0 of SceyeSphere

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

DocumentGetDetails DocumentGetInfoByIdentifier DocumentGetInfoByIndex DocumentQueryExistence
DocumentGetCount DocumentSetPageFilter WorkspaceSetDocFilter

36. /DocumentSetPageFilter [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long DocumentSetPageFilter   (void* pApiData,
                                          const wchar_t* strPageFilterString)
```

FUNCTION

activates a filter to limit the result of the page queries

INPUTS

pApiData -- pointer to the internal data struct
strPageFilterString -- the filter string which gets used for filtering the results

NOTES

this function is not used in version 1.0 of SceyeSphere

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

DocumentGetDetails DocumentGetInfoByIdentifier DocumentGetInfoByIndex DocumentQueryExistence
DocumentSetDetails DocumentGetCount WorkspaceSetDocFilter

37. /GlobalSettingRead [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long GlobalSettingRead   (void* pApiData,
                                      const wchar_t* strSettingsPath,
```

```
                                        const wchar_t* strDefault,
                                        wchar_t* strSettingValue,
                                        long iMaxBufferSize_Value)
```

FUNCTION

retrieve a global settings value

INPUTS

pApiData -- pointer to the internal data struct
strSettingsPath -- contains the path (in the settings hierarchy) to the selected setting value
strDefault -- default value which is used when we don't find the specified setting
strSettingValue -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Value -- max. text length of the previous wchar_t buffer

OUTPUT

strSettingValue (the selected setting value or the default value if the setting is not found)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

NOTES

strSettingsPath look like this: "SceyeSphereToolboxTesterNet|GlobalStartCount"

SEE ALSO

GlobalSettingWrite

38. /GlobalSettingWrite [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long GlobalSettingWrite  (void* pApiData,
                                      const wchar_t* strSettingsPath,
                                      const wchar_t* strSettingValue)
```

FUNCTION

write a global settings value

INPUTS

pApiData -- pointer to the internal data struct
strSettingsPath -- contains the path (in the settings hierarchy) to the selected setting value
strSettingValue -- the value that will be set

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

NOTES

strSettingsPath look like this: "SceyeSphereToolboxTesterNet|GlobalStartCount"

SEE ALSO

GlobalSettingRead

39. /ImageAddData [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ImageAddData    (void* pApiData,
                                  const wchar_t* strImageIdentifier,
                                  const wchar_t* strImageType,
                                  const wchar_t* strColormode,
                                  const wchar_t* strExtension,
```

```
                                    unsigned char* pImageData, long iImageDataSize,
                                    long iWidth,
                                    long iHeight,
                                    long iResolution,
                                    long iPageRotation)
```

FUNCTION

adds the binary image data along with its metadata to the archive

INPUTS

```
pApiData -- pointer to the internal data struct
strImageIdentifier -- the page identifier of the selected page
strImageType -- the image type of the selected page
strColormode -- the colormode of the selected image
strExtension -- the file extension of the selected image
pImageData -- pointer to the image data that will be written
iImageDataSize -- size of the previous image data buffer
iWidth -- width of the image (in pixel)
iHeight -- height of the image (in pixel)
iResolution -- resolution of the image (in dpi)
iPageRotation -- rotation angle of the page (angle from 0 to 359, no negative values)
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

NOTES

```
strImageType can be one of the following
   * FULL
   * PREVIEW2
   * PREVIEW4
   * PREVIEW8
the pImageData buffer holds the binary data of the image file
(the data is a complete file with file header, etc.)
```

SEE ALSO

ImageGetData

40. /ImageGetData [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long ImageGetData    (void* pApiData,
                                  const wchar_t* strImageIdentifier,
                                  const wchar_t* strImageType,
                                  unsigned char* pImageDataBuffer,
                                  unsigned long iMaxSize_ImageDataBuffer)
```

FUNCTION

retrieves the binary image data along with its metadata from the archive

INPUTS

```
pApiData -- pointer to the internal data struct
strImageIdentifier -- the page identifier of the selected page
strImageType -- the image type of the selected page
pImageDataBuffer -- pointer to a image data buffer which gets filled during method execution
iMaxSize_ImageDataBuffer -- maximum size of the previous image data buffer
```

OUTPUT

pImageDataBuffer (binary image data of the selected image will be copied into this buffer)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

NOTES

```
strImageType can be one of the following
   * FULL
```

```
   * PREVIEW2
   * PREVIEW4
   * PREVIEW8
the pImageData buffer holds the binary data of the image file (the data is a complete file with file header, etc.)
```

SEE ALSO

ImageAddData

41. /PageGetDetails [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long PageGetDetails  (void* pApiData,
                                  const wchar_t* strPageIdentifier,
                                  long iDetailIndex,
                                  wchar_t* strDetailInformation, long iMaxBufferSize_DetailInformation)
```

FUNCTION

retrieves the detail information of the selected page

INPUTS

```
pApiData -- pointer to the internal data struct
strPageIdentifier -- the page identifier of the selected page
iDetailIndex -- the index of the requested detail information
strDetailInformation -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_DetailInformation -- max. text length of the previous wchar_t buffer
```

OUTPUT

strDetailInformation (detail information selected by page identifier and detail index)

NOTES

this function is not used in version 1.0 of SceyeSphere

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

PageGetInfoByDocIdAndIndex PageGetInfoByIdentifier PageGetInfoByIndex PageQueryExistence
PageSetDetails PageSetRotation DocumentSetPageFilter

42. /PageGetInfoByDocIdAndIndex [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long PageGetInfoByDocIdAndIndex  (void* pApiData,
                                              const wchar_t* strDocIdentifier,
                                              long iPageIndex,
                                              const wchar_t* strInfoType,
                                              wchar_t* strPageIdentifier, long iMaxBufferSize_PageID,
                                              long* iWidth,
                                              long* iHeight,
                                              long* iFileSize,
                                              long* iResolution,
                                              wchar_t* strColormode, long iMaxBufferSize_Colormode,
                                              wchar_t* strSource, long iMaxBufferSize_Source,
                                              wchar_t* strFileExtension, long iMaxBufferSize_FileExtension,
                                              long* iPageRotation)
```

FUNCTION

```
retrieves the page metadata (page selected by document identifier and page index)
usually this method gets called prior to a call to ImageGetData
(to determine the right size of the image buffer to be allocated)
```

INPUTS

```
pApiData -- pointer to the internal data struct
strDocIdentifier -- the identifier of the document containing the selected page
iPageIndex -- the index of the selected page
strInfoType -- the selected information type you want to retrieve (TODO)
iWidth -- pointer to a long variable which gets filled by this function
iHeight -- pointer to a long variable which gets filled by this function
iFileSize -- pointer to a long variable which gets filled by this function
iResolution -- pointer to a long variable which gets filled by this function
strColormode --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Colormode -- max. text length of the previous wchar_t buffer
strSource --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Source -- max. text length of the previous wchar_t buffer
strFileExtension --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_FileExtension -- max. text length of the previous wchar_t buffer
iPageRotation -- pointer to a long variable which gets filled by this function
```

OUTPUT

```
iWidth (the width of the selected image in pixel)
iHeight (the height of the selected image in pixel)
iFileSize (the filesize of the selected image)
iResolution (the resolution of the selected image in dpi)
strColormode (the colormode of the selected image)
strSource (the source where the selected page came from)
strFileExtension (the file extension of the selected image)
iPageRotation (the rotation of the page in degrees, 0 to 359 degrees, no negative values)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

NOTES

```
strInfoType can be one of the following
   * FULL
   * PREVIEW2
   * PREVIEW4
   * PREVIEW8
```

SEE ALSO

PageGetDetails PageGetInfoByIdentifier PageGetInfoByIndex PageQueryExistence
PageSetDetails PageSetRotation DocumentSetPageFilter

43. /PageGetInfoByIdentifier [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long PageGetInfoByIdentifier (void* pApiData,
                                          const wchar_t* strPageIdentifier,
                                          const wchar_t* strInfoType,
                                          long* iWidth,
                                          long* iHeight,
                                          long* iFileSize,
                                          long* iResolution,
                                          wchar_t* strColormode, long iMaxBufferSize_Colormode,
                                          wchar_t* strSource, long iMaxBufferSize_Source,
                                          wchar_t* strFileExtension, long iMaxBufferSize_FileExtension,
                                          long* iPageRotation)
```

FUNCTION

```
retrieves the page metadata (page selected by page identifier)
usually this method gets called prior to a call to ImageGetData
(to determine the right size of the image buffer to be allocated)
```

INPUTS

```
pApiData -- pointer to the internal data struct
strPageIdentifier -- the identifier of the selected page
strInfoType -- the selected information type you want to retrieve (TODO)
iWidth -- pointer to a long variable which gets filled by this function
iHeight -- pointer to a long variable which gets filled by this function
iFileSize -- pointer to a long variable which gets filled by this function
iResolution -- pointer to a long variable which gets filled by this function
strColormode --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Colormode -- max. text length of the previous wchar_t buffer
strSource --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Source -- max. text length of the previous wchar_t buffer
strFileExtension --  pointer to a wchar_t buffer which gets filled by this function
```

```
iMaxBufferSize_FileExtension -- max. text length of the previous wchar_t buffer
iPageRotation -- pointer to a long variable which gets filled by this function
```

OUTPUT

```
iWidth (the width of the selected image in pixel)
iHeight (the height of the selected image in pixel)
iFileSize (the filesize of the selected image)
iResolution (the resolution of the selected image in dpi)
strColormode (the colormode of the selected image)
strSource (the source where the selected page came from)
strFileExtension (the file extension of the selected image)
iPageRotation (the rotation of the page in degrees, 0 to 359 degrees, no negative values)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

NOTES

```
strInfoType can be one of the following
   * FULL
   * PREVIEW2
   * PREVIEW4
   * PREVIEW8
```

SEE ALSO

PageGetInfoByDocIdAndIndex PageGetDetails PageGetInfoByIndex PageQueryExistence
PageSetDetails PageSetRotation DocumentSetPageFilter

44. /PageGetInfoByIndex [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long PageGetInfoByIndex  (void* pApiData,
                                      long iDocIndex,
                                      long iPageIndex,
                                      const wchar_t* strInfoType,
                                      wchar_t* strPageIdentifier, long iMaxBufferSize_PageID,
                                      long* iWidth,
                                      long* iHeight,
                                      long* iFileSize,
                                      long* iResolution,
                                      wchar_t* strColormode, long iMaxBufferSize_Colormode,
                                      wchar_t* strSource, long iMaxBufferSize_Source,
                                      wchar_t* strFileExtension, long iMaxBufferSize_FileExtension,
                                      long* iPageRotation)
```

FUNCTION

```
retrieves the page metadata (page selected by page identifier)
usually this method gets called prior to a call to ImageGetData
(to determine the right size of the image buffer to be allocated)
```

INPUTS

```
pApiData -- pointer to the internal data struct
iDocIndex -- index of the document containing the selected page
iPageIndex -- index of the selected page
strInfoType -- the selected information type you want to retrieve
strPageIdentifier -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_PageID -- max. text length of the previous wchar_t buffer
iWidth -- pointer to a long variable which gets filled by this function
iHeight -- pointer to a long variable which gets filled by this function
iFileSize -- pointer to a long variable which gets filled by this function
iResolution -- pointer to a long variable which gets filled by this function
strColormode --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Colormode -- max. text length of the previous wchar_t buffer
strSource --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Source -- max. text length of the previous wchar_t buffer
strFileExtension --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_FileExtension -- max. text length of the previous wchar_t buffer
iPageRotation -- pointer to a long variable which gets filled by this function
```

OUTPUT

```
strPageIdentifier -- the identifier of the selected page
iWidth (the width of the selected image in pixel)
iHeight (the height of the selected image in pixel)
```

```
iFileSize (the filesize of the selected image)
iResolution (the resolution of the selected image in dpi)
strColormode (the colormode of the selected image)
strSource (the source where the selected page came from)
strFileExtension (the file extension of the selected image)
iPageRotation (the rotation of the page in degrees, 0 to 359 degrees, no negative values)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

NOTES

```
strInfoType can be one of the following
   * FULL
   * PREVIEW2
   * PREVIEW4
   * PREVIEW8
```

SEE ALSO

PageGetInfoByDocIdAndIndex PageGetInfoByIdentifier PageGetDetails PageQueryExistence
PageSetDetails PageSetRotation DocumentSetPageFilter

45. /PageQueryExistence [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long PageQueryExistence  (void* pApiData,
                                      const wchar_t* strPageIdentifier,
                                      bool* bPageExists)
```

FUNCTION

```
check if a page exists (page selected by page identifier)
```

INPUTS

```
pApiData -- pointer to the internal data struct
strPageIdentifier -- page identifier of the selected page
bPageExists -- pointer to a bool variable which gets filled during method execution
```

OUTPUT

```
bPageExists (flag that indicates if a page exists)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

PageGetInfoByDocIdAndIndex PageGetInfoByIdentifier PageGetInfoByIndex PageGetDetails
PageSetDetails PageSetRotation DocumentSetPageFilter

46. /PageSetDetails [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long PageSetDetails  (void* pApiData,
                                  const wchar_t* strPageIdentifier,
                                  long iDetailIndex,
                                  const wchar_t* strDetailInformation)
```

FUNCTION

```
sets additional detail information of the selected page
```

INPUTS

```
pApiData -- pointer to the internal data struct
strPageIdentifier -- the page identifier of the selected page
```

iDetailIndex -- the index of the requested detail information
strDetailInformation -- detail information for the selected page

NOTES

this function is not used in version 1.0 of SceyeSphere

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

PageGetInfoByDocIdAndIndex PageGetInfoByIdentifier PageGetInfoByIndex PageQueryExistence
PageGetDetails PageSetRotation DocumentSetPageFilter

47. /PageSetRotation [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long PageSetRotation (void* pApiData,
                                  const wchar_t* strPageIdentifier,
                                  long iPageRotation)
```

FUNCTION

fast way to directly set page rotation angle

INPUTS

pApiData -- pointer to the internal data struct
strPageIdentifier -- the page identifier of the selected page
iPageRotation -- the rotation angle of the selected page (0 - 359 degrees, no negative values)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

PageGetInfoByDocIdAndIndex PageGetInfoByIdentifier PageGetInfoByIndex PageQueryExistence
PageSetDetails PageGetDetails DocumentSetPageFilter

48. /TrashboxClear [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxClear (void* pApiData)
```

FUNCTION

clears the internal trashbox (remove all trashbox content)

INPUTS

pApiData -- pointer to the internal data struct

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

49. /TrashboxClearDoc [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxClearDoc    (void* pApiData,
                                      const wchar_t* strDocumentIdentifier)
```

FUNCTION

clears a document in the internal trashbox (remove a specific document from the trashbox)

INPUTS

pApiData -- pointer to the internal data struct
strDocumentIdentifier -- the identifier of the document that will be removed

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

TrashboxClear TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

50. /TrashboxCopyDocToTrash [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxCopyDocToTrash  (void* pApiData,
                                          const wchar_t* strDocIdentifier)
```

FUNCTION

copies a document (specified by identifier) from the current archive to the trashbox

INPUTS

pApiData -- pointer to the internal data struct
strDocIdentifier -- the identifier of the document that will be copied

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

TrashboxClearDoc TrashboxClear TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

51. /TrashboxCopyPageToTrash [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxCopyPageToTrash (void* pApiData,
                                          const wchar_t* strPageIdentifier)
```

FUNCTION

copies a page (specified by identifier) from the current archive to the trashbox

INPUTS

pApiData -- pointer to the internal data struct
strPageIdentifier -- the identifier of the page that will be copied

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxClear TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

52. /TrashboxGetDocCount [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxGetDocCount (void* pApiData,
                                      long* iDocCount)
```

FUNCTION

retrieves the document count from the trashbox

INPUTS

pApiData -- pointer to the internal data struct
iDocCount -- pointer to a long variable which will be filled during execution of this method

OUTPUT

iDocCount (number of documents currently stored in trashbox)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxClear
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

53. /TrashboxGetDocInfo [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxGetDocInfo  (void* pApiData,
                                      long iDocIndex,
                                      wchar_t* strDocID, long iMaxBufferSize_DocID,
                                      wchar_t* strDocType, long iMaxBufferSize_Type,
                                      wchar_t* strDocName, long iMaxBufferSize_Name,
                                      wchar_t* strDocCreatedTimestamp, long iMaxBufferSize_Created,
                                      wchar_t* strDocModifiedTimestamp, long iMaxBufferSize_Modified,
                                      long* iPageCount)
```

FUNCTION

retrieves the document metadata from the trashbox

INPUTS

pApiData -- pointer to the internal data struct
iDocIndex -- index of the selected document
strDocID -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_DocID -- max. text length of the previous wchar_t buffer
strDocType -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Type -- max. text length of the previous wchar_t buffer
strDocName -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Name -- max. text length of the previous wchar_t buffer
strDocCreatedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Created -- max. text length of the previous wchar_t buffer
strDocModifiedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Modified -- max. text length of the previous wchar_t buffer
iPageCount -- pointer to a long variable which gets filled during method execution

OUTPUT

```
strDocID (identifier of the selected document)
strDocType (type of the selected document)
strDocName (name of the selected document)
strDocCreatedTimestamp (selected document creation timestamp)
strDocModifiedTimestamp (selected document modification timestamp)
iPageCount (selected document page count)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxClear TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

54. /TrashboxGetImageData [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxGetImageData    (void* pApiData,
                                          const wchar_t* strImageIdentifier,
                                          const wchar_t*  strImageType,
                                          unsigned char* pImageDataBuffer, long iMaxSize_ImageDataBuffer)
```

FUNCTION

```
retrieves the binary image data from the trashbox
```

INPUTS

```
pApiData -- pointer to the internal data struct
strImageIdentifier -- identifier of the selected image
strImageType -- type of the selected image TODO
pImageDataBuffer -- pointer to a data buffer which gets filled with the binary image data
iMaxSize_ImageDataBuffer -- maximum size of the previous data buffer
```

OUTPUT

```
iImageDataBuffer (holds the binary image data)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxClear TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

55. /TrashboxGetInformation [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxGetInformation  (void* pApiData,
                                          wchar_t* strVersion, long iMaxBuffer_Version,
                                          wchar_t* strCreatedTimestamp, long iMaxBuffer_Created,
                                          wchar_t* strModifiedTimestamp, long iMaxBuffer_Modified,
                                          long* iDocCount)
```

FUNCTION

```
retrieves the metadata information of the trashbox
```

INPUTS

```
pApiData -- pointer to the internal data struct
strVersion -- pointer to a wchar_t buffer which gets filled by this function
iMaxBuffer_Version -- max. text length of the previous wchar_t buffer
strCreatedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBuffer_Created -- max. text length of the previous wchar_t buffer
```

```
strModifiedTimestamp -- pointer to a wchar_t buffer which gets filled by this function
iMaxBuffer_Modified -- max. text length of the previous wchar_t buffer
iDocCount -- pointer to a long variable which gets filled by this function
```

OUTPUT

```
strVersion (version number of the trashbox (currently "1")
strCreatedTimestamp (trashbox creation timestamp)
strModifiedTimestamp (trashbox modification timestamp)
iDocCount (count of documents currently stored in trashbox)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxClear TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

56. /TrashboxGetPageInfoByIdentifier [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxGetPageInfoByIdentifier (void* pApiData,
                                                  const wchar_t* strPageIdentifier,
                                                  const wchar_t* strInfoType,
                                                  long* iWidth,
                                                  long* iHeight,
                                                  long* iFileSize,
                                                  long* iResolution,
                                                  wchar_t* strColormode, long iMaxBufferSize_Colormode,
                                                  wchar_t* strSource, long iMaxBufferSize_Source,
                                                  wchar_t* strFileExtension, long iMaxBufferSize_FileExtension,
                                                  long* iPageRotation)
```

FUNCTION

```
retrieves the page metadata information of the trashbox
usually this method gets called prior to a call to TrashboxGetImageData
(to determine the right size of the image buffer to be allocated)
```

INPUTS

```
pApiData -- pointer to the internal data struct
strPageIdentifier -- the identifier of the selected page
strInfoType -- the selected information type you want to retrieve
iWidth -- pointer to a long variable which gets filled by this function
iHeight -- pointer to a long variable which gets filled by this function
iFileSize -- pointer to a long variable which gets filled by this function
iResolution -- pointer to a long variable which gets filled by this function
strColormode --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Colormode -- max. text length of the previous wchar_t buffer
strSource --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Source -- max. text length of the previous wchar_t buffer
strFileExtension --  pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_FileExtension -- max. text length of the previous wchar_t buffer
iPageRotation -- pointer to a long variable which gets filled by this function
```

OUTPUT

```
iWidth (the width of the selected image in pixel)
iHeight (the height of the selected image in pixel)
iFileSize (the filesize of the selected image)
iResolution (the resolution of the selected image in dpi)
strColormode (the colormode of the selected image)
strSource (the source where the selected page came from)
strFileExtension (the file extension of the selected image)
iPageRotation (the rotation of the page in degrees, 0 to 359 degrees, no negative values)
```

RETURN VALUE

```
ErrorCode ( >=0 => success, < 0 => failed)
```

NOTES

```
strInfoType can be one of the following
   * FULL
```

* PREVIEW2
* PREVIEW4
* PREVIEW8

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxClear
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

57. /TrashboxGetPageInfoByIndex [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxGetPageInfoByIndex  (void* pApiData,
                                              long iDocIndex,
                                              long iPageIndex,
                                              const wchar_t* strInfoType,
                                              wchar_t* strPageIdentifier, long iMaxBufferSize_PageID,
                                              long* iWidth,
                                              long* iHeight,
                                              long* iFileSize,
                                              long* iResolution,
                                              wchar_t* strColormode, long iMaxBufferSize_Colormode,
                                              wchar_t* strSource, long iMaxBufferSize_Source,
                                              wchar_t* strFileExtension, long iMaxBufferSize_FileExtension,
                                              long* iPageRotation)
```

FUNCTION

retrieves the page metadata information of the trashbox (page selected by doc index and page index)
usually this method gets called prior to a call to TrashboxGetImageData
(to determine the right size of the image buffer to be allocated)

INPUTS

pApiData -- pointer to the internal data struct
iDocIndex -- index of the document containing the selected page
iPageIndex -- index of the selected page
strInfoType -- the selected information type you want to retrieve
strPageIdentifier -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_PageID -- max. text length of the previous wchar_t buffer
iWidth -- pointer to a long variable which gets filled by this function
iHeight -- pointer to a long variable which gets filled by this function
iFileSize -- pointer to a long variable which gets filled by this function
iResolution -- pointer to a long variable which gets filled by this function
strColormode -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Colormode -- max. text length of the previous wchar_t buffer
strSource -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Source -- max. text length of the previous wchar_t buffer
strFileExtension -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_FileExtension -- max. text length of the previous wchar_t buffer
iPageRotation -- pointer to a long variable which gets filled by this function

OUTPUT

strPageIdentifier -- page identifier of the selected page
iWidth (the width of the selected image in pixel)
iHeight (the height of the selected image in pixel)
iFileSize (the filesize of the selected image)
iResolution (the resolution of the selected image in dpi)
strColormode (the colormode of the selected image)
strSource (the source where the selected page came from)
strFileExtension (the file extension of the selected image)
iPageRotation (the rotation of the page in degrees, 0 to 359 degrees, no negative values)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

NOTES

strInfoType can be one of the following
* FULL
* PREVIEW2
* PREVIEW4
* PREVIEW8

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxClear TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxRestorePage

58. /TrashboxGetUsedDiskSpace [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxGetUsedDiskSpace    (void* pApiData,
                                               long* iDiskSpace)
```

FUNCTION

calculates the disk space used by the trashbox

INPUTS

pApiData -- pointer to the internal data struct
iDiskSpace -- pointer to a long variable which gets filled by this function

OUTPUT

iDiskSpace (used disk space measured in MB)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxClear TrashboxRestoreDocument TrashboxRestorePage

59. /TrashboxRestoreDocument [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxRestoreDocument (void* pApiData,
                                          const wchar_t* strDocIdentifier,
                                          wchar_t* strNewDocIdentifier, long iBufferSize_NewDocID)
```

FUNCTION

copies a document from the trashbox to the archive (and creating a new document identifier in the archive)

INPUTS

```
pApiData               -- pointer to the internal data struct
strDocIdentifier       -- the document identifier you want to restore from the trashbox
strNewDocIdentifier    -- pointer to a wchar_t buffer which gets filled by this function
iBufferSize_NewDocID   -- max. text length of the previous wchar_t buffer
```

OUTPUT

strNewDocIdentifier (the new document identifier in the archive)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxClear TrashboxRestorePage

60. /TrashboxRestorePage [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long TrashboxRestorePage (void* pApiData,
                                      const wchar_t* strPageIdentifier,
                                      wchar_t* strNewDocIdentifier,
                                      long iBufferSize_NewDocID)
```

FUNCTION

copies a page from the trashbox to the archive
(and creating a new document identifier for a single page document in the archive)

INPUTS

```
pApiData               -- pointer to the internal data struct
strPageIdentifier      -- the page identifier of the page you want to restore from the trashbox
strNewDocIdentifier    -- pointer to a wchar_t buffer which gets filled by this function
iBufferSize_NewDocID   -- max. text length of the previous wchar_t buffer
```

OUTPUT

strNewDocIdentifier (the new document identifier in the archive)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

TrashboxClearDoc TrashboxCopyDocToTrash TrashboxCopyPageToTrash TrashboxGetDocCount
TrashboxGetDocInfo TrashboxGetImageData TrashboxGetInformation TrashboxGetPageInfoByIdentifier
TrashboxGetPageInfoByIndex TrashboxGetUsedDiskSpace TrashboxRestoreDocument TrashboxClear

61. /WorkspaceGetDetails [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long WorkspaceGetDetails (void* pApiData,
                                      long iDetailIndex,
                                      wchar_t* strDetailInformation,
                                      long iMaxBufferSize_DetailInformation)
```

FUNCTION

get additional metadata information of the workspace

INPUTS

```
pApiData                          -- pointer to the internal data struct
iDetailIndex                     -- index of the detail information (0-based)
strDetailInformation              -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_DetailInformation  -- max. text length of the previous wchar_t buffer
```

OUTPUT

strDetailInformation (additional detail information of the workspace)

NOTES

this function is not used in version 1.0 of SceyeSphere

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

WorkspaceGetInfo WorkspaceSetDetails WorkspaceSetDocFilter WorkspaceSetFilter
WorkspaceSettingRead WorkspaceSettingWrite

62. /WorkspaceGetInfo [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long WorkspaceGetInfo    (void* pApiData,
                                      wchar_t* strName, long iMaxBufferSize_Name,
                                      wchar_t* strVersion, long iMaxBufferSize_Version,
                                      wchar_t* strCreated, long iMaxBufferSize_Created,
                                      wchar_t* strModified, long iMaxBufferSize_Modified,
                                      bool* bReadOnly,
                                      long* iDocCount)
```

FUNCTION

get additional metadata information of the workspace

INPUTS

pApiData -- pointer to the internal data struct
strName -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Name -- max. text length of the previous wchar_t buffer
strVersion -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Version -- max. text length of the previous wchar_t buffer
strCreated -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Created -- max. text length of the previous wchar_t buffer
strModified -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Modified -- max. text length of the previous wchar_t buffer
bReadOnly -- pointer to a bool variable which gets filled during method execution
iDocCount -- pointer to a long variable which gets filled during method execution

OUTPUT

strName (name of the archive)
strCreated (archive creation timestamp)
strModified (archive modification timestamp)
bReadOnly (indicates if archive is readonly)
iDocCount (number of documents stored in archive)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

WorkspaceGetDetails WorkspaceSetDetails WorkspaceSetDocFilter WorkspaceSetFilter
WorkspaceSettingRead WorkspaceSettingWrite

63. /WorkspaceSetDetails [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long WorkspaceSetDetails (void* pApiData,
                                      long iDetailIndex,
                                      const wchar_t* strDetailInformation)
```

FUNCTION

sets additional detail information for current archive

INPUTS

pApiData -- pointer to the internal data struct
iDetailIndex -- index of the detail information
strDetailInformation -- the detail information you want to set for the current archive

NOTES

this function is not used in version 1.0 of SceyeSphere

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

WorkspaceGetInfo WorkspaceGetDetails WorkspaceSetDocFilter WorkspaceSetFilter
WorkspaceSettingRead WorkspaceSettingWrite

64. /WorkspaceSetDocFilter [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long WorkspaceSetDocFilter   (void* pApiData,
                                          const wchar_t* strDocumentFilterString)
```

FUNCTION

activates a filter to limit the result of the document queries

INPUTS

pApiData -- pointer to the internal data struct
strDocumentFilterString -- the filter string which gets used for filtering the results

NOTES

this function is not used in version 1.0 of SceyeSphere

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

WorkspaceGetInfo WorkspaceSetDetails WorkspaceGetDetails WorkspaceSetFilter
WorkspaceSettingRead WorkspaceSettingWrite
DocumentGetDetails DocumentGetInfoByIdentifier DocumentGetInfoByIndex DocumentQueryExistence
DocumentSetDetails DocumentSetPageFilter DocumentGetCount

65. /WorkspaceSetFilter [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long WorkspaceSetFilter  (void* pApiData,
                                      const wchar_t* strWorkspaceFilterString)
```

FUNCTION

activates a filter to limit the result of the archive queries

INPUTS

pApiData -- pointer to the internal data struct
strWorkspaceFilterString -- the filter string which gets used for filtering the results

NOTES

this function is not used in version 1.0 of SceyeSphere

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

WorkspaceGetInfo WorkspaceSetDetails WorkspaceSetDocFilter WorkspaceGetDetails
WorkspaceSettingRead WorkspaceSettingWrite

66. /WorkspaceSettingRead [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long WorkspaceSettingRead    (void* pApiData,
                                          const wchar_t* strSettingsPath,
                                          const wchar_t* strDefault,
                                          wchar_t* strSettingValue, long iMaxBufferSize_Value)
```

FUNCTION

retrieve a archive settings value

INPUTS

```
pApiData -- pointer to the internal data struct
strSettingsPath -- contains the path (in the settings hierarchy) to the selected setting value
strDefault -- default value which is used when we don't find the specified setting
strSettingValue -- pointer to a wchar_t buffer which gets filled by this function
iMaxBufferSize_Value -- max. text length of the previous wchar_t buffer
```

OUTPUT

strSettingValue (the selected setting value or the default value if the setting is not found)

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

WorkspaceGetInfo WorkspaceSetDetails WorkspaceSetDocFilter WorkspaceSetFilter WorkspaceGetDetails WorkspaceSettingWrite

67. /WorkspaceSettingWrite [Functions]

[Top] [Functions]

SYNOPSIS

```
STORAGE_API long WorkspaceSettingWrite   (void* pApiData,
                                          const wchar_t* strSettingsPath,
                                          const wchar_t* strSettingValue)
```

FUNCTION

write a archive settings value

INPUTS

```
pApiData -- pointer to the internal data struct
strSettingsPath -- contains the path (in the settings hierarchy) to the selected setting value
strSettingValue -- the value that will be set
```

RETURN VALUE

ErrorCode (>=0 => success, < 0 => failed)

SEE ALSO

WorkspaceGetInfo WorkspaceSetDetails WorkspaceSetDocFilter WorkspaceSetFilter WorkspaceSettingRead WorkspaceGetDetails

Generated from ./ with ROBODoc V4.99.41 on Thu Jan 09 2014 15:10:06