

CA Harvest Software Change Manager

SDK Reference Guide

Release 12.5



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This documentation set references the following CA Technologies products:

- CA Harvest Software Change Manager (CA Harvest SCM)
- CA Software Delivery
- CA IT Client Automation (formerly, CA Desktop and Server Management, CA IT Client Manager)
- CA Agile Vision™
- CA Product Vision

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: Using the Software Development Kit 35

Overview	35
Installing and Setting Up the SDK Components	36
Using the SDK Components.....	36
HSDK Programming Fundamentals	36
Variable Portability Type Definitions	36
HSDK Context	36
Object Lists	37
Object Security	37
Constructors.....	37
HSDK Class Descriptions	37
HSDK Class Hierarchy	38
Additional Class Dependencies	39
HSDK Sample Program	39
Messages.....	39
Using the JHSDK	39
Requirements.....	40
Sample Programs	40
Building and Running the JHSDK Sample Programs	41

Chapter 2: HSDK Class Descriptions 43

CaApprove	44
Void SetApprove(bool bFlag)	44
Void SetDescription(LPCTSTR szDescription)	45
bool GetApprove(void).....	46
LPCTSTR GetDescription(void)	46
HINT32 Execute(void).....	47
CaCheckin	47
bool SetCheckinMode(HINT32 iMode)	49
bool SetPathOption(HINT32 iOption)	50
bool SetItemOption(HINT32 iOption)	51
bool SetClientDir(LPCTSTR szCDir)	52
bool SetViewPath(LPCTSTR szVPath)	52
bool SetDeleteAfterCI(bool bFlag)	53
bool SetDescription(LPCTSTR szDesc)	54
void SetRemoteMachineName(LPCTSTR szRemoteMachineName)	54

void SetAgentPortNumber(LPCTSTR szPortNumber).....	55
void SetRemoteUserName(LPCTSTR szRemoteUserName).....	56
void SetRemotePassword(LPCTSTR szRemotePassword).....	57
void SetCallBack(PFN_CACICB pfCallBack)	58
void SetCBUserData(void* pUserData)	59
void SetFileList(CaStringList &FileList)	59
bool SetNewItemOnBranch(bool bFlag)	60
bool SetNewItemOnTrunk(bool bFlag).....	61
bool SetBranchOut(bool bFlag).....	61
HINT32 GetItemOption(void).....	62
LPCTSTR GetDescription(void)	63
HINT32 GetCheckinMode(void)	63
HINT32 GetPathOption(void)	64
LPCTSTR GetClientDir(void).....	64
LPCTSTR GetViewPath(void)	65
bool GetDeleteAfterCI(void)	66
LPCTSTR GetRemoteMachineName(void)	66
LPCTSTR GetAgentPortNumber(void)	67
LPCTSTR GetRemoteUserName(void)	68
LPCTSTR GetRemotePassword(void).....	68
CaStringList GetFileList(void)	69
bool GetForceDescription(void) const	70
bool GetNewItemOnBranch(void)	71
bool GetNewItemOnTrunk(void)	72
bool GetBranchOut(void)	72
HINT32 CreateItemPath(LPCTSTR szPathName, LPCTSTR szParentPath)	73
PFN_CACICB GetCallBack(void);.....	74
void* GetCBUserData(void);	75
HINT32 Execute(bool bFileList)	76
CaCheckout	77
bool SetCheckoutMode(HINT32 iMode).....	78
bool SetPathOption(HINT32 iOption)	79
bool SetReplaceFile(bool bFlag)	80
bool SetClientDir(LPCTSTR szCDir)	81
bool SetViewPath(LPCTSTR szVPath)	81
bool SetShareWorkDir(bool bFlag)	82
bool SetUseCITimeStamp(bool bFlag).....	83
void SetRemoteMachineName(LPCTSTR szRemoteMachineName).....	83
void SetAgentPortNumber(LPCTSTR szPortNumber).....	84
void SetRemoteUserName(LPCTSTR szRemoteUserName).....	85
void SetRemotePassword(LPCTSTR szRemotePassword)	86
bool SetMVSFileTypes(HINT32 iOption)	88

HINT32 GetCheckoutMode(void)	89
HINT32 GetPathOption(void)	89
LPCTSTR GetClientDir(void)	90
LPCTSTR GetViewPath(void)	90
bool GetBrowseMode(void)	91
bool GetConcUpdateMode(void)	91
bool GetSyncMode(void)	92
bool GetUpdateMode(void)	92
bool GetReplaceFile(void)	93
bool GetShareWorkDir(void)	93
bool GetUseCITimeStamp(void)	94
LPCTSTR GetRemoteMachineName(void)	95
LPCTSTR GetAgentPortNumber(void)	95
LPCTSTR GetRemoteUserName(void)	96
LPCTSTR GetRemotePassword(void)	97
HINT32 GetMVSFileTypes(void)	97
void SetCallBack(PFN_CACOCB pfCallBack)	98
void SetCBUserData(void* pUserData)	99
PFN_CACOCB GetCallBack(void);	100
void* GetCBUserData(void);	101
HINT32 Execute(void)	102
CaConcurrentMerge	103
void SetMergeRule(LPCTSTR szRule)	103
void SetVersionDescription(LPCTSTR szDescription)	104
LPCTSTR GetMergeRule(void)	105
LPCTSTR GetVersionDescription(void)	105
HINT32 Execute(bUseCurrentContext)	106
CaContext	106
LPCTSTR GetProjectName(void) const	110
HAROBJID GetProjectId(void) const	110
LPCTSTR GetStateName(void) const	111
HAROBJID GetStateId(void) const	111
HAROBJID GetProcessId(void) const	112
LPCTSTR GetPackageName(void) const	113
HAROBJID GetPackageId(void) const	113
HAROBJID GetViewName(void) const	114
LPCTSTR GetViewName(void) const	114
HAROBJID GetViewPathId(void) const	115
LPCTSTR GetSnapshotViewName(void) const	115
HAROBJID GetSnapshotViewId(void) const	116
LPCTSTR GetCheckoutName(void) const	117
LPCTSTR GetCheckinName(void) const	117

LPCTSTR GetPromoteName(void) const	118
LPCTSTR GetDemoteName(void) const.....	118
LPCTSTR GetApproveName(void) const.....	119
LPCTSTR GetCreatePackageName(void) const.....	119
LPCTSTR GetDeleteVersionName(void) const.....	120
LPCTSTR GetRemoveItemName(void) const.....	120
LPCTSTR GetConcurrentMergeName(void) const.....	121
LPCTSTR GetCrossProjectMergeName(void) const.....	121
LPCTSTR GetTakeSnapshotName(void) const	122
LPCTSTR GetUDPName(void) const	122
LPCTSTR GetRenameItemName(void) const.....	123
LPCTSTR GetMoveItemName(void) const.....	123
LPCTSTR GetMovePathName(void) const.....	124
LPCTSTR GetRemovePathName(void) const	125
LPCTSTR GetRenamePathName(void) const.....	125
LPCTSTR GetUserName(void) const	126
LPCTSTR GetWebserverName(void)	126
HAROBJID GetUserId(void) const.....	127
LPCTSTR GetClientPath(void) const	127
LPCTSTR GetViewPath(void) const.....	128
LPCTSTR GetInteractiveMergeName(void) const.....	128
Caltem& GetViewPathItem(void)	129
Caltem* GetItemById(HAROBJID itemId).....	130
Caltem* GetItemPathInView(LPCTSTR szFullPath)	130
CaVersion* GetVersionById(HAROBJID iVersionId)	131
CaltemList& GetItemList(bool bRefresh, HINT32 itemPathOption).....	132
CaCheckout& GetCheckout(void) const.....	132
CaCheckin& GetCheckin(void) const.....	133
CaPromote& GetPromote(void) const.....	134
CaDemote& GetDemote(void) const	134
CaApprove& GetApprove(void) const.....	135
CaCreatePackage& GetCreatePackage(void) const	136
CaDeleteVersion& GetDeleteVersion(void) const	136
CaRemoveItem& GetRemoveItem(void) const.....	137
CaConcurrentMerge& GetConcurrentMerge(void) const	138
CaCrossProjectMerge& GetCrossProjectMerge(void) const.....	138
CaInteractiveMerge& GetInteractiveMerge(void) const	139
CaTakeSnapshot& GetTakeSnapshot(void) const.....	140
CaUDP& GetUDP(void) const	140
CaRenameItem& GetRenameItem(void) const.....	141
CaMoveItem& GetMoveItem(void) const.....	142
CaMovePath& GetMovePath(void) const.....	142

CaRemovePath& GetRemovePath(void) const	143
CaRenamePath& GetRenamePath(void) const	144
CaSQL& GetSQL(void) const	144
CaPackageChooser& GetPackageChooser(void) const	145
CaVersionChooser& GetVersionChooser(void) const	146
CaHarvest& GetHarvestSession (void) const	146
CaContainer GetStateList(void) const	147
CaContainer GetViewList(void) const	148
CaContainer GetProcessList(void) const	149
bool SetProject(LPCTSTR szProjectName)	150
bool SetProject(HAROBJID iObjId)	151
bool SetState(LPCTSTR szStateName)	151
bool SetState(HAROBJID iObjId)	152
bool SetCheckout(LPCTSTR szCheckoutName)	153
bool SetCheckout(HAROBJID iObjId)	153
bool SetCheckin(LPCTSTR szCheckinName)	154
bool SetCheckin(HAROBJID iObjId)	155
bool SetInteractiveMerge(LPCTSTR szInteractiveMergeName)	155
bool SetInteractiveMerge(HAROBJID iObjId)	156
bool SetPromote(LPCTSTR szPromoteName)	157
bool SetPromote(HAROBJID iObjId)	157
bool SetDemote(LPCTSTR szDemoteName)	158
bool SetDemote(HAROBJID iObjId)	159
bool SetCrossProjectMerge(LPCTSTR szCrossProjectMergeName)	159
bool SetCrossProjectMerge(HAROBJID iObjId)	160
bool SetTakeSnapshot(LPCTSTR szTakeSnapshotName)	161
bool SetTakeSnapshot(HAROBJID iObjId)	161
bool SetUDP(LPCTSTR szUDPName)	162
bool SetUDP(HAROBJID iObjId)	163
bool SetApprove(LPCTSTR szApproveName)	163
bool SetApprove(HAROBJID iObjId)	164
void SetWebserverName(LPCTSTR szWebserverName)	165
bool SetCreatePackage(LPCTSTR szCreatePackageName)	165
bool SetCreatePackage(HAROBJID iObjId)	166
bool SetDeleteVersion(LPCTSTR szDeleteVersionName)	167
bool SetDeleteVersion(HAROBJID iObjId)	167
bool SetRemoveItem(LPCTSTR szRemoveItemName)	168
bool SetRemoveItem(HAROBJID iObjId)	169
bool SetConcurrentMerge(LPCTSTR szConcurrentMergeName)	169
bool SetConcurrentMerge(HAROBJID iObjId)	170
bool SetRenameItem(LPCTSTR szRenameItemName)	171
bool SetRenameItem(HAROBJID iObjId)	171

bool SetMoveItem(LPCTSTR szMoveItemName)	172
bool SetMoveItem(HAROBJID iObjId)	173
bool SetMovePath(LPCTSTR szMovePathName)	173
bool SetMovePath(HAROBJID iObjId)	174
bool SetRemovePath(LPCTSTR szRemovePathName)	175
bool SetRemovePath(HAROBJID iObjId)	175
bool SetRenamePath(LPCTSTR szRenamePathName)	176
bool SetRenamePath(HAROBJID iObjId)	177
bool SetPackage(LPCTSTR szPackageName)	177
bool SetPackage(HAROBJID iObjId)	178
bool SetSnapshotView(LPCTSTR szSnapshotViewName)	179
bool SetSnapshotView(HAROBJID iObjid)	179
void SetClientPath(LPCTSTR szClientPath)	180
void SetViewPath(LPCTSTR szViewPath)	181
void SetViewPathById(HAROBJID iObjId)	181
bool IsPackageSet()	182
CaCreatePackage	182
void SetPackageName(LPCTSTR szPackageName)	183
void SetAssignedUserName(LPCTSTR szAssignedUserName)	184
void SetDescription(LPCTSTR szDescription)	184
void SetPriority(HINT32 iPriority)	185
LPCTSTR GetDefaultName (void)	186
LPCTSTR GetPackageName(void)	186
LPCTSTR GetAssignedUserName(void)	187
LPCTSTR GetDescription(void)	187
HINT32 GetPriority(void)	188
HINT32 Execute(void)	188
CaCrossProjectMerge	189
void SetFromState(LPCTSTR szProjectName, LPCTSTR szStateName)	190
void SetFromState(HAROBJID iProjectId, HAROBJID iStatId)	191
void SetMergeRule(LPCTSTR szRule)	192
void SetPlacement(LPCTSTR szPlacement)	193
void SetVersionDescription(LPCTSTR szDescription)	193
HAROBJID GetFromProjectId(void)	194
LPCTSTR GetFromProjectName(void)	194
HAROBJID GetFromStatId(void)	195
LPCTSTR GetFromStateName(void)	196
LPCTSTR GetMergeRule(void)	196
LPCTSTR GetPlacement(void)	197
LPCTSTR GetVersionDescription(void)	198
HINT32 Execute(void)	199
CaDeletePackage	199

HINT32 Execute(LPCTSTR SourcePackage).....	200
HINT32 Execute(HAROBJID SourcePackage)	201
CaDeleteVersion.....	201
HINT32 Execute(void).....	202
CaDemote.....	202
bool SetCheckPackageDependencies(bool bCheckPackageDependencies)	203
bool SetCheckPackageGroupBind(bool bFlag)	203
bool GetCheckPackageDependencies (void).....	204
bool GetCheckPackageGroupBind (void)	204
HINT32 Execute(void).....	205
CaException	206
CaException(bool bKey, LPCTSTR szMessage)	207
CaException(const CaException &CRight).....	207
CaException(LPCTSTR szKey, ...).....	208
void SetReturnCode(HINT32 iReturnCode).....	209
HINT32 GetReturnCode(void)	209
CaString GetMessage(void).....	210
void LogMessage(void).....	210
CaForm	211
CaForm(CaFormType& rFormType, LPCTSTR szName).....	212
CaForm(CaPackage& rPackage)	212
CaContainer& GetHistory()	213
bool GetColumnHasValue(LPCTSTR szColName)	214
CaFormType& GetFormType(void)	214
CaData& GetColumnValue(LPCTSTR szColName)	215
bool SetColumnValue(LPCTSTR szColName, const CaData& rValue).....	216
LPCTSTR GetFormTypeName(void)	216
HAROBJID GetFormTypeObjId(void)	217
bool AddPackage(CaPackage *pPackage)	218
bool RemovePackage(CaPackage& rPackage)	218
CaPackageList& GetPackageList()	219
CaPackageList& GetNotInPackageList()	220
CaContainer& GetFormAttachmentList().....	220
bool AddFormAttachment(CaString szAttachmentName, int iType, bool bCompress).....	221
bool RemoveFormAttachment(CaString szAttachmentName)	222
bool RemoveFormAttachment(HAROBJID iObjId)	223
CaFormAttachment* GetFormAttachmentById(HAROBJID ObjId).....	224
HINT32 Delete().....	225
HINT32 Get(HAROBJID iObjId)	225
bool SetName(LPCTSTR szName).....	226
HINT32 Update()	227

void SetFileAgent(LPTSTR szRemoteMachineName, LPTSTR szRemoteUserName, LPTSTR szRemotePassword, LPTSTR szClientPath, LPTSTR szPortNumber)	227
CaFormType	228
CaData GetColumnDefaultValue(HINT32 iIndex)	229
LPCTSTR GetTableName(void)	229
HINT32 GetColumnIndex(LPCTSTR szColumnName)	230
HINT32 GetColumnMaxLength(HINT32 iIndex)	231
LPCTSTR GetColumnName(HINT32 iIndex)	231
CaData::Types GetColumnType(HINT32 iIndex)	232
HUINT32 GetColumnCount(void)	233
CaFormAttachment	233
bool GetCompression(void)	234
void SetCompression(bool bCompression)	234
HINT32 GetAttachmentType(void)	235
void SetAttachmentType(HINT32 iType)	236
void SetFormObjId(HINT32 ObjId)	236
HINT32 GetFileSize(void)	237
bool CopyFileData()	237
void SetFileAgent(LPTSTR szRemoteMachineName, LPTSTR szRemoteUserName, LPTSTR szRemotePassword, LPTSTR szClientPath, LPTSTR szPortNumber)	238
CaHarvest	239
CaHarvest(void)	240
CaHarvest(LPCTSTR szBrokerName)	240
CaString GetHSDKRelease(void)	241
HINT32 SetBroker(LPCTSTR szBrokerName)	242
HINT32 Login(LPCTSTR szBrokerName, LPCTSTR szUserName, LPCTSTR szPassword)	242
HINT32 Login(LPCTSTR szUserName, LPCTSTR szPassword)	243
bool GetAuthenticationSetting(enum tAuthenticationServerSettings nSetting)	244
bool GetCaseSensLogin()	245
LPCTSTR GetLastMessage()	245
HINT32 ChangePassword(LPCTSTR szPassword)	246
void Logout(void)	247
LPCTSTR GetBrokerName(void) const	248
CaContext& GetContext(void) const	248
CaContainer GetProductInfo(void) const	249
CaContainer GetProjectList(void) const	250
CaContainer GetActiveProjectList(void) const	251
CaContainer GetInactiveProjectList(void) const	252
CaContainer GetStateList(HAROBJID iProject) const	253
CaContainer GetUserList(void) const	254
bool SetLog(ostream &rLogStream)	255
static void SetStaticLog(ostream &rLogStream)	256

static ostream& GetLog(void)	256
HINT32 SetFileAccess(LPCTSTR szFullPath, LPCTSTR szAccess).....	257
CaFormTypeList GetFormTypeList(bool bRefresh)	258
CalInteractiveMerge.....	258
LPCTSTR GetCompress(void)	259
LPCTSTR GetDescription(void)	260
HINT32 GetExecuteFlag(void)	260
LPCTSTR GetParentPackageName(void)	261
LPCTSTR GetParentStateName(void)	261
LPCTSTR GetParentMappedVer(void)	262
LPCTSTR GetMergedEnvName(void).....	262
LPCTSTR GetMergedStateName(void)	263
LPCTSTR GetMergedPackageName(void)	263
LPCTSTR GetMergedMappedVer(void)	264
bool GetExtMergeToolFlag(void)	264
LPCTSTR GetMergedFile(void)	265
LPCTSTR GetResolveFlag(void).....	265
LPCTSTR GetScriptFile(void)	266
bool SetDescription(LPCTSTR szDesc)	267
void SetExecuteFlag(HINT32 iExecuteFlag)	267
void SetExtMergeToolFlag(bool bExtTool).....	268
void SetMergedFile(LPCTSTR szMergedFile)	268
void SetResolveFlag(LPCTSTR szResolveFlag).....	269
void SetScriptFile(LPCTSTR szScriptFile)	269
HINT32 Execute(HINT32 iVersionId)	270
HINT32 Resolve(HINT32 iVersionId).....	270
Caltem	271
Caltem(const Caltem &src)	272
CaTimeStamp& GetCreationTime(void) const	273
LPCTSTR GetCreatorName(void) const	273
HAROBJID GetCreatorObjId(void) const	274
CaltemList& GetItemList(void) const	275
CaltemList& GetItemPathList(void) const	275
CaTimeStamp& GetModifiedTime(void) const	276
LPCTSTR GetModifierName(void) const.....	277
HAROBJID GetModifierObjId(void) const.....	277
LPCTSTR GetName(void) const.....	278
HAROBJID GetObjId(void) const.....	279
LPCTSTR GetParentPathName(void) const	280
HAROBJID GetParentRepositoryObjId(void) const.....	280
LPCTSTR GetPathFullName(void) const	281
HAROBJID GetViewObjId(void) const.....	282

CaVersionList& GetVersionList(void) const	283
LPCTSTR GetViewType(void) const	283
bool IsItemPath(void) const	284
CaMoveItem	285
bool SetDescription(LPCTSTR szDesc)	286
void SetItemObjId(HAROBJID iItemObjId)	287
void SetItemName(LPCTSTR szItemName)	288
void SetItemPath(LPCTSTR szItemPath)	288
bool SetTargetItemPath (LPCTSTR szTargetItemPathName)	289
void SetTargetItemPathObjId(HAROBJID iTargetItemPathObjId)	290
bool SetClientDir(LPCTSTR szClientDirectory)	290
bool SetCreateOnBranch(bool bFlag)	291
bool SetCreateOnTrunk(bool bFlag)	292
bool SetBranchOut(bool bFlag)	292
bool SetRefactorMode(HINT32 iRefactorMode)	293
void SetRemoteMachineName(LPCTSTR szRemoteMachineName)	294
void SetAgentPortNumber(LPCTSTR szPortNumber)	295
void SetRemoteUserName(LPCTSTR szRemoteUserName)	296
void SetRemotePassword(LPCTSTR szRemotePassword)	297
LPCTSTR GetRemoteMachineName(void)	297
LPCTSTR GetAgentPortNumber(void)	298
LPCTSTR GetRemoteUserName(void)	299
LPCTSTR GetRemotePassword(void)	299
void SetCallBack(PFN_CAMVICB pfCallBack)	300
void SetCBUserData(void* pUserData)	301
LPCTSTR GetDescription(void)	302
HAROBJID GetItemObjId(void)	303
LPCTSTR GetItemName(void)	303
LPCTSTR GetItemPath (void)	304
LPCTSTR GetTargetItemPath (void)	304
HAROBJID GetTargetItemPathObjId(void)	305
SetItemObjIdLPCTSTR GetClientDir(void)	305
bool GetCreateOnBranch(void)	306
bool GetCreateOnTrunk(void)	306
bool GetBranchOut(void)	307
HINT32 GetRefactorMode(void)	307
PFN_CARNICB GetCallBack(void);	308
void* GetCBUserData(void);	309
HINT32 Execute(void)	310
CaMovePackage	311
void SetToState(LPCTSTR szProjectName, LPCTSTR szStateName)	311
void SetToState(HAROBJID szProjectId, HAROBJID szStateld)	312

void SetIncludePkgHistory(bool bFlag).....	312
HAROBJID GetToProjectId(void)	313
LPCTSTR GetToProjectName(void).....	313
HAROBJID GetToStateId(void).....	314
LPCTSTR GetToStateName(void).....	314
bool GetIncludepkgHistory(void)	315
HINT32 Execute(void).....	315
CaMovePath	316
bool SetDescription(LPCTSTR szDesc)	317
void SetItemPathObjId(HAROBJID iItemPathObjId).....	318
void SetItemPath(LPCTSTR szItemPathName)	318
bool SetTargetItemPath (LPCTSTR szTargetItemPathName)	319
void SetTargetItemPathObjId(HAROBJID iTargetItemPathObjId)	320
bool SetCreateOnBranch(bool bFlag)	320
bool SetCreateOnTrunk(bool bFlag)	321
bool SetBranchOut(bool bFlag).....	322
bool SetRefactorMode(HINT32 iRefactorMode)	322
LPCTSTR GetDescription(void)	323
HAROBJID GetItemPathObjId(void)	323
LPCTSTR GetItemPath (void)	324
LPCTSTR GetTargetItemPath (void).....	324
HAROBJID GetTargetItemPathObjId(void)	325
bool GetCreateOnBranch(void).....	325
bool GetCreateOnTrunk(void).....	326
bool GetBranchOut(void)	327
HINT32 GetRefactorMode(void)	327
HINT32 Execute(void).....	328
CaPackage	329
CaPackage(const CaPackage &src).....	330
CaFormList GetFormListAsList(bool bRefresh).....	331
CaContainer GetApprovalReport(void) const	332
LPCTSTR GetAssignedUserName(void) const.....	333
HAROBJID GetAssignedUserObjId(void) const	333
HAROBJID GetBindPackageGroupObjId(void) const	334
CaTimeStamp& GetCreationTime(void) const	335
LPCTSTR GetCreatorName(void) const	335
HAROBJID GetCreatorObjId(void) const	336
LPCTSTR GetDescription(void) const.....	337
LPCTSTR GetEnvironmentName(void) const.....	337
HAROBJID GetEnvironmentObjId(void) const	338
CaContainer GetFormList(void) const	339
CaContainer GetHistory(void) const	340

CaTimeStamp& GetModifiedTime(void) const	340
LPCTSTR GetModifierName(void) const.....	341
HAROBJID GetModifierObjId(void) const.....	342
LPCTSTR GetName(void) const.....	342
HAROBJID GetObjId(void) const.....	343
HINT32 GetPriority(void) const	343
CaTimeStamp& GetStateEntryTime(void) const	344
LPCTSTR GetStateName(void) const	344
HAROBJID GetStateObjId(void) const	345
CaVersionList& GetVersionList(void) const.....	346
LPCTSTR GetViewName(void) const.....	346
HAROBJID GetViewObjId(void) const.....	347
CaPackageChooser	348
void Clear(void)	349
void SetPackageName(LPCTSTR szName, int iIndex = 0)	350
void SetPackageObjId(HAROBJID iObjId, int iIndex = 0).....	350
void SetProjectObjId(HAROBJID iObjId)	351
void SetStateObjId (HAROBJID iObjId)	351
void SetPkgGroupObjId(HAROBJID iObjId).....	352
void SetAssigneeObjId (HAROBJID iObjId)	352
void SetFromDate(CaTimeStamp& Time)	353
void SetToDate(CaTimeStamp& Time).....	353
void SetCreatorObjId (HAROBJID iObjId)	354
void SetModifierObjId (HAROBJID iObjId).....	354
void SetApproveUserObjId (HAROBJID iObjId)	355
void SetPriority(HINT32 iPriority).....	355
void SetSubString(LPCTSTR szString).....	356
void SetPkgStatus(HINT32 iPkgStatus).....	356
void SetPkgEntryFromDate(CaTimeStamp& Time)	357
void SetPkgEntryToDate(CaTimeStamp& Time)	358
CaContainer GetPackageList(void) const	358
CaPackageList& GetPackageListAsList(void)	359
void Execute(bool bUseCurrentContext)	360
CaPromote	361
bool SetMergedPkgsOnly(bool bFlag)	361
bool SetCheckPackageDependencies(bool bCheckPackageDependencies)	362
bool SetCheckPackageGroupBind(bool bFlag)	362
bool GetMergedPkgsOnly(void)	363
bool GetCheckPackageDependencies (void).....	363
bool GetCheckPackageGroupBind (void)	364
HINT32 Execute(void).....	364
CaRemoveItem	365

bool SetDescription(LPCTSTR szDesc)	366
void SetItemObjId(HAROBJID iltemObjId)	367
void SetItemName(LPCTSTR szItemName)	367
void SetItemPath(LPCTSTR szItemPath)	368
void SetRecursive(bool bRecursive)	369
bool SetCreateOnBranch(bool bFlag)	369
bool SetCreateOnTrunk(bool bFlag)	370
bool SetBranchOut(bool bFlag)	371
LPCTSTR GetDescription(void)	371
HAROBJID GetItemObjId(void)	372
LPCTSTR GetItemName(void)	372
LPCTSTR GetItemPath (void)	373
bool GetRecursive(void)	373
bool GetCreateOnBranch(void)	374
bool GetCreateOnTrunk(void)	374
bool GetBranchOut(void)	375
HINT32 Execute(void)	375
CaRemovePath	376
bool SetDescription(LPCTSTR szDesc)	377
void SetItemPathObjId(HAROBJID iltemPathObjId)	378
void SetItemPath(LPCTSTR szItemPathName)	378
bool SetCreateOnBranch(bool bFlag)	379
bool SetCreateOnTrunk(bool bFlag)	380
bool SetBranchOut(bool bFlag)	380
LPCTSTR GetDescription(void)	381
HAROBJID GetItemPathObjId(void)	381
LPCTSTR GetItemPath (void)	382
bool GetCreateOnBranch(void)	382
bool GetCreateOnTrunk(void)	383
bool GetBranchOut(void)	384
HINT32 Execute(void)	384
CaRenameItem	385
bool SetDescription(LPCTSTR szDesc)	387
void SetItemObjId(HAROBJID iltemObjId)	387
void SetItemName(LPCTSTR szItemName)	388
void SetItemPath(LPCTSTR szItemPath)	389
bool SetNewItemName(LPCTSTR szNewItemName)	389
bool SetClientDir(LPCTSTR szClientDirectory)	390
bool SetCreateOnBranch(bool bFlag)	391
bool SetCreateOnTrunk(bool bFlag)	391
bool SetBranchOut(bool bFlag)	392
bool SetRefactorMode(HINT32 iRefactorMode)	393

void SetRemoteMachineName(LPCTSTR szRemoteMachineName)	393
void SetAgentPortNumber(LPCTSTR szPortNumber)	394
void SetRemoteUserName(LPCTSTR szRemoteUserName)	395
void SetRemotePassword(LPCTSTR szRemotePassword)	396
void SetCallBack(PFN_CARMICB pfCallBack)	397
void SetCBUserData(void* pUserData)	398
LPCTSTR GetDescription(void)	398
HAROBJID GetItemObjId(void)	399
LPCTSTR GetItemName(void)	399
LPCTSTR GetItemPath (void)	400
LPCTSTR GetNewItemName(void)	400
LPCTSTR GetClientDir(void)	401
bool GetCreateOnBranch(void)	401
bool GetCreateOnTrunk(void)	402
bool GetBranchOut(void)	403
HINT32 GetRefactorMode(void)	403
LPCTSTR GetRemoteMachineName(void)	404
LPCTSTR GetAgentPortNumber(void)	405
LPCTSTR GetRemoteUserName(void)	405
LPCTSTR GetRemotePassword(void)	406
PFN_CARNICB GetCallBack(void);	407
void* GetCBUserData(void);	407
HINT32 Execute(void)	408
CaRenamePath	409
bool SetDescription(LPCTSTR szDesc)	410
void SetItemPathObjId(HAROBJID iItemPathObjId)	411
void SetItemPath(LPCTSTR szItemPath)	411
bool SetNewItemPathName(LPCTSTR szNewItemPathName)	412
bool SetCreateOnBranch(bool bFlag)	413
bool SetCreateOnTrunk(bool bFlag)	413
bool SetBranchOut(bool bFlag)	414
bool SetRefactorMode(HINT32 iRefactorMode)	415
LPCTSTR GetDescription(void)	415
HAROBJID GetItemPathObjId(void)	416
LPCTSTR GetItemPath (void)	416
LPCTSTR GetNewItemPathName(void)	417
bool GetCreateOnBranch(void)	417
bool GetCreateOnTrunk(void)	418
bool GetBranchOut(void)	419
HINT32 GetRefactorMode(void)	419
HINT32 Execute(void)	420
CaSQL	421

void SetSQLStatement(LPCTSTR szSQLStr)	422
CaContainer GetSQLResult(void)	422
HINT32 Execute(void)	423
CaSwitchPackage	424
HINT32 Execute(LPCTSTR SourcePackage, LPCTSTR TargetPackage)	424
HINT32 Execute(HAROBJID SourcePackage, HAROBJID TargetPackage)	425
CaTakeSnapshot	426
bool SetCanViewExternallyFlag(bool bFlag)	426
bool GetCanViewExternallyFlag(bool bFlag)	427
HINT32 Execute(LPCTSTR szViewName, CaTimeStamp snapshottime=NULL_CATIMESTAMP, LPCTSTR szReferenceSnapshot = NULL)	428
CaUDP	428
bool SetUsePackageList(bool bUsePackageChooser)	429
bool SetUseVersionList(bool bUseVersionChooser)	430
void SetInput(LPCTSTR szAddInput)	431
void SetAddCmdLineArg(LPCTSTR szCmdLineArg)	432
LPCTSTR GetUDPTYPE(void)	433
LPCTSTR GetProgramName(void)	433
LPCTSTR GetDescription(void)	434
bool GetUserCanModifyInput(void)	434
bool GetUsePackageList(void)	435
bool GetUseVersionList(void)	436
LPCTSTR GetAddCmdLineArg(void)	437
LPCTSTR GetInput(void)	438
HINT32 Execute(void)	439
CaVersion	439
CaVersion(const CaVersion &src)	441
LPCTSTR GetClientMachine(void) const	441
LPCTSTR GetClientPath(void) const	442
CaTimeStamp& GetCreationTime(void) const	443
LPCTSTR GetCreatorName(void) const	443
HAROBJID GetCreatorObjId(void) const	444
LPCTSTR GetDescription(void) const	445
HAROBJID GetItemObjId(void) const	445
HINT32 GetItemType(void) const	446
LPCTSTR GetMappedVersionName(void) const	447
HAROBJID GetMergedVersionId(void) const	447
CaTimeStamp& GetModifiedTime(void) const	448
LPCTSTR GetModifierName(void) const	449
HAROBJID GetModifierObjId(void) const	449
LPCTSTR GetName(void) const	450
HAROBJID GetObjId(void) const	451

LPCTSTR GetPackageName(void) const	452
HAROBJID GetPackageObjId(void) const	452
LPCTSTR GetParentPathName(void) const	453
HAROBJID GetParentVersionObjId(void) const	454
LPCTSTR GetPathFullName(void) const	454
HAROBJID GetVersionDataObjId(void) const	455
LPCTSTR GetVersionStatus(void) const	456
HAROBJID GetViewObjId(void) const	457
LPCTSTR GetRefactorType(void) const	457
HAROBJID GetRefactorByPathVersionObjId(void) const	458
HAROBJID GetPathVersionObjId(void) const	459
CaVersionChooser	460
void Clear(void)	461
void SetParentPath(LPCTSTR szParentPath)	462
void SetExcludePaths(bool bFlag)	462
void RemoveVersionFromList(HAROBJID iVersionObjId)	463
void SetItemObjId(HAROBJID iItemObjId, int iIndex)	463
void SetItemName(LPCTSTR szItemName)	464
void SetItemPackageObjId(HAROBJID iPackageObjId, int iIndex)	465
void SetPackageObjId(HAROBJID iPackageObjId, int iIndex)	465
void SetRecursive(bool bFlag)	466
void SetBranchOption(HINT32 iOptions)	467
void SetVersionOption(HINT32 iOptions)	468
void SetVersionItemOption(HINT32 iOptions)	469
void SetVersionNumber(LPCTSTR szVersion)	469
void SetVersionStatusOption(HINT32 iOptions)	470
void SetVersionDateOption(HINT32 iOptions)	471
void SetVersionObjId(HAROBJID iVersionObjId, int iIndex)	472
CaContextvoid SetFromDate(CaTimeStamp& Time	473
void SetToDate(CaTimeStamp& Time)	473
CaContainer GetVersionList(void) const	474
CaVersionList& GetVersionListAsList(void)	475
void Execute(void)	476

Chapter 3: HUtilities Class Descriptions 477

HUtilities Programming Fundamentals	477
CaBlob	478
void CaBlob(const CaBlob& rBlob)	478
void CaBlob(HINT32 nSize, void * lpData)	479
HINT32 GetSize(void) const	479
void * GetData(void) const	480

CaBlob& operator=(const CaBlob &blob)	481
bool operator==(const CaBlob &rSrc)	481
bool operator!=(const CaBlob &rSrc)	481
friend ostream& operator<<(ostream &o, const CaBlob &rBlob)	482
CaContainer	482
void Clear(void)	483
CaData & Props(LPCTSTR strKey, HINT32 iIndex)	484
HUINT32 GetKeyCount(void) const	484
CaStringList GetKeyList(void) const	485
HUINT32 GetKeyElementCount(LPCTSTR strKey) const	486
bool IsKeyExist(LPCTSTR strKey) const	486
bool IsEmpty(void) const	487
CaDataArray FindData(LPCTSTR strKey, LPCTSTR strValue) const	487
CaDataArray FindData(LPCTSTR strKey, HINT32 iValue) const	488
CaContainer & operator=(const CaContainer &CRight)	489
CaContainer & operator+=(const CaContainer &CRight)	489
friend ostream& operator<<(ostream &o, const CaContainer &c)	490
CaData	491
void SetNull(void)	491
bool IsNull(void) const	492
CaData::Types GetType(void) const	493
CaData& operator=(const CaData &aData)	494
CaData& operator=(HINT32 nValue)	494
CaData& operator=(LPCTSTR szValue)	495
CaData& operator=(const CaBlob& rBlob)	495
CaData& operator=(const CaTimeStamp& rTimeStamp)	496
bool operator==(const CaData &rSrc)	496
bool operator!=(const CaData &rSrc)	497
operator HINT32()	497
operator LPCTSTR()	497
operator CaString()	498
operator (CaBlob&)()	498
operator (CaTimeStamp&)()	499
friend ostream& operator<<(ostream &o, const CaData &d)	499
CaDataArray	500
void Clear(void)	500
int GetSize(void) const	501
bool Find (const CaData& rData, HUINT32& rIndex) const	501
CaData & operator[] (HUINT32 index)	502
friend ostream& operator<<(ostream &o, const CaDataArray &a)	503
CaString	503
CaString(void)	504

CaString(const CaString &src)	505
CaString(LPCTSTR lpStr).....	505
CaString(char* lpStr)	506
CaString& Append(LPCTSTR src)	506
CaString& ToUpper (void)	507
CaString& ToLower (void)	507
bool IsEmpty(void) const.....	508
CaString& MakeEmpty(void)	508
long GetLength(void) const	509
CaString& Trim(void)	509
bool IsValid(void) const	510
const char* GetMultiByteString(void)	510
CaString& operator=(const CaString &src)	511
CaString& operator=(LPCTSTR szSrc)	511
CaString& operator=(char* szSrc)	512
bool operator==(const CaString &rSrc)	512
bool operator==(LPCTSTR src) const	513
bool operator!=(const CaString &rSrc) const	513
bool operator!=(LPCTSTR src) const.....	514
CaString& operator+=(LPCTSTR src)	514
CaString& operator-=(LPCTSTR src).....	514
operator LPCTSTR()	515
operator char *().....	515
friend ostream& operator<<(ostream &o, const CaString &rString)	516
CaStringList.....	516
CaStringList(void)	517
CaStringList(const CaStringList &src)	518
bool Add(const CaString szString)	518
CaString & Find(const CaString szString).....	519
bool Remove(const CaString szString)	519
bool Add(LPCTSTR szString)	520
CaString & Find(LPCTSTR szString)	520
bool Remove(LPCTSTR szString).....	521
bool IsEmpty(void) const.....	522
void Clean(void).....	522
int GetSize(void) const	523
CaString GetAllNames(void) const	523
Int GetIndex(LPCTSTR szString) const	524
CaStringList& operator=(const CaStringList & CRight)	524
bool operator==(const CaStringList & CRight) const.....	525
bool operator!=(const CaStringList & CRight) const	525
CaStringList& operator+=(const CaStringList & CRight)	525

CaStringList& operator==(const CaStringList & CRight).....	526
CaString & operator[] (int i)	526
operator bool() const	527
friend ostream& operator<<(ostream &o, const CaStringList &rList).....	528
CaTimeStamp	528
void CaTimeStamp(void)	529
void CaTimeStamp(const CaTimeStamp& rTimeStamp).....	529
void CaTimeStamp(HINT32 nYear, HINT32 nMonth, HINT32 nDay, HINT32 nHour, HINT32 nMinute, HINT32 nSecond, HINT32 nFraction)	530
CaTimeStamp& SetYear(HINT32 iYear)	530
CaTimeStamp& SetMonth(HINT32 iMonth)	531
CaTimeStamp& SetDay(HINT32 iDay)	531
CaTimeStamp& SetHour(HINT32 iHour)	532
CaTimeStamp& SetMinute(HINT32 iMinute)	532
CaTimeStamp& SetSecond(HINT32 iSecond)	533
CaTimeStamp& SetFraction(HINT32 iFraction)	533
HINT32 GetYear(void) const.....	534
HINT32 GetMonth(void) const	534
HINT32 GetDay(void) const	535
HINT32 GetHour(void) const	535
HINT32 GetMinute(void) const	536
HINT32 GetSecond(void) const	536
HINT32 GetFraction(void) const.....	537
CpTTimeStamp& ChangeTime(HINT32 iTimezone).....	537
CpTTimeStamp& ConvertGMTTime(void).....	538
CaTimeStamp& operator=(const CaTimeStamp &rTimeStamp)	539
bool operator==(const CaTimeStamp &rTimeStamp) const	539
bool operator!=(const CaTimeStamp &rTimeStamp) const.....	540
friend ostream& operator<<(ostream &o, const CaTimeStamp &rTimeStamp)	540

Chapter 4: Using the COM Interface of the Software Development Kit (CHSDK) 541

Overview	541
CHSDK Messages.....	541
CHSDK Programming Fundamentals	541
Usage.....	541
CHSDK Context	542
Object Lists and Security	542

Chapter 5: CHSDK Class Descriptions 543

CaWApprove	543
------------------	-----

Approve Property	544
Description Property	545
Execute Method	545
IsValid Method	546
CaWCheckin	546
RemoteMachineName Property	548
AgentPortNumber Property	550
RemoteUserName Property	551
RemotePassword Property	552
SetCheckinMode Method	553
SetPathOption MethodCaCheckin	554
SetItemOption Method	555
SetClientDir Method	556
SetViewPath Method	557
SetDeleteAfterCI Method.....	557
SetDescription Method	558
SetFileList Method	559
GetItemOption Method	560
GetDescription Method	560
GetCheckinMode Method.....	561
GetNewItemOnBranch Method	562
GetNewItemOnTrunk Method	562
SetBranchOut Method	563
GetBranchOut Method	564
GetPathOption Method	564
GetClientDir Method.....	565
GetViewPath Method	566
GetDeleteAfterCI Method.....	567
GetFileList Method.....	567
Execute Method	568
IsValid Method	569
CaWCheckout.....	570
RemoteMachineName Property	571
AgentPortNumber Property	573
RemoteUserName Property	574
RemotePassword Property	574
SetCheckoutMode Method.....	575
SetPathOption Method	577
SetReplaceFile Method	578
SetClientDir Method	578
SetViewPath Method	579
SetShareWorkDir Method.....	580

SetUseCITimeStamp Method	581
SetMVSFileTypes Method	582
GetCheckoutMode Method	583
GetPathOption Method	584
GetClientDir Method.....	585
GetViewPath Method	585
GetReplaceFile Method	586
GetShareWorkDir Method	587
GetUseCITimeStamp Method	587
GetMVSFileTypes Method	588
Execute Method	589
IsValid Method	590
CaWConcurrentMerge	591
MergeRule Property.....	592
VersionDescription Property.....	593
Execute Method	593
IsValid Method	594
CaWContext	595
GetProjectName Method.....	598
GetProjectId Method	599
GetStateName Method.....	600
GetStatId Method	601
GetProcessId Method	602
GetPackageName Method	603
GetPackagId Method	604
GetViewName Method	605
GetSnapshotViewName Method	606
GetCheckoutName Method	606
GetCheckinName Method	607
GetPromoteName Method	608
GetDemoteName Method	609
GetApproveName Method	610
GetCreatePackageName Method	611
GetDeleteVersionName Method	612
GetRemoveItemName Method.....	613
GetConcurrentMergeName Method	614
GetClientPath Method	615
GetViewPath Method	616
GetViewPathId Method	616
GetCheckout Method.....	617
GetCheckin Method	618
GetPromote Method.....	619

GetDemote Method	620
GetApprove Method	621
GetCreatePackage Method	622
GetDeleteVersion Method	623
GetInteractiveMerge Method	624
GetInteractiveMergeName Method	625
GetRemoveItem Method	626
GetConcurrentMerge Method	627
GetPackageChooser Method	628
GetVersionChooser Method	629
GetStateList Method	630
GetViewList Method	631
GetProcessList Method	632
SetProjectName Method	633
SetProjectId Method	634
SetStateName Method	635
SetStatId Method	636
SetCheckoutName Method	637
SetCheckoutId Method	638
SetCheckinName Method	638
SetCheckinId Method	639
SetPromoteName Method	640
SetPromoteId Method	640
SetDemoteName Method	641
SetDemoteId Method	642
SetApproveName Method	642
SetApproveId Method	643
SetCreatePackageName Method	644
SetCreatePackageId Method	644
SetDeleteVersionName Method	645
SetDeleteVersionId Method	646
SetInteractiveMergeName Method	646
SetInteractiveMergeId Method	647
SetRemoveItemName Method	648
SetRemoveItemId Method	648
SetConcurrentMergeName Method	649
SetConcurrentMergeId Method	650
SetPackageName Method	650
SetPackageId Method	651
SetSnapshotView Method	652
SetClientPath Method	652
SetViewPath Method	653

SetViewPathById Method	654
GetCrossProjectMerge Method	654
GetCrossProjectMergeName Method	655
GetSnapshotViewId Method	656
GetSQL Method.....	657
GetTakeSnapshot Method	657
GetTakeSnapshotName Method.....	658
GetUDP Method.....	659
GetUDPName Method	660
SetCrossProjectMergeById Method.....	660
SetCrossProjectMergeByName Method	661
SetSnapshotViewById Method	662
SetTakeSnapshotById Method.....	662
SetTakeSnapshotByName Method	663
SetUDPByName Method.....	664
SetUDPById Method	664
CaWCreatePackage	665
PackageName Property.....	666
AssignedUserName Property	667
Description Property	668
Priority Property.....	668
DefaultName Property (Read Only)	669
Execute Method	670
IsValid Method	670
CaWCrossProjectMerge	671
MergeRule Property.....	672
Placement Property	673
VersionDescription Property.....	674
SetFromStateByName Method	674
SetFromStateById Method.....	675
GetFromProjectId Method.....	676
GetFromProjectName Method	677
GetFromStateId Method.....	678
GetFromStateName Method	678
IsValid Method	679
Execute Method	680
CaWDeleteVersion	681
Execute Method	681
IsValid Method	682
CaWDemote	683
SetCheckPackageDependencies Method	683
SetCheckPackageGroupBind Method	684

GetCheckPackageDependencies Method	685
GetCheckPackageGroupBind Method.....	685
Execute Method	686
IsValid Method	687
CaWForm.....	687
GetHistory Method	688
GetColumnHasValue Method	689
GetFormType Method.....	690
GetColumnValue Method	690
SetColumnValue Method	691
GetFormTypeName Method	692
GetFormTypeObjId Method	692
AddPackage Method	693
RemovePackage Method	694
GetPackageList Method	694
GetNotInPackageList Method	695
GetFormAttachmentList Method.....	696
AddFormAttachment Method	696
RemoveFormAttachment Method.....	697
RemoveFormAttachment Method.....	698
GetFormAttachmentById Method	699
Update Method.....	700
Get Method.....	700
Delete Method	701
SetName Method	701
InitHObject Method	702
SetFileAgent Method	702
SetFileAgent2 Method	703
CaWFormType.....	704
GetColumnDefaultValue Method	704
GetTableName Method	705
GetColumnIndex Method	706
GetColumnMaxLength Method	706
GetColumnName Method.....	707
GetColumnType Method.....	708
GetColumnCount Method.....	708
CaWFormAttachment	709
GetCompression Method.....	710
SetCompression Method	710
GetAttachmentType Method.....	711
SetAttachmentType Method	712
SetFormObjId Method	712

GetFileSize Method	713
CopyFileData Method	713
SetFileAgent Method	714
SetFileAgent2 Method	715
CaWHarvest.....	715
SetBroker Method.....	716
Login Method.....	717
GetLastMessage Method	717
ChangePassword Method	718
Logout Method	719
GetBrokerName Method	720
GetContext Method	721
GetAuthenticationSetting Method	721
GetCaseSensLogin Method	722
GetProductInfo Method.....	723
GetProjectList Method.....	723
GetUserList Method.....	724
SetLogName Method	725
GetStateList Method	726
CaWInteractiveMerge	727
GetCompress Method	728
GetDescription Method	729
GetExecuteFlag Method.....	730
GetParentPackageName Method	731
GetParentStateName Method	732
GetParentMappedVer Method.....	733
GetMergedEnvName Method.....	734
GetMergedStateName Method	735
GetMergedPackageName Method	735
GetMergedMappedVer Method.....	736
GetExtMergeToolFlag Method.....	737
GetMergedFile Method	738
GetResolveFlag Method.....	739
GetScriptFile Method.....	740
SetDescription Method	741
SetExecuteFlag Method	742
SetExtMergeToolFlag Method	743
SetMergedFile Method	744
SetResolveFlag Method	745
SetScriptFile Method.....	746
Execute Method	747
Resolve Method	748

CaWPackage	749
Delete Method	750
Update Method	750
CaWPackageChooser	751
ProjectObjId Property (Write Only)	752
StateObjId Property (Write Only)	753
PkgGroupObjId Property (Write Only)	754
AssigneeObjId Property (Write Only)	754
FromDate Property (Write Only)	755
ToDate Property (Write Only)	755
CreatorObjId Property (Write Only)	756
ModifierObjId Property (Write Only)	756
ApproveUserObjId Property (Write Only)	757
Priority Property (Write Only)	757
SubString Property (Write Only)	758
PkgStatus Property (Write Only)	758
PkgEntryFromDate Property (Write Only)	759
PkgEntryToDate Property (Write Only)	760
Clear Method	761
GetPackageList Method	761
Execute Method	763
IsValid Method	763
SetPackageName Method	764
CaWPromote	765
SetMergedPkgsOnly Method	765
SetCheckPackageDependencies Method	766
SetCheckPackageGroupBind Method	767
GetMergedPkgsOnly Method	767
GetCheckPackageDependencies Method	768
GetCheckPackageGroupBind Method	769
Execute Method	769
IsValid Method	770
CaWRemoveItem	771
ItemObjId Property	771
ItemName Property	772
ItemPath Property	773
Recursive Property	774
SetDescription Method	775
GetDescription Method	776
Execute Method	776
IsValid Method	777
CaWSwitchPackage	778

Execute Method	778
CaWSQL	779
SetSQLStatement Method	780
GetSQLResult Method	780
Execute Method	781
CaWTakeSnapshot	782
SetCanViewExternallyFlag Method	782
GetCanViewExternallyFlag Method	783
IsValid Method	784
Execute Method	785
CaWUDP	786
UsePackageList Property	787
UseVersionList Property	788
UDPType Property (Read Only)	789
ProgramName Property (Read Only)	789
Description Property (Read Only)	790
UserCanModifyInput Property (Read Only)	790
SetInput Method	791
GetInput Method	792
SetAddCmdLineArg Method	793
GetAddCmdLineArg Method	794
Execute Method	795
CaWVersionChooser	796
ParentPath Property (Write Only)	798
ItemName Property (Write Only)	798
Recursive Property (Write Only)	799
BranchOption Property (Write Only)	800
VersionOption Property (Write Only)	801
VersionItemOption Property (Write Only)	801
VersionStatusOption Property (Write Only)	802
VersionDateOption Property (Write Only)	803
FromDate Property (Write Only)	804
ToDate Property (Write Only)	805
Clear Method	806
SetItemObjId Method	807
SetPackageObjId Method	807
GetVersionList Method	808
Execute Method	810
IsValid Method	810

Chapter 6: COM Wrapper for HUtilities Class Descriptions 813

HUtilities Programming Fundamentals	813
CaWContainer	813
Clear Method	814
Props Method	814
GetKeyCount Method	815
GetKeyList Method	816
GetKeyElementCount Method	816
IsKeyExist Method	817
IsEmpty Method	818
FindData Method	818
FindData2 Method	819
CaWData	820
SetNull Method	821
IsNull Method	821
GetType Method	822
InitHObject Method	823
SetLongValue Method	824
SetStringValue Method	824
SetTimeStampValue Method	825
GetLongValue Method	825
GetStringValue Method	826
GetTimeStampValue Method	826
CaWDataArray	827
Clear Method	827
GetSize Method	828
Find Method	828
GetAt Method	829
CaWStringList	830
Add Method	830
Find Method	831
Remove Method	832
Add Method	832
IsEmpty Method	833
Clean Method	834
GetSize Method	834
GetAllNames Method	835
GetAt Method	835
IsValid Method	836
InitHObject Method	836
CaWTimeStamp	837

SetYear Method	838
SetMonth Method	838
SetDay Method	839
SetHour Method	840
SetMinute Method	840
SetSecond Method	841
SetFraction Method	842
GetYear Method	842
GetMonth Method	843
GetDay Method	844
GetHour Method	844
GetMinute Method	845
GetSecond Method	846
GetFraction Method	846
ChangeTime Method	847
ConvertGMTTime Method	848
InitHObject Method	848

Chapter 1: Using the Software Development Kit

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section provides a brief overview of the CA Harvest SCM Software Development Kit (HSDK) and the Java CA Harvest SCM Software Development Kit (JHSDK), and describes how to use SDK components.

This section contains the following topics:

[Overview](#) (see page 35)

[Installing and Setting Up the SDK Components](#) (see page 36)

[Using the SDK Components](#) (see page 36)

[HSDK Programming Fundamentals](#) (see page 36)

[HSDK Class Descriptions](#) (see page 37)

[HSDK Sample Program](#) (see page 39)

[Messages](#) (see page 39)

[Using the JHSDK](#) (see page 39)

Overview

The HSDK is a set of C++ classes that abstract and encapsulate the persistence and manipulation of CA Harvest SCM data objects. It presents an object model that represents the data and their relationships. The HSDK implements a set of platform-independent client-side interfaces to access CA Harvest SCM data.

The JHSDK is a set of Java classes that “wrap” (act as an interface between) Java and the HSDK. The JHSDK ensures that the Java programs you execute with the HSDK are compatible with CA Harvest SCM and comply with CA Harvest SCM standards. If you want to use Java programs with the HSDK, you *must* use the JHSDK.

The JHSDK retains most of the original HSDK functionality. The JHSDK is dependent on the HSDK at runtime. This document assumes a thorough knowledge of the CA Harvest SCM product and its usage.

Installing and Setting Up the SDK Components

You can install the CA Harvest SCM SDK components on Windows, UNIX, and Linux.

On Windows, if you install the server, the client, or the agent, the SDK components are installed automatically.

On UNIX and Linux, whenever the CA Harvest SCM client (command line utilities) is installed, the CA Harvest SCM SDK components are installed automatically.

After installation, you must set up the SDK components. You can test to ensure that the SDK components were installed successfully.

Note: For instructions to set up and test the SDK components, see the *Implementation Guide*.

Using the SDK Components

After installing, setting up, and testing the SDK components, you can use them to build applications.

Each HSDK class has its own header file that is included in the `h sdk.h` file. The relationships between the various HSDK classes make the order of header file inclusion important. The `h sdk.h` file includes all required HSDK header files in the proper sequence. It is the only header file required to be included in your application files.

HSDK Programming Fundamentals

This section provides basic instructions for writing programs using the HSDK and JHSDK.

Variable Portability Type Definitions

CA Harvest SCM runs on multiple platforms. Integer and character size may differ across the various platforms. C++ typedef statements and #ifdef statements facilitate porting.

HSDK Context

HSDK operations are performed in a specified context. After you log in to the principal CaHarvest object, a CaContext object is available. As you change the current context, HSDK objects are reset or made available based on the new context. The CaContext object owns the other HSDK objects. Get functions return C++ references to objects to indicate this ownership and prevent their accidental deletion.

Object Lists

The HSDK provides a CaContainer object. The CaContainer is a text-keyed hash map of CaDataArray objects.

Object Security

HSDK object security is maintained through the underlying CA Harvest SCM modules.

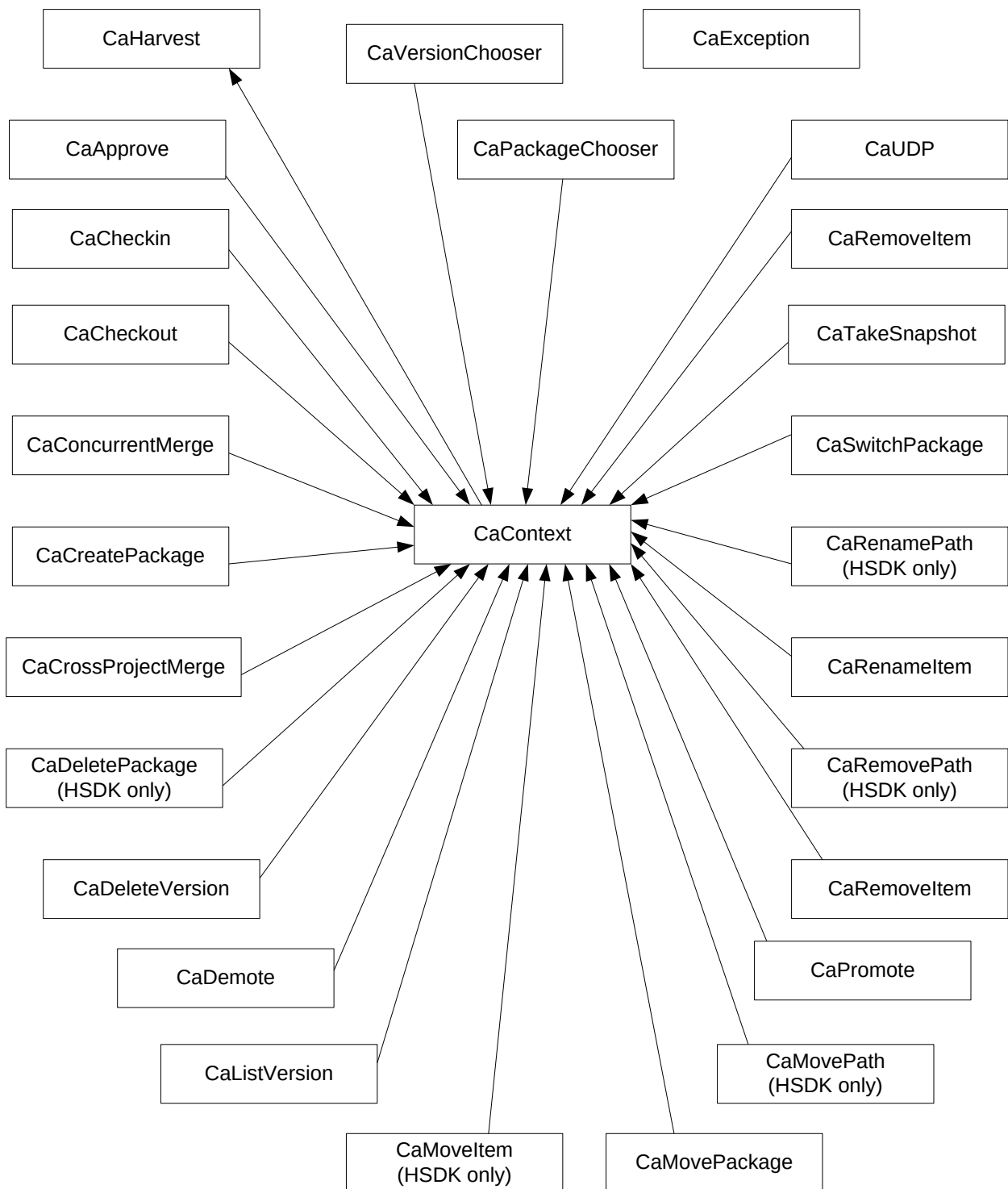
Constructors

The constructors for most HSDK classes are declared private so that object instantiation is restricted to the proper context.

HSDK Class Descriptions

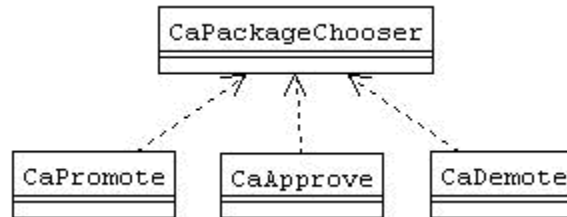
The following diagrams show class hierarchy and dependencies.

HSDK Class Hierarchy

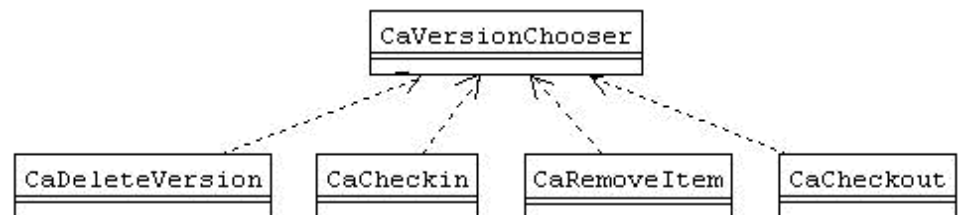


Additional Class Dependencies

Classes that perform operations on lists of packages depend on the CaPackageChooser.



Classes that perform operations on lists of versions depend on the CaVersionChooser.



Note: For more information, see [HSDK Programming Fundamentals](#) (see page 36) in this chapter, and the chapter “Utilities Class Descriptions.”

HSDK Sample Program

The HSDK installation includes a sample program, hsdksample.cpp. The program comes with build algorithms for UNIX, Linux, and Windows environments. Hsdksample demonstrates HSDK functionality by exercising an operational scenario of CA Harvest SCM operations.

Messages

HSDK messages display during the execution of HSDK functions. Other CA Harvest SCM messages may also display, from the CA Harvest SCM dlls.

Note: To view information about messages, see the *Messages Guide*.

Using the JHSDK

Before you can use the JHSDK, it must be installed and you must perform the required setup tasks. For details, see [Installing and Setting Up the SDK Components](#) (see page 36) in this chapter and the *Implementation Guide*.

Requirements

When creating JHSDK programs, ensure that you do the following:

- Enter the names of projects, states, and processes in Java classes names carefully and exactly. These names are *case-sensitive*.
- Follow the instructions in the JHSDK online help.

Sample Programs

The JHSDK includes four cumulative sample programs. That is, Sample 2 builds on Sample 1, Sample 3 builds on Sample 2, and so forth. You *must* execute Sample 2 before Sample 3.

- **Sample 1** (sample1.java) displays some information about projects, users, states, and processes in the repository.
- **Sample 2** (sample2.java) demonstrates the create package and check-in functionality in the JHSDK.
- **Sample 3** (sample3.java) demonstrates the check-out functionality in the JHSDK. Sample checks out the file that Sample 2 checked in.
- **Sample 4** (sample4.java) demonstrates the use of the user-defined process (UDP) functionality in the JHSDK.
- **Sample 5** (sample5.java) demonstrates the use of interactive merge functionality in the JHSDK.
- **Sample 6** (sample6.java) demonstrates the use of form functionality in the JHSDK.

Building and Running the JHSDK Sample Programs

This section describes how to build and run the JHSDK sample programs.

To build the sample programs

1. Ensure the following:

- A version of Sun Java SDK Standard Edition that is supported by CA Harvest SCM is installed on your machine.
- The Java binary programs are specified in the PATH environment variable of your machine.

Note: When running the SampleRunner program on UNIX, set the `java.library.path` to point to the directories that contain libraries to be loaded by the Java virtual machine. Include the directories that contain the CA Harvest SCM and PEC libraries. You can set the `java.library.path` using an environment variable, or by defining the path on the command line:

```
java -Djava.library.path=directories where you put your libraries [ClassToRun] [Parameters]
```

Important! Failure to specify the `java.library.path` may result in unresolved symbols when the `SampleRunner` class is run.

2. Complete the following step:

- (Windows) Create the `%CA_SCM_HOME%\jhsdk\samples\classes` directory.
- (UNIX) Create the `$CA_SCM_HOME/jhsdk/samples/classes` directory.

3. At the command prompt, browse to the `CA_SCM_HOME` directory.

4. Enter the following command to build the sample files:

Windows

```
javac -classpath jhsdk.jar -d jhsdk\samples\classes  
jhsdk\samples\com\ca\harvest\jhsdk\sample\*.java
```

UNIX

```
javac -classpath jhsdk.jar -d jhsdk/samples/classes  
jhsdk/samples/com/ca/harvest/jhsdk/sample/*.java
```

Note: You can optionally create a different directory, rather than the directory referenced in the previous steps, to store the sample programs. If you do so, ensure that any commands you use for building and running the sample programs specify the new directory path.

To run the sample programs

1. Ensure that the `CA_SCM_HOME` directory is specified in the PATH environment variable of your computer.
2. Ensure the classpath includes the `jhsdk.jar` file.

3. Browse to one of the following directories:

- (Windows) jhsdk\samples\classes
- (UNIX) jhsdk/samples/classes

4. At the command prompt, enter the following command:

```
java com.ca.harvest.jhsdk.sample.SampleRunner -b broker name  
-u sdkuser -p sdkpass -s n
```

In this command, *broker name* is the name of your CA Harvest SCM broker. *n* specifies the number corresponding to the Java sample program to run.

The command line arguments for the JHSDK class SampleRunner are documented in the sample code, SampleRunner.java. SampleRunner.java is part of the sample code included in the JHSDK directory contents that you copied earlier, when you installed the JHSDK.

Note: For additional information about the Java classes used by the JHSDK, see the JHSDK HTML Help files located in your local CA Harvest SCM installation directory. For example: C:\Program Files\CA\Software Change Manager\JHSDK\javadoc. The index.html file provides hyperlinks to all JHSDK HTML Help files. Double-click index.html to locate and view the JHSDK HTML help files most efficiently.

Chapter 2: HSDK Class Descriptions

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

- [CaApprove](#) (see page 44)
- [CaCheckin](#) (see page 47)
- [CaCheckout](#) (see page 77)
- [CaConcurrentMerge](#) (see page 103)
- [CaContext](#) (see page 106)
- [CaCreatePackage](#) (see page 182)
- [CaCrossProjectMerge](#) (see page 189)
- [CaDeletePackage](#) (see page 199)
- [CaDeleteVersion](#) (see page 201)
- [CaDemote](#) (see page 202)
- [CaException](#) (see page 206)
- [CaForm](#) (see page 211)
- [CaFormType](#) (see page 228)
- [CaFormAttachment](#) (see page 233)
- [CaHarvest](#) (see page 239)
- [CaInteractiveMerge](#) (see page 258)
- [Caltem](#) (see page 271)
- [CaMoveItem](#) (see page 285)
- [CaMovePackage](#) (see page 311)
- [CaMovePath](#) (see page 316)
- [CaPackage](#) (see page 329)
- [CaPackageChooser](#) (see page 348)
- [CaPromote](#) (see page 361)
- [CaRemoveItem](#) (see page 365)
- [CaRemovePath](#) (see page 376)
- [CaRenameItem](#) (see page 385)
- [CaRenamePath](#) (see page 409)
- [CaSQL](#) (see page 421)
- [CaSwitchPackage](#) (see page 424)
- [CaTakeSnapshot](#) (see page 426)
- [CaUDP](#) (see page 428)
- [CaVersion](#) (see page 439)
- [CaVersionChooser](#) (see page 460)

CaApprove

CaApprove encapsulates the Approve process. It cannot be instantiated directly. The Approve process can be set in the CaContext object either by name or object ID. The CaApprove object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, and Approve Context.
- Select a set of Change Packages to be Approved or Rejected using the CaPackageChooser object.
- Set the Approve attributes.
- Invoke the Execute function.

Public Methods

- SetApprove
- SetDescription
- GetApprove
- GetDescription
- Execute

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[CaPackageChooser](#) (see page 348)

Void SetApprove(bool bFlag)

Sets the Approve/Reject flag true or false.

Default: The Approve/Reject flag defaults to true.

Owning Class

CaApprove

Arguments

bool bFlag-if true, Approve the Change Package list; if false, Reject it

Return Value

None

Example

```
CaApprove& rApprove = rContext.GetApprove(); rApprove.SetApprove(true);
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[bool GetApprove\(void\)](#) (see page 46)

Void setDescription(LPCTSTR szDescription)

Sets the Approve/Reject Description is used to annotate why the package list was approved or rejected.

Default: The Approve/Reject flag defaults to true.

Owning Class

CaApprove

Arguments

LPCTSTR szDescription-description

Return Value

None

Example

```
CaApprove& rApprove = rContext.GetApprove();  
rApprove.SetDescription ("I approve this package");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[LPCTSTR GetDescription\(void\)](#) (see page 46)

bool GetApprove(void)

Gets the current value of the Approve/Reject flag.

Default: The Approve/Reject flag defaults to true.

Owning Class

CaApprove

Arguments

None

Return Value

Flag value-true or false

Example

```
CaApprove& rApprove = rContext.GetApprove();  
bool bFlag = rApprove.GetApprove();
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[Void SetApprove\(bool bFlag\)](#) (see page 44)

LPCTSTR GetDescription(void)

Gets the current value of the Approve/Reject Description.

Owning Class

CaApprove

Arguments

None

Return Value

LPCTSTR-description

Example

```
CaApprove& rApprove = rContext.GetApprove();  
CaString = rApprove.GetDescription();
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[Void SetDescription\(LPCTSTR szDescription\)](#) (see page 45)

HINT32 Execute(void)

Executes the Approve process on the Package list currently selected using the Package Chooser.

Owning Class

CaApprove

Arguments

None

Return Value

HINT32-zero if success, nonzero if failure

Example

```
CaApprove& rApprove = rContext.GetApprove();  
HINT32 iResult = rApprove.Execute();
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[CaPackageChooser](#) (see page 348)

CaCheckin

CaCheckin encapsulates the Checkin process. It cannot be instantiated directly. The Checkin process can be set in the CaContext object either by name or object ID. The CaCheckin object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, Package, and Checkin Context.
- Select either a set of Versions to checkin using the CaVersionChooser object or add a set of file names to a file name list, CaStringList.
- Invoke the Execute function.

Public Methods

- SetCheckinMode
- SetPathOption
- SetItemOption
- SetClientDir
- SetViewPath
- SetDeleteAfterCI
- SetDescription
- SetParentTrunkMappedVersionList
- SetRemoteMachineName
- SetAgentPortNumber
- SetRemoteUserName
- SetRemotePassword
- SetFileList
- SetNewItemOnBranch
- SetNewItemOnTrunk
- SetBranchOut
- GetItemOption
- GetDescription
- GetParentTrunkMappedVersionList
- ClearParentTrunkMappedVersionList
- GetCheckinMode
- GetPathOption
- GetClientDir
- GetViewPath
- GetDeleteAfterCI
- GetRemoteMachineName
- GetAgentPortNumber
- GetRemoteUserName
- GetRemotePassword
- GetFileList

- GetForceDescription
- GetNewItemOnBranch
- GetNewItemOnTrunk
- GetBranchOut
- CreateItemPath
- SetCallBack
- SetCBUserData
- GetCallBack
- GetCBUserData
- Execute

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[CaVersionChooser](#) (see page 460)

[CaStringList](#) (see page 516)

bool SetCheckinMode(HINT32 iMode)

Sets the Checkin Mode for this process.

Note: The following are the allowable values for the Checkin Mode. They are defined in an include file built into the project.

- HAR_UPDATE_AND_RELEASE-Check in the file and release the lock.
- HAR_UPDATE_AND_KEEP-Check in the file and reserve it again for update.
- HAR_RELEASE_ONLY-Release the lock but do not check in the file.

Owning Class

CaCheckin

Arguments

HINT32 iMode-the Checkin Mode

Return Value

True if success, false if fail

Example

```
CaCheckin& rCheckin = rContext.GetCheckin();  
bool bResult = rCheckin.SetCheckinMode(HAR_UPDATE_AND_RELEASE);
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[HINT32 GetCheckinMode\(void\)](#) (see page 63)

bool SetPathOption(HINT32 iOption)

Sets the Path Option for this process. Files are checked into the repository in locations related to the view path, client directory, and the path option.

Note: The following are the allowable values for the Path Option. They are defined in an include file built into the project.

- `CI_OPTION_PRESERVE_DIR`-Check in each new version to its existing view path location that mirrors the location in the client path.
- `CI_OPTION_PRESERVE_AND_CREATE_PATH`-Check in each new version to its view path location that mirrors the location in the client path and create the view path if it does not yet exist.
- `CI_OPTION_ALL_FILE_TO_SAME_PATH`-Check in each new version to the single view path location specified.

Owning Class

CaCheckin

Arguments

HINT32 iOption-the Path Option

Return Value

True if success, false if fail

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); bool bResult = rCheckin.SetPathOption  
(CI_OPTION_PRESERVE_AND_CREATE_PATH);
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[HINT32 GetPathOption\(void\)](#) (see page 64)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 52)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 52)

bool SetItemOption(HINT32 iOption)

Sets the Item Option for this process.

Note: The following are the allowable values for the Item Option. They are defined in an include file built into the project.

- **CI_FILTER_NEW_OR_EXISTING_ITEMS**-When this option is selected, all selected files are checked in if they are reserved by the package or did not previously exist in the repository. This filter does not require the corresponding item to be in the current view. However, if it is in the current view, then it must have a reserved version in the current package.
- **CI_FILTER_NEW_ITEMS_ONLY**-This option limits the check-in to files that do not have corresponding items in the current view. If the item has been removed from the current view, then the file can be checked in using this filter.
- If the corresponding item was renamed in the current project, then the file cannot be checked in. The file must be renamed before it can be checked in.
- **CI_FILTER_EXISTING_ITEMS_ONLY**-This option limits the check-in to files that have corresponding items reserved by the package. Any files without corresponding items are skipped. This filter can be used to prevent the existence of unwanted files, such as temporary files or templates, in your repository.

Owning Class

CaCheckin

Arguments

HINT32 iOption-the Item Option

Return Value

True if success, false if fail

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); bool bResult = rCheckin.SetItemOption(CI_FILTER_NEW_OR_EXISTING_ITEMS);
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[HINT32.GetItemOption\(void\)](#) (see page 62)

bool SetClientDir(LPCTSTR szCDir)

Sets the Client Directory anchor for this process.

Note: This method is valid for file list checkin only; version list checkin does not use this method. If the checkin is from a remote machine, this directory is from the remote machine. (CA Harvest SCM stores the host location at checkout and uses these stored values for the version checkin. Version checkin may span many directories and remote machines and therefore cannot be driven by the SetClientDir location.)

Owning Class

CaCheckin

Arguments

LPCTSTR szCDir-a valid directory

Return Value

True if success, false if fail

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); bool bResult =  
rCheckin.SetClientDir("c:\\temp\\work");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[LPCTSTR GetClientDir\(void\)](#) (see page 64)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 52)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 50)

bool SetViewPath(LPCTSTR szVPath)

Sets the repository View Path anchor for this process.

Owning Class

CaCheckin

Arguments

LPCTSTR szVPath-a valid repository view path

Return Value

True if success, false if fail

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); bool bResult =  
rCheckin.SetViewPath("\\rep\\work");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 52)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[LPCTSTR GetViewPath\(void\)](#) (see page 65)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 50)

bool SetDeleteAfterCI(bool bFlag)

Sets the Delete After Checkin attribute of this process.

Owning Class

CaCheckin

Arguments

bool bFlag-the flag value: true if delete the file, false if not

Return Value

True if success, false if fail

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); bool bResult =  
rCheckin.SetDeleteAfterCI(false);
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool GetDeleteAfterCI\(void\)](#) (see page 66)

bool SetDescription(LPCTSTR szDesc)

Sets the Description to be associated with versions created by this Checkin.

Owning Class

CaCheckin

Arguments

LPCTSTR szDesc-the description

Return Value

True if success, false if fail

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); bool bResult =  
rCheckin.SetDescription("Save work prior to vacation.");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[LPCTSTR GetDescription\(void\)](#) (see page 63)

void SetRemoteMachineName(LPCTSTR szRemoteMachineName)

Sets the host name of the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, a remote username and password must also be supplied. The Client Directory must exist on the remote machine. A remote CA Harvest SCM File Agent must be running on the specified machine.

Owning Class

CaCheckin

Arguments

LPCTSTR szRemoteMachineName-the machine name

Return Value

None

Example

```
CaCheckin& rCheckin =  
rContext.GetCheckin(); rCheckin.SetRemoteMachineName("MyBuildMachine");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 52)

[LPCTSTR GetRemoteMachineName\(void\)](#) (see page 66)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 52)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 50)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 55)

void SetAgentPortNumber(LPCTSTR szPortNumber)

Sets the port number of the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine. A remote CA Harvest SCM File Agent must be running on the specified machine.

Owning Class

CaCheckin

Arguments

LPCTSTR szPortNumber-the agent port number

Return Value

None

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); rCheckin.SetAgentPortNumber("7001");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 52)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 52)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 50)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

void SetRemoteUserName(LPCTSTR szRemoteUserName)

Sets the user name for the login to the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaCheckin

Arguments

LPCTSTR szRemoteUserName

Return Value

None

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); rCheckin.SetRemoteUserName("George");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 52)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 52)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 50)

[LPCTSTR GetRemoteUserName\(void\)](#) (see page 68)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 55)

void SetRemotePassword(LPCTSTR szRemotePassword)

Sets the password for the username used to log in to the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaCheckin

Arguments

LPCTSTR szRemotePassword

Return Value

None

Example

```
CaCheckin& rCheckin =  
rContext.GetCheckin(); rCheckin.SetRemotePassword("georgespassword");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 52)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 55)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 52)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 50)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[LPCTSTR GetRemotePassword\(void\)](#) (see page 68)

void SetCallBack(PFN_CACICB pfCallBack)

Sets the callback function for the checkin function that is called after each item is checked in.

Note: PFN_CACICB is a function pointer with the following return value and argument list: void (*PFN_CACICB)(int iBaseCBReturn, CaCheckin* pCheckin, void *pUserData, CaContainer *pBag = NULL);

pBag contains data from the checkout operation. The callback function has ownership of pBag and must delete the data it points to.

Owning Class

CaCheckin

Arguments

PFN_CACICB-function pointer to callback

Return Value

None

Example

```
void cbFunction(int iBaseCBReturn, CaCheckin* pCheckin, void *pUserData, CaContainer
*pBag = NULL)
{
    CaString* pString = (CaString*)pUserData;
    CaString szViewPath = (LPCTSTR)pBag->Props(ATTR_KEY_VIEWPATH, 0);
    cout << *pString << szViewPath << endl;
}

CaCheckin& rCheckin = rContext.GetCheckin(); rCheckin.SetCallBack( cbFunction );
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[void SetCBUserData\(void* pUserData \)](#) (see page 59)

[PFN_CACICB GetCallBack\(void\);](#) (see page 74)

[void* GetCBUserData\(void\);](#) (see page 75)

void SetCBUserData(void* pUserData)

Stores the user data that will be passed to the callback function on each callback execution.

Note: The callback function takes as an argument a void*, and during execution the data set by this method is passed to the callback function through this argument. The callback function must know how to interpret this data itself.

Owning Class

CaCheckin

Arguments

void*-user defined data type that is expected by callback function

Return Value

None

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); rCheckin.SetCBUserData( (void*)
&usrData );
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[void SetCallBack\(PFN_CACICB pfCallBack \)](#) (see page 58)

[PFN_CACICB GetCallBack\(void\);](#) (see page 74)

[void* GetCBUserData\(void\);](#) (see page 75)

void SetFileList(CaStringList &FileList)

Sets the file list to be checked in by this process. If a remote machine name is set, then these files must exist on the remote machine.

Note: If a remote machine is set, then a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaCheckin

Arguments

CaStringList &FileList-a list of file names to check in

Return Value

None

Example

```
CaCheckin& rCheckin = rContext.GetCheckin();rCheckin.SetFileList(FileList);
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaStringList GetFileList\(void\)](#) (see page 69)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 52)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 52)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 50)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

[CaStringList](#) (see page 516)

bool SetNewItemOnBranch(bool bFlag)

Create versions for new items on a branch. Only one of the attributes *NewItemsOnBranch* or *NewItemsOnTrunk* can be specified as true; not both.

Owning Class

CaRemoveItem

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
rRemoveItem.SetNewItemOnBranch(true);
```

More information:

- [HSDK Programming Fundamentals](#)
- [GetNewItemOnBranch](#)
- [GetBranchOut](#)

bool SetNewItemsOnTrunk(bool bFlag)

Create versions for new items on the trunk. Only one of the attributes *NewItemsOnBranch* or *NewItemsOnTrunk* can be specified as true; not both.

Owning Class

CaRemoveItem

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
rRemoveItem.SetNewItemsOnTrunk(true);
```

More information:

- HSDK Programming Fundamentals
- GetNewItemsOnTrunk
- GetBranchOut

bool SetBranchOut(bool bFlag)

On Execute, allow the creation of versions for new items on a trunk or branch version, depending of whether *NewItemsOnBranch* or *NewItemsOnTrunk* has been specified as true.

Owning Class

CaRemoveItem

Arguments

BOOL bFlag-the flag value, true=create on branch, false=create on trunk

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
rRemoveItem.SetBranchOut(true);
```

More information:

- HSDK Programming Fundamentals
- SetNewItemOnBranch
- SetNewItemOnTrunk

HINT32 GetItemOption(void)

Gets the Item Option attribute for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

HINT32-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); HINT32 iOption =  
rCheckin.GetItemOption();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetItemOption

LPCTSTR GetDescription(void)

Gets the Description attribute for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaString szValue =  
rCheckin.GetDescription();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetDescription

HINT32 GetCheckinMode(void)

Gets the Checkin Mode attribute for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

HINT32-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); HINT32 iOption =  
rCheckin.GetCheckinMode();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCheckinMode

HINT32 GetPathOption(void)

Gets the Path Option attribute for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

HINT32-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); HINT32 iOption =  
rCheckin.GetPathOption();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetPathOption

LPCTSTR GetClientDir(void)

Note: If a remote machine name is set, then this is a directory on the remote machine.

Gets the Client Directory anchor for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaString szValue =  
rCheckin.GetClientDir();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- SetRemoteMachineName

LPCTSTR GetViewPath(void)

Gets the repository View Path anchor for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaString szValue =  
rCheckin.GetViewPath();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetViewPath

bool GetDeleteAfterCI(void)

Gets the Delete After Checkin attribute for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

bool-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); bool bFlag =  
rCheckin.GetDeleteAfterCI();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetDeleteAfterCI

LPCTSTR GetRemoteMachineName(void)

Gets remote machine name used for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaString szValue =  
rCheckin.GetRemoteMachineName();
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 55)

LPCTSTR GetAgentPortNumber(void)

Gets agent port number used for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaString szValue =  
rCheckin.GetAgentPortNumber();
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 55)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

LPCTSTR GetRemoteUserName(void)

Gets remote machine User Name used for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaString szValue =  
rCheckin.GetRemoteUserName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetAgentPortNumber
- SetRemoteUserName
- SetRemoteMachineName
- SetRemotePassword

LPCTSTR GetRemotePassword(void)

Gets remote machine Password used for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaString szValue =  
rCheckin.GetRemotePassword();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetAgentPortNumber
- SetRemotePassword
- SetRemoteUserName
- SetRemoteMachineName

CaStringList GetFileList(void)

Gets a copy of the file list to be checked in by this process. If a remote machine name is set, then these files must exist on the remote machine.

Note: This list is a copy. Any modifications will only take effect if another SetFileList function is called. If a remote machine is set, then a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaCheckin

Arguments

None

Return Value

CaStringList-a copy of the file list for checkin

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaStringList FileList =  
rCheckin.FetFileList();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetFileList
- SetClientDir
- SetRemoteMachineName
- SetViewPath
- SetPathOption
- SetRemoteUserName
- SetRemotePassword
- CaStringList

bool GetForceDescription(void) const

Gets the boolean value indicating whether a Description (set by SetDescription) is required for the CaCheckin.Execute process to succeed. The administrator sets this value.

Note: The CaCheckin process will throw an exception if GetForceDescription is true but no Description has been set for this CaCheckin process.

Owning Class

CaCheckin

Arguments

None

Return Value

bool-true if description required, false if no description required

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); bool bNeedDescription =  
rCheckin.GetForceDescription();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetDescription
- GetDescription

bool GetNewItemOnBranch(void)

Retrieves the current value for creating versions for new items on a branch.

Owning Class

CaRemoveItem

Arguments

None

Return Value

bool-current value for “CreateOnBranch” flag associated with the RemoveItem process.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
bool bCreateOnBranch = rRemoveItem.GetNewItemOnBranch();
```

More information:

- HSDK Programming Fundamentals
- GetNewItemOnBranch

bool GetNewItemOnTrunk(void)

Retrieves the current value for creating refactored versions for new items on the trunk.

Owning Class

CaRemoveItem

Arguments

None

Return Value

bool-current value for “CreateOnTrunk” flag associated with the RemoveItem process.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
bool bCreateOnTrunk = rRemoveItem.GetNewItemOnTrunk();
```

More information:

- HSDK Programming Fundamentals
- GetNewItemOnTrunk

bool GetBranchOut(void)

Retrieves the current value for whether the Execute will create refactored versions for new items on the trunk or on a branch.

Owning Class

CaRemoveItem

Arguments

None

Return Value

bool-current value for “BranchOut” flag associated with the RemoveItem process.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
bool bBranchOutOnExecute = rRemoveItem.GetBranchOut();
```

More information:

- HSDK Programming Fundamentals
- GetNewItemOnTrunk
- GetNewItemOnBranch

HINT32 CreateItemPath(LPCTSTR szPathName, LPCTSTR szParentPath)

Creates the path in the current view.

Note: To Create Item Path, the package needs to be set in the context.

Owning Class

CaCheckin

Arguments

LPCTSTR-pathname to be created

LPCTSTR-path of parent that already exists

Return Value

HINT32-result code for operation

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); /* To create 'h sdk' in '\HARVEST\src'
*/ rCheckin.CreateItemPath("h sdk", "\HARVEST\src");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetFileList
- SetClientDir
- SetRemoteMachineName
- SetViewPath
- SetPathOption
- SetRemoteUserName
- SetRemotePassword

PFN_CACICB GetCallBack(void);

Returns the callback function pointer for the checkin function that is called after each item is checked in.

Note: PFN_CACICB is a function pointer with the following return value and argument list: void (*PFN_CACICB)(int iBaseCBReturn, CaCheckin *pCheckin, void *pUserData, CaContainer *pBag = NULL);

Owning Class

CaCheckin

Arguments

None

Return Value

PFN_CACICB-function pointer to callback

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); PFN_CACICB pfCallback =  
rCheckin.GetCallBack();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCBUserData
- SetCallBack
- GetCBUserData

void* GetCBUserData(void);

Retrieves the user data that will be passed to the callback function on each callback execution.

Note: The callback function takes as an argument a void*, and during execution the data set by this method is passed to the callback function through this argument. The callback function must know how to interpret this data itself.

Owning Class

CaCheckin

Arguments

None

Return Value

void*-user defined data type that is expected by callback function

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); void*rCheckin.GetCBUserData( (void*)
&usrData );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCallBack
- GetCallBack
- GetCBUserData

HINT32 Execute(bool bFileList)

Executes the Checkin process on the Version list currently selected using the Version Chooser.

Owning Class

CaCheckin

Arguments

bool bFileList-if true, checks in a file list.; if false, gets the list from the version chooser

Default: true

Return Value

HINT32-zero if success, nonzero if failure

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); HINT32 iResult =
rCheckin.Execute(false);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaVersionChooser
- CaStringList

CaCheckout

CaCheckout encapsulates the Checkout process. It cannot be instantiated directly. The Checkout process can be set in the CaContext object either by name or object ID. The CaCheckout object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, Package, and Checkout Context.
- Select a set of Versions to Checkout using the CaVersionChooser.
- Invoke the Execute function.

Public Methods

- SetCheckoutMode
- SetPathOption
- SetReplaceFile
- SetClientDir
- SetViewPath
- SetShareWorkDir
- SetUseCITimeStamp
- SetRemoteMachineName
- SetAgentPortNumber
- SetRemoteUserName
- SetRemotePassword
- SetMVSFileTypes
- GetReplaceFile
- GetShareWorkDir
- GetCheckoutMode
- GetPathOption
- GetClientDir
- GetViewPath
- GetBrowseMode
- GetConcUpdateMode
- GetSyncMode

- GetUpdateMode
- GetUseCITimeStamp
- GetRemoteMachineName
- GetAgentPortNumber
- GetRemoteUserName
- GetRemotePassword
- GetMVSFileTypes
- SetCallBack
- SetCBUserData
- GetCallBack
- GetCBUserData
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext
- CaVersionChooser
- CaStringList

bool SetCheckoutMode(HINT32 iMode)

Sets the Checkout Mode for this process.

Note: The following are the allowable values for the Checkout Mode. They are defined in an include file built into the project.

- CO_MODE_BROWSE-Check out the versions for read only. No lock is set.
- CO_MODE_UPDATE-Check out the versions for reserve.
- CO_MODE_RESERVE_ONLY-Reserve the versions but do not write them to the file system. Set a lock.
- CO_MODE_SYNCHRONIZE-Check out selected versions for read only but only if they are newer than the ones previously checked out to the file system. No lock is set.
- CO_MODE_CONCURRENT-Check out the versions for concurrent update and still allow other reservations with other packages.

Owning Class

CaCheckout

Arguments

HINT32 iMode-the Checkout Mode

Return Value

True if success, false if fail

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bResult =  
rCheckout.SetCheckoutMode(CO_MODE_BROWSE);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- GetCheckoutMode

bool SetPathOption(HINT32 iOption)

Sets the Path Option for this process. Files are checked out to the directory structure in locations related to the view path, client directory, and the path option.

Note: The following are the allowable values for the Path Option. They are defined in an include file built into the project.

- CO_OPTION_PRESERVE-Check out each new version to its existing client path location that mirrors the location in the view path.
- CO_OPTION_PRESERVE_AND_CREATE-Check out each new version to its client path location that mirrors the location in the view path and create the client path if it does not yet exist.
- CO_OPTION_ALL_ITEM_TO_SAME_DIRECTORY-Check out each new version to the single client path location specified.

Owning Class

CaCheckout

Arguments

HINT32 iOption-the Path Option.

Return Value

True if success, false if fail

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bResult =  
rCheckout.SetPathOption (CO_OPTION_PRESERVE_AND_CREATE);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- GetPathOption
- SetClientDir
- SetViewPath

bool SetReplaceFile(bool bFlag)

Sets Replace File attribute for this process.

Owning Class

CaCheckout

Arguments

bool bFlag-if true, then read only files are replaced; if false, they are not

Return Value

True if success, false if fail

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bResult =  
rCheckout.SetReplaceFile(true);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- GetReplaceFile

bool SetClientDir(LPCTSTR szCDir)

Sets the Client Directory anchor for this process.

Note: If the Checkout is from a remote machine, then this directory is from the remote machine.

Owning Class

CaCheckout

Arguments

LPCTSTR szCDir-a valid directory

Return Value

True if success, false if fail

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bResult =  
rCheckout.SetClientDir("c:\\temp\\work");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- GetClientDir
- SetRemoteMachineName
- SetViewPath
- SetPathOption

bool SetViewPath(LPCTSTR szVPath)

Sets the repository View Path anchor for this process.

Owning Class

CaCheckout

Arguments

LPCTSTR szVPath-a valid repository view path

Return Value

True if success, false if fail

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bResult =  
rCheckout.SetViewPath("\\rep\\work");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- SetRemoteMachineName
- GetViewPath
- SetPathOption

bool SetShareWorkDir(bool bFlag)

Sets the Delete After Checkout attribute of this process.

Note: This option is for UNIX and Linux file systems. The file access is set such that group has access.

Owning Class

CaCheckout

Arguments

bool bFlag-the flag value: true if the client directory is shared, false if not

Return Value

True if success, false if fail

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bResult =  
rCheckout.SetShareWorkDir(false);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- GetShareWorkDir

bool SetUseCITimeStamp(bool bFlag)

Sets the Use Checkin Timestamp attribute to be used by this Checkout.

Owning Class

CaCheckout

Arguments

bool bFlag-if true, sets time on file to time version was checked in; if false, sets it to current time

Return Value

True if success, false if fail

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bResult =  
rCheckout.SetUseCITimeStamp(true);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- GetUseCITimeStamp

void SetRemoteMachineName(LPCTSTR szRemoteMachineName)

Sets the host name of the remote machine where the files are to be located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine. A remote CA Harvest SCM File Agent must be running on the specified machine.

Owning Class

CaCheckout

Arguments

LPCTSTR szRemoteMachineName-the machine name

Return Value

None

Example

```
CaCheckout& rCheckout =  
rContext.GetCheckout(); rCheckout.SetRemoteMachineName("MyBuildMachine");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- GetRemoteMachineName
- SetViewPath
- SetPathOption
- SetAgentPortNumber
- SetRemoteUserName
- SetRemotePassword

void SetAgentPortNumber(LPCTSTR szPortNumber)

Sets the port number of the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, a remote username and password must also be supplied. The Client Directory must exist on the remote machine. A remote CA Harvest SCM File Agent must be running on the specified machine.

Owning Class

CaCheckout

Arguments

LPCTSTR szPortNumber-the agent port number

Return Value

None

Example

```
CaCheckout& rCheckout =  
rContext.GetCheckout(); rCheckout.SetAgentPortNumber("7001");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 52)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 52)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 50)

[LPCTSTR GetAgentPortNumber\(void\)](#) (see page 67)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

void SetRemoteUserName(LPCTSTR szRemoteUserName)

Sets the user name for the login to the remote machine where the files are to be located.

Note: If a remote machine is set, then an agent port number, a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaCheckout

Arguments

LPCTSTR szRemoteUserName

Return Value

None

Example

```
CaCheckout& rCheckout =  
rContext.GetCheckout(); rCheckout.SetRemoteUserName("George");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- SetRemoteMachineName
- SetAgentPortNumber
- SetViewPath
- SetPathOption
- GetRemoteUserName
- SetRemotePassword

void SetRemotePassword(LPCTSTR szRemotePassword)

Sets the password for the username used to login to the remote machine where the files are to be located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaCheckout

Arguments

LPCTSTR szRemotePassword

Return Value

None

Example

```
CaCheckout& rCheckout =  
rContext.GetCheckout(); rCheckout.SetRemotePassword("georgespassword");
```

More information:

- [HSDK Programming Fundamentals](#)
- [CaContext](#)
- [SetClientDir](#)
- [SetRemoteMachineName](#)
- [SetViewPath](#)
- [SetPathOption](#)
- [SetRemoteUserName](#)
- [GetRemotePassword](#)

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 81)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 83)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 84)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 81)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 79)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 85)

[LPCTSTR GetRemotePassword\(void\)](#) (see page 97)

bool SetMVSFileTypes(HINT32 iOption)

Sets the MVS File Types attribute to be used by this process. This is only used if the remote file system is on the MVS platform.

Note: The following are the allowable values for the Checkout Mode. They are defined in an include file built into the project.

- HAR_MVS_PDS-the target file location is a PDS
- HAR_MVS_SEQUENTIAL-the target file location is a Sequential file

The remote machine name must be set to an MVS platform for this option to be used. The Client Directory must exist on the remote machine.

Owning Class

CaCheckout

Arguments

HINT32 iOption-the MVS File Types Option

Return Value

True if success, false if fail

Example

```
CaCheckout& rCheckout = rContext.GetCheckout();  
rCheckout.SetMVSFileTypes(HAR_MVS_PDS);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- GetMVSFileTypes
- SetClientDir
- SetRemoteMachineName
- SetViewPath
- SetPathOption
- SetRemoteUserName
- SetRemotePassword

HINT32 GetCheckoutMode(void)

Gets the Checkout Mode attribute for this process.

Owning Class

CaCheckout

Arguments

None

Return Value

HINT32-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); HINT32 iOption =  
rCheckout.GetCheckoutMode();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCheckoutMode

HINT32 GetPathOption(void)

Gets the Path Option attribute for this process.

Owning Class

CaCheckout

Arguments

None

Return Value

HINT32-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); HINT32 iOption =  
rCheckout.GetPathOption();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetPathOption

LPCTSTR GetClientDir(void)

Gets the Client Directory anchor for this process.

Note: If a remote machine name is set, then this is a directory on the remote machine.

Owning Class

CaCheckout

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); CaString szValue =  
rCheckout.GetClientDir();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- SetRemoteMachineName

LPCTSTR GetViewPath(void)

Gets the repository View Path anchor for this process.

Owning Class

CaCheckout

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); CaString szValue =  
rCheckout.GetViewPath();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetViewPath

bool GetBrowseMode(void)

True if browse mode is available in this process.

Owning Class

CaCheckout

Arguments

None

Return Value

bool-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bBrowseSupported =  
rCheckout.GetBrowseMode();
```

More information:

- HSDK Programming Fundamentals
- CaContext

bool GetConcUpdateMode(void)

True if concurrent update mode is available in this process.

Owning Class

CaCheckout

Arguments

None

Return Value

bool-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bCUSupported =  
rCheckout.GetConcUpdateMode();
```

More information:

- HSDK Programming Fundamentals
- CaContext

bool GetSyncMode(void)

True if synchronize mode is available in this process.

Owning Class

CaCheckout

Arguments

None

Return Value

bool-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bSyncSupported =  
rCheckout.GetSyncMode();
```

More information:

- HSDK Programming Fundamentals
- CaContext

bool GetUpdateMode(void)

True if update mode is available in this process.

Owning Class

CaCheckout

Arguments

None

Return Value

bool-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bUpdateSupported =  
rCheckout.GetConcUpdateMode();
```

More information:

- HSDK Programming Fundamentals
- CaContext

bool GetReplaceFile(void)

Gets the Replace File attribute for this process.

Owning Class

CaCheckout

Arguments

None

Return Value

bool-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bOption =  
rCheckout.GetReplaceFile();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetReplaceFile

bool GetShareWorkDir(void)

Gets the Share Working Directory attribute for this process.

Owning Class

CaCheckout

Arguments

None

Return Value

bool-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bFlag =  
rCheckout.GetShareWorkDir();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetShareWorkDir

bool GetUseCITimeStamp(void)

Gets the Use Checkin Timestamp attribute for this process.

Owning Class

CaCheckout

Arguments

None

Return Value

bool-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); bool bFlag =  
rCheckout.GetUseCITimeStamp();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetUseCITimeStamp

LPCTSTR GetRemoteMachineName(void)

Gets remote machine name used for this process.

Owning Class

CaCheckout

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); CaString szValue =  
rCheckout.GetRemoteMachineName();
```

More information:

[CaContext](#) (see page 106)

[HSDK Programming Fundamentals](#) (see page 36)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 83)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 84)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 85)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 86)

LPCTSTR GetAgentPortNumber(void)

Gets agent port number used for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaString szValue =  
rCheckin.GetAgentPortNumber();
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 55)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

LPCTSTR GetRemoteUserName(void)

Gets remote machine User Name used for this process.

Owning Class

CaCheckout

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); CaString szValue =  
rCheckout.GetRemoteUserName();
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 294)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 85)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 86)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 55)

LPCTSTR GetRemotePassword(void)

Gets remote machine Password used for this process.

Owning Class

CaCheckout

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); CaString szValue =  
rCheckout.GetRemotePassword();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetAgentPortNumber
- SetRemotePassword
- SetRemoteUserName
- SetRemoteMachineName

HINT32 GetMVSFileTypes(void)

Gets the MVSFile Types attribute used by this process.

Note: This attribute is only used if the remote machine name is set to an MVS platform.

Owning Class

CaCheckout

Arguments

None

Return Value

HINT32-the value

Example

```
CaCheckout& rCheckout = rContext.GetCheckout();HINT32 iValue =  
rCheckout.GetMVSFileTypes();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetMVSFileTypes
- SetRemoteMachineName

void SetCallBack(PFN_CACOCB pfCallBack)

Sets the callback function for the checkout function that is called after each item is checked in.

Note: PFN_CACOCB is a function pointer with the following return value and argument list: void (*PFN_CACOCB)(int iBaseCBReturn, CaCheckout* pCheckout, void *pUserData, CaContainer *pBag = NULL);

pBag contains data from checkout operation. The callback function has ownership of pBag and must delete the data it points to.

Owning Class

CaCheckout

Arguments

PFN_CACOCB-function pointer to callback

Return Value

None

Example

```
void cbFunction(int iBaseCBReturn, CaCheckout* pCheckout, void *pUserData,
CaContainer *pBag = NULL)
{
    CaString* pString = (CaString*)pUserData;
    CaString szViewPath = (LPCTSTR)pBag->Props(ATTR_KEY_VIEWPATH, 0);
    cout << *pString << szViewPath << endl;
}

CaCheckout& rCheckout = rContext.GetCheckout(); rCheckout.SetCallBack(
    cbFunction );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCBUserData
- GetCallBack
- GetCBUserData

void SetCBUserData(void* pUserData)

Stores the user data that will be passed to the callback function on each callback execution.

Note: The callback function takes as an argument a void*, and during execution the data set by this method is passed to the callback function through this argument. The callback function must know how to interpret this data itself.

Owning Class

CaCheckout

Arguments

void*-user-defined data type that is expected by callback function

Return Value

None

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); rCheckout.SetCBUserData( (void*)
&usrData );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCallBack
- GetCallBack
- GetCBUserData

PFN_CACOCB GetCallBack(void);

Returns the callback function pointer for the checkout function that is called after each item is checked in.

Note: PFN_CACOCB is a function pointer with the following return value and argument list: void (*PFN_CACOCB)(int iBaseCBReturn, CaCheckout *pCheckout, void *pUserData, CaContainer *pBag = NULL);

Owning Class

CaCheckout

Arguments

None

Return Value

PFN_CACOCB-function pointer to callback

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); PFN_CACOCB pfCallback =  
rCheckout.GetCallBack();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCBUserData
- SetCallBack
- GetCBUserData

void* GetCBUserData(void);

Retrieves the user data that will be passed to the callback function on each callback execution.

Note: The callback function takes as an argument a void*, and during execution the data set by this method is passed to the callback function through this argument. The callback function must know how to interpret this data itself.

Owning Class

CaCheckout

Arguments

None

Return Value

void*-user-defined data type that is expected by callback function

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); void*rCheckout.GetCBUserData(  
(void*) &usrData );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCallBack
- GetCallBack
- GetCBUserData

HINT32 Execute(void)

Executes the Checkout process on the Version list currently selected using the Version Chooser.

Owning Class

CaCheckout

Arguments

None

Return Value

HINT32-zero if success, nonzero if failure

Example

```
CaCheckout& rCheckout = rContext.GetCheckout(); HINT32 iResult =  
rCheckout.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaVersionChooser

CaConcurrentMerge

CaConcurrentMerge encapsulates the ConcurrentMerge process. It cannot be instantiated directly. The ConcurrentMerge process can be set in the CaContext object either by name or object ID. The CaConcurrentMerge object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, Package, and ConcurrentMerge Context.
- Concurrent Merge operates only on the Package in the current context.
- Set the ConcurrentMerge attributes.
- Invoke the Execute function.

Public Methods

- SetMergeRule
- SetVersionDescription
- GetMergeRule
- GetVersionDescription
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

void SetMergeRule(LPCTSTR szRule)

Sets Rule for how to merge the versions of files in the package.

Note: The following are the allowable values for the Merge Rule. They are defined in an include file built into the project.

- MERGE_RULE_TRUNK_MERGE-Take the entire trunk version and ignore the version on the branch. This corresponds to the GUI TakeTrunk Version option.
- MERGE_RULE_BRANCH_MERGE-Take the entire branch version and ignore the version on the trunk. This corresponds to the GUI Take Branch Version option.
- MERGE_RULE_CONFLICT_TAG-Create a merge tag only if changes in branch and trunk conflict. This corresponds to the GUI Merge Aggressively option.
- MERGE_RULE_ALWAYS_TAG-Create a merge tag if there is any difference in the trunk and branch version at all. This corresponds to the GUI Merge Conservatively option.

Owning Class

CaConcurrentMerge

Arguments

LPCTSTR szRule-the rule value

Return Value

None

Example

```
CaConcurrentMerge& rConcurrentMerge = rContext.GetConcurrentMerge();  
rConcurrentMerge.SetMergeRule(MERGE_RULE_CONFLICT_TAG);
```

More information:

- HSDK Programming Fundamentals
- GetMergeRule

void SetVersionDescription(LPCTSTR szDescription)

Provides a description to be associated with the merged version.

Owning Class

CaConcurrentMerge

Arguments

LPCTSTR szDescription

Return Value

None

Example

```
CaConcurrentMerge& rConcurrentMerge = rContext.GetConcurrentMerge();  
rConcurrentMerge.SetVersionDescription("My Merged Version");
```

More information:

- HSDK Programming Fundamentals
- GetVersionDescription

LPCTSTR GetMergeRule(void)

Gets the current value of the Merge Rule Option.

Owning Class

CaConcurrentMerge

Arguments

None

Return Value

LPCTSTR-current value of option

Example

```
CaConcurrentMerge& rConcurrentMerge = rContext.GetConcurrentMerge(); CaString =  
rConcurrentMerge.GetMergeRule();
```

■ More information:

- HSDK Programming Fundamentals
- SetMergeRule

LPCTSTR GetVersionDescription(void)

Gets the current value of the Version Description.

Owning Class

CaConcurrentMerge

Arguments

None

Return Value

LPCTSTR-current value of description

Example

```
CaConcurrentMerge& rConcurrentMerge = rContext.GetConcurrentMerge(); CaString =  
rConcurrentMerge.GetVersionDescription();
```

More information:

- HSDK Programming Fundamentals
- SetVersionDescription

HINT32 Execute(bUseCurrentContext)

Executes the ConcurrentMerge process on the Package in the current context.

Note: Default value is true.

Owning Class

CaConcurrentMerge

Arguments

bool bUseCurrentContext-if true, restrict concurrent merge to current context package; if false, use packages selected in the Package Chooser

Return Value

HINT32-zero if success, nonzero if failure

Example

```
CaConcurrentMerge& rConcurrentMerge = rContext.GetConcurrentMerge(); HINT32 iResult  
= rConcurrentMerge.Execute(false);
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaContext

CaContext is a smart container that manages the current CA Harvest SCM context. If a Project is reset then the State is reset. If the State is reset then the View, Package, and all Processes are reset. State, View, and Process lists are retrieved based on the current context. A Package chooser and a Version chooser allow filtered selection based on the context.

The CaContext Object has no public constructor. One copy is instantiated and maintained by the CaHarvest object.

Public Methods

- GetProjectName
- GetProjectId
- GetStateName
- GetStatId
- GetProcessId
- GetPackageName

- GetPackageld
- GetViewName
- GetViewId
- GetViewPathId
- GetSnapshotViewName
- GetSnapshotViewId
- GetCheckoutName
- GetPromoteName
- GetDemoteName
- GetSnapshotViewName
- GetCheckout
- GetCheckin
- GetPromote
- GetDemote
- GetClientPath
- GetViewPath
- GetSQL
- GetViewPathItem
- GetItemById
- GetItemPathInView
- GetVersionById
- GetItemList
- GetPackageChooser
- GetVersionChooser
- GetStateList
- SetProject
- SetState
- SetCheckout
- SetCheckin
- SetPromote
- SetDemote
- SetPackage

- SetSnapshotView
- SetClientPath
- SetViewPath
- SetViewPathbyId
- GetApproveName
- GetApprove
- SetApprove
- GetCreatePackageName
- GetCreatePackage
- SetCreatePackage
- SetWebserverName
- GetDeleteVersionName
- GetDeleteVersion
- SetDeleteVersion
- GetRemoveItemName
- GetRemoveItem
- SetRemoveItemName
- GetRenameItemName
- GetRenameItem
- SetRenameItemName
- GetConcurrentMergeName
- GetConcurrentMerge
- SetConcurrentMergeName
- SetCrossProjectMerge
- GetCrossProjectMerge
- GetCrossProjectMergeName
- SetInteractiveMerge
- GetInteractiveMerge
- GetInteractiveMergeName
- SetTakeSnapshot
- GetTakeSnapshot
- GetTakeSnapshotName

- SetUDP
- GetUDP
- GetUDPName
- GetUserName
- GetWebserverName
- IsPackageSet
- GetMoveItemName
- GetMoveItem
- SetMoveItem
- GetMovePathName
- GetMovePath
- SetMovePath
- GetRemovePathName
- GetRemovePath
- SetRemovePath
- GetRenamePathName
- GetRenamePath
- SetRenamePath

More information:

- HSDK Programming Fundamentals
- CaException
- CaHarvest
- CaContainer
- CaPromote
- CaCheckout
- CaPackageChooser
- CaVersionChooser
- CaDemote CaApprove
- CaCreatePackage
- CaDeleteVersion
- CaRemoveItem
- CaConcurrentMerge

LPCTSTR GetProjectName(void) const

Retrieve the name of the current Project.

Owning Class

CaContext

Arguments

None

Return Value

Project name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetProjectName();
```

■ More information:

- HSDK Programming Fundamentals
- SetProject
- GetProjectId

HAROBJID GetProjectId(void) const

Retrieve the Object Id of the current Project.

Owning Class

CaContext

Arguments

None

Return Value

Project Object Id if set, NULL_HAROBJID if empty.

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); HAROBJID iObjId =  
rContext.GetProjectId();
```

■ More information:

- HSDK Programming Fundamentals
- SetProject
- GetProjectName

LPCTSTR GetStateName(void) const

Retrieve the name of the current State.

Owning Class

CaContext

Arguments

None

Return Value

State name if set, empty string if empty.

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetStateName();
```

■ More information:

- HSDK Programming Fundamentals
- SetState
- GetStateId

HAROBJID GetStateId(void) const

Retrieve the Object Id of the current State.

Owning Class

CaContext

Arguments

None

Return Value

State Object Id if set, NULL_HAROBJID if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); HAROBJID iObjId =  
rContext.GetStateId();
```

■ More information:

- HSDK Programming Fundamentals
- SetState
- GetStateName

HAROBJID GetProcessId(void) const

Retrieves the object ID of the process.

Owning Class

CaContext

Arguments

None

Example

```
CaHarvest Harvest;  
CaContext& rContext = Harvest.GetContext();  
HAROBJID iProcessId;  
if (rContext.IsProcessSet(HAR_CHECKOUT_PROCESS_TYPE))  
    iProcessId = rContext.GetProcessId(HAR_CHECKOUT_PROCESS_TYPE);
```

The valid process types defined are the following:

- HAR_APPROVE_PROCESS_TYPE = 200
- HAR_CHECKIN_PROCESS_TYPE = 201
- HAR_CHECKOUT_PROCESS_TYPE = 202
- HAR_DELETEVERSIONS_PROCESS_TYPE = 203
- HAR_DEMOTE_PROCESS_TYPE = 204
- HAR_LISTDIFFERENCE_PROCESS_TYPE = 205
- HAR_LISTVERSION_PROCESS_TYPE = 206
- HAR_CONCURRENT_MERGE_PROCESS_TYPE = 207
- HAR_CROSSPROJECT_MERGE_PROCESS_TYPE = 208
- HAR_INTERACTIVE_MERGE_PROCESS_TYPE = 209
- HAR_NOTIFY_PROCESS_TYPE = 210
- HAR_CREATE_PACKAGE_PROCESS_TYPE = 211
- HAR_MOVE_PACKAGE_PROCESS_TYPE = 212
- HAR_PROMOTE_PROCESS_TYPE = 213
- HAR_REMOVEITEM_PROCESS_TYPE = 214
- HAR_TAKESNAPSHOT_PROCESS_TYPE = 215

- HAR_UDP_PROCESS_TYPE = 216
- HAR_RenameItem_PROCESS_TYPE = 217
- HAR_SWITCH_PACKAGE_PROCESS_TYPE = 218

More information:

- HSDK Programming Fundamentals

LPCTSTR GetPackageName(void) const

Retrieves the name of the current Package.

Owning Class

CaContext

Arguments

None

Return Value

Package name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetPackageName();
```

More information:

- HSDK Programming Fundamentals
- SetPackage
- GetPackageId

HAROBJID GetPackageId(void) const

Retrieves the Object Id of the current Package.

Owning Class

CaContext

Arguments

None

Return Value

Package Object Id if set, NULL_HAROBJID if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); HAROBJID iObjId =  
rContext.GetPackageId();
```

■ More information:

- HSDK Programming Fundamentals
- SetPackage
- GetPackageName

HAROBJID GetViewName(void) const

Retrieves the object ID of the current View.

Owning Class

CaContext

Arguments

None

Return Value

View object ID if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); HAROBJID iId =  
rContext.GetViewId();
```

More information:

- HSDK Programming Fundamentals
- SetState

LPCTSTR GetViewName(void) const

Retrieves the name of the current View.

Owning Class

CaContext

Arguments

None

Return Value

View name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetViewName();
```

More information:

- HSDK Programming Fundamentals
- SetState

HAROBJID GetViewPathId(void) const

Retrieve the object ID of the viewpath.

Owning Class

CaContext

Arguments

None

Example

```
CaHarvest Harvest;  
CaContext& rContext = Harvest.GetContext();  
bool bResult = rContext.SetViewPath("\\rep\\work");  
if ( bResult)  
    HAROBJID iViewPath = rContext.GetViewPathId();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetSnapshotViewName(void) const

Retrieves the name of the current SnapshotView.

Owning Class

CaContext

Arguments

None

Return Value

SnapshotView name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetSnapshotViewName();
```

■ More information:

- HSDK Programming Fundamentals
- SetSnapshotView
- GetSnapshotViewId

HAROBJID GetSnapshotViewId(void) const

Retrieves the object ID of the current SnapshotView.

Owning Class

CaContext

Arguments

None

Return Value

SnapshotView ID if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); HAROBJID iSnapshot =  
rContext.GetSnapshotViewId();
```

■ More information:

- HSDK Programming Fundamentals
- SetSnapshotView
- GetSnapshotViewName

LPCTSTR GetCheckoutName(void) const

Retrieve the name of the current Checkout.

Owning Class

CaContext

Arguments

None

Return Value

Checkout name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetCheckoutName();
```

■ More information:

- HSDK Programming Fundamentals
- SetCheckout
- GetCheckout

LPCTSTR GetCheckinName(void) const

Retrieves the name of the current Checkin.

Owning Class

CaContext

Arguments

None

Return Value

Checkin name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetCheckinName();
```

■ More information:

- HSDK Programming Fundamentals
- SetCheckin
- GetCheckin

LPCTSTR GetPromoteName(void) const

Retrieves the name of the current Promote.

Owning Class

CaContext

Arguments

None

Return Value

Promote name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetPromoteName();
```

■ More information:

- HSDK Programming Fundamentals
- SetPromote
- GetPromote

LPCTSTR GetDemoteName(void) const

Retrieves the name of the current Demote.

Owning Class

CaContext

Arguments

None

Return Value

Demote name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetDemoteName();
```

■ More information:

- HSDK Programming Fundamentals
- SetDemote
- GetDemote

LPCTSTR GetApproveName(void) const

Retrieves the name of the current Approve.

Owning Class

CaContext

Arguments

None

Return Value

Approve name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetApproveName();
```

■ More information:

- HSDK Programming Fundamentals
- SetApprove
- GetApprove

LPCTSTR GetCreatePackageName(void) const

Retrieves the name of the current CreatePackage.

Owning Class

CaContext

Arguments

None

Return Value

CreatePackage name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetCreatePackageName();
```

■ More information:

- HSDK Programming Fundamentals
- SetCreatePackage
- GetCreatePackage

LPCTSTR GetDeleteVersionName(void) const

Retrieves the name of the current DeleteVersion.

Owning Class

CaContext

Arguments

None

Return Value

DeleteVersion name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetDeleteVersionName();
```

■ More information:

- HSDK Programming Fundamentals
- SetDeleteVersion
- GetDeleteVersion

LPCTSTR GetRemoveItemName(void) const

Retrieves the name of the current RemoveItem.

Owning Class

CaContext

Arguments

None

Return Value

RemoveItem name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetRemoveItemName();
```

■ More information:

- HSDK Programming Fundamentals
- SetRemoveItem
- GetRemoveItem

LPCTSTR GetConcurrentMergeName(void) const

Retrieves the name of the current ConcurrentMerge.

Owning Class

CaContext

Arguments

None

Return Value

ConcurrentMerge name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetConcurrentMergeName();
```

■ More information:

- HSDK Programming Fundamentals
- SetConcurrentMerge
- GetConcurrentMerge

LPCTSTR GetCrossProjectMergeName(void) const

Retrieves the name of the current CrossProjectMerge.

Owning Class

CaContext

Arguments

None

Return Value

CrossProjectMerge name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetCrossProjectMergeName();
```

More information:

- HSDK Programming Fundamentals
- SetCrossProjectMerge
- GetCrossProjectMerge

LPCTSTR GetTakeSnapshotName(void) const

Retrieves the name of the current TakeSnapshot.

Owning Class

CaContext

Arguments

None

Return Value

TakeSnapshot name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetTakeSnapshotName();
```

■ More information:

- HSDK Programming Fundamentals
- SetTakeSnapshot
- GetTakeSnapshot

LPCTSTR GetUDPName(void) const

Retrieves the name of the current UDP.

Owning Class

CaContext

Arguments

None

Return Value

UDP name if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetUDPName();
```

■ More information:

- HSDK Programming Fundamentals
- SetUDP
- GetUDP

LPCTSTR GetRenameItemName(void) const

Retrieves the name of the current RenameItem.

Owning Class

CaContext

Arguments

None

Return Value

RenameItem process name if set, empty string if empty

Example

```
CaHarvest Harvest;  
CaContext& rContext = Harvest.GetContext();  
CaString szName = rContext.GetRenameItemName();
```

More information:

- HSDK Programming Fundamentals
- SetRenameItem
- GetRenameItem

LPCTSTR GetMoveItemName(void) const

Retrieves the name of the current MoveItem.

Owning Class

CaContext

Arguments

None

Return Value

MoveItem process name if set, empty string if empty

Example

```
CaHarvest Harvest;  
CaContext& rContext = Harvest.GetContext();  
CaString szName = rContext.GetMoveItemName();
```

More information:

- HSDK Programming Fundamentals
- SetMoveItem
- GetMoveItem

LPCTSTR GetMovePathName(void) const

Retrieves the name of the current MovePath.

Owning Class

CaContext

Arguments

None

Return Value

MovePath process name if set, empty string if empty

Example

```
CaHarvest Harvest;  
CaContext& rContext = Harvest.GetContext();  
CaString szName = rContext.GetMovePathName();
```

More information:

- HSDK Programming Fundamentals
- SetMovePath
- GetMovePath

LPCTSTR GetRemovePathName(void) const

Retrieves the name of the current RemovePath.

Owning Class

CaContext

Arguments

None

Return Value

RemovePath process name if set, empty string if empty

Example

```
CaHarvest Harvest;  
CaContext& rContext = Harvest.GetContext();  
CaString szName = rContext.GetRemovePathName();
```

More information:

- HSDK Programming Fundamentals
- SetRemovePath
- GetRemovePath

LPCTSTR GetRenamePathName(void) const

Retrieves the name of the current RenamePath.

Owning Class

CaContext

Arguments

None

Return Value

RenamePath process name if set, empty string if empty

Example

```
CaHarvest Harvest;  
CaContext& rContext = Harvest.GetContext();  
CaString szName = rContext.GetRenamePathName();
```

More information:

- HSDK Programming Fundamentals
- SetRenamePath
- GetRenamePath

LPCTSTR GetUserName(void) const

Retrieves the name of the current CA Harvest SCM user.

Owning Class

CaContext

Arguments

None

Return Value

User name of currently logged in user

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szName =  
rContext.GetUserName();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetWebserverName(void)

Retrieves the name of the current Web Interface server.

Owning Class

CaContext

Arguments

None

Return Value

Server name of Web Interface server

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szWebServer  
= rContext.GetWebserverName();
```

More information:

- HSDK Programming Fundamentals

HAROBJID GetUserId(void) const

Retrieves the object ID of the current CA Harvest SCM user.

Owning Class

CaContext

Arguments

None

Return Value

Object ID of currently logged in user

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); HAROBJID iUser =  
rContext.GetUserId();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetClientPath(void) const

Retrieves the current CA Harvest SCM client path.

Owning Class

CaContext

Arguments

None

Return Value

Client path if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szPath =  
rContext.GetClientPath();
```

■ More information:

- HSDK Programming Fundamentals
- SetClientPath

LPCTSTR GetViewPath(void) const

Retrieves the current View path.

Owning Class

CaContext

Arguments

None

Return Value

View path if set, empty string if empty

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaString szPath =  
rContext.GetViewPath();
```

■ More information:

- HSDK Programming Fundamentals
- SetViewPath

LPCTSTR GetInteractiveMergeName(void) const

Retrieves the name of the current InteractiveMerge

Owning Class

CaContext

Arguments

None

Return Value

InteractiveMerge name if set, empty string if empty

Example

```
CaHarvest Harvest;  
CaContext& rContext = Harvest.GetContext();  
CaString szName = rContext.GetInteractiveMergeName();
```

More information:

- HSDK Programming Fundamentals
- SetInteractiveMerge
- GetInteractiveMerge

CaItem& GetViewPathItem(void)

Retrieves the view path represented as a CA Harvest SCM item.

Note: The context, including the state and viewpath, must be set before calling this function.

Owning Class

CaContext

Arguments

None

Return Value

View path item.

Exceptions

Throws CaException on error

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaItem& rItem =  
rContext.GetViewPathItem();
```

More information:

- HSDK Programming Fundamentals
- SetState SetViewPath

CaItem* GetItemById(HAROBJID iItemId)

Retrieves the item specified by the provided ID.

Note: This object is owned by whoever creates it and must be deleted by its owner.

Owning Class

None

Arguments

HAROBJID iItemId

Return Value

The item with ID iItemId

Exceptions

Throws CaException on error.

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaItem *pItem =  
rContext.GetItemById(32912);
```

- **More information:**
- HSDK Programming Fundamentals
- Caltem

CaItem* GetItemPathInView(LPCTSTR szFullPath)

Retrieves the item path specified by the provided path string.

Note: This object is owned by whoever creates it and must be deleted by its owner.

Owning Class

None

Arguments

LPCTSTR szFullPath

Return Value

The item path

Exceptions

Throws CaException on error.

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaItem* pItem =  
rContext.GetItemPathInView( _T ( "\\a\\b\\c" ));
```

More information:

- HSDK Programming Fundamentals
- Caltem

CaVersion* GetVersionById(HAROBJID iVersionId)

Retrieves the version specified by the provided id.

Note: This object is owned by whoever creates it and must be deleted by its owner.

Owning Class

None

Arguments

HAROBJID iVersionId

Return Value

The version with ID iVersionId

Exceptions

Throws CaException on error.

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaVersion *pVersion  
= rContext.GetVersionById(12345);
```

More information:

- HSDK Programming Fundamentals
- CaVersion

CaItemList& GetItemList(bool bRefresh, HINT32 iItemPathOption)

Retrieves list of items that are children of the current view path item.

Owning Class

CaContext

Arguments

bool bRefresh = false, HINT32 iItemPathOption = ITEMS_AND_PATHS_IN_VIEW

Return Value

The item list

Exceptions

Throws CaException on error.

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext(); CaItemList& rList =  
rContext.GetItemList(true, ITEMS_AND_PATHS_IN_VIEW);
```

More information:

- HSDK Programming Fundamentals
- Caltem
- CaltemList

CaCheckout& GetCheckout(void) const

Retrieves the current Checkout object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current Checkout object, if set

Exceptions

Throws CaException on error.

Example

```
CaCheckout& rCheckout = rContext.GetCheckout();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetCheckout
- GetCheckoutName

CaCheckin& GetCheckin(void) const

Retrieves the current Checkin object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current Checkin object, if set

Exceptions

Throws CaException on error.

Example

```
CaCheckin& rCheckin = rContext.GetCheckin();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetCheckin
- GetCheckinName

CaPromote& GetPromote(void) const

Retrieves the current Promote object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current Promote object, if set

Exceptions

Throws CaException on error.

Example

```
CaPromote& rPromote = rContext.GetPromote();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetPromote
- GetPromoteName

CaDemote& GetDemote(void) const

Retrieves the current Demote object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current Demote object, if set

Exceptions

Throws CaException on error.

Example

```
CaDemote& rDemote = rContext.GetDemote();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetDemote
- GetDemoteName

CaApprove& GetApprove(void) const

Retrieves the current Approve object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current Approve object, if set

Exceptions

Throws CaException on error.

Example

```
CaApprove& rApprove = rContext.GetApprove();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetApprove
- GetApproveName

CaCreatePackage& GetCreatePackage(void) const

Retrieve the current CreatePackage object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current CreatePackage object, if set

Exceptions

Throws CaException on error.

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetCreatePackage
- GetCreatePackageName

CaDeleteVersion& GetDeleteVersion(void) const

Retrieves the current DeleteVersion object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current DeleteVersion object, if set

Exceptions

Throws CaException on error.

Example

```
CaDeleteVersion& rDeleteVersion = rContext.GetDeleteVersion();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetDeleteVersion
- GetDeleteVersionName

CaRemoveItem& GetRemoveItem(void) const

Retrieves the current RemoveItem object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current RemoveItem object, if set

Exceptions

Throws CaException on error.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetRemoveItem
- GetRemoveItemName

CaConcurrentMerge& GetConcurrentMerge(void) const

Retrieves the current ConcurrentMerge object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current ConcurrentMerge object, if set

Exceptions

Throws CaException on error.

Example

```
CaConcurrentMerge& rConcurrentMerge = rContext.GetConcurrentMerge();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetConcurrentMerge
- GetConcurrentMergeName

CaCrossProjectMerge& GetCrossProjectMerge(void) const

Retrieves the current CrossProjectMerge object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current CrossProjectMerge object, if set

Exceptions

Throws CaException on error.

Example

```
CaCrossProjectMerge& rCrossProjectMerge = rContext.GetCrossProjectMerge();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetCrossProjectMerge
- GetCrossProjectProjectMergeName

CaInteractiveMerge& GetInteractiveMerge(void) const

Retrieves the current InteractiveMerge object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current InteractiveMerge object, if set

Exceptions

Throws CaException on error

Example

```
CaInteractiveMerge& rInteractiveMerge = rContext.GetInteractiveMerge();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetInteractiveMerge
- GetInteractiveMergeName

CaTakeSnapshot& GetTakeSnapshot(void) const

Retrieves the current TakeSnapshot object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current TakeSnapshot object, if set

Exceptions

Throws CaException on error.

Example

```
CaTakeSnapshot& rTakeSnapshot = rContext.GetTakeSnapshot();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetTakeSnapshot
- GetTakeSnapshotName

CaUDP& GetUDP(void) const

Retrieves the current UDP object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current UDP object, if set

Exceptions

Throws CaException on error.

Example

```
CaUDP& rUDP = rContext.GetUDP();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetUDP
- GetUDPName

CaRenameItem& GetRenameItem(void) const

Retrieves the current RenameItem object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current RenameItem object, if set

Exceptions

Throws CaException on error.

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetRenameItem
- GetRenameItemName

CaMoveItem& GetMoveItem(void) const

Retrieves the current MoveItem object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current MoveItem object, if set

Exceptions

Throws CaException on error.

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetMoveItem
- GetMoveItemName

CaMovePath& GetMovePath(void) const

Retrieves the current MovePath object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current MovePath object, if set

Exceptions

Throws CaException on error.

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetMovePath
- GetMovePathName

CaRemovePath& GetRemovePath(void) const

Retrieves the current RemovePath object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current RemovePath object, if set

Exceptions

Throws CaException on error.

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetRemovePath
- GetRemovePathName

CaRenamePath& GetRenamePath(void) const

Retrieves the current RenamePath object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the current RenamePath object, if set

Exceptions

Throws CaException on error.

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetRenamePath
- GetRenamePathName

CaSQL& GetSQL(void) const

Retrieves the SQL object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the SQL object

Exceptions

Throws CaException on error.

Example

```
CaSQL& rSQL = rContext.GetSQL();
```

More information:

- HSDK Programming Fundamentals
- CaException

CaPackageChooser& GetPackageChooser(void) const

Retrieves the PackageChooser object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the PackageChooser object

Exceptions

Throws CaException on error.

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();
```

More information:

- HSDK Programming Fundamentals
- CaException

CaVersionChooser& GetVersionChooser(void) const

Retrieves the VersionChooser object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the VersionChooser object

Exceptions

Throws CaException on error.

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();
```

More information:

- HSDK Programming Fundamentals
- CaException

CaHarvest& GetHarvestSession (void) const

Retrieves the current CaHarvest object.

Owning Class

CaContext

Arguments

None

Return Value

Reference to the CaHarvest object

Exceptions

Throws CaException on error.

Example

```
CaHarvest& rHS = rContext.GetHarvestSession();
```

More information:

- HSDK Programming Fundamentals
- CaException

CaContainer GetStateList(void) const

Retrieves a CaContainer object that contains the attributes for each State in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME
- CA_ATTRKEY_PROJECTID
- CA_ATTRKEY_SNAPSHOT
- CA_ATTRKEY_VIEW_NAME
- CA_ATTRKEY_VIEWID

Owning Class

CaContext

Arguments

None

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = rContext.GetStateList ();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaContainer GetViewList(void) const

Retrieves a CaContainer object that contains the attributes for each View in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_PROJECTID
- CA_ATTRKEY_VIEW_TYPE

Owning Class

CaContext

Arguments

None

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = rContext.GetViewList ();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaContainer GetProcessList(void) const

Retrieves a CaContainer object that contains the attributes for each Process in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_PROJECTID
- CA_ATTRKEY_STATEID
- CA_ATTRKEY_PROCTYPE
- CA_ATTRKEY_EXECUTEACCESS

Owning Class

CaContext

Arguments

None

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = rContext.GetProcessList ();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

bool SetProject(LPCTSTR szProjectName)

Sets the specified Project as the current context Project.

Owning Class

CaContext

Arguments

LPCTSTR szProjectName

Return Value

True if success, false if fails

Example

```
rContext.SetProject(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetProjectName
- GetProject
- CaHarvest

bool SetProject(HAROBJID iObjId)

Sets the specified Project as the current context Project.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetProject(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetProjectName
- GetProject
- CaHarvest

bool SetState(LPCTSTR szStateName)

Sets the specified State as the current context State.

Owning Class

CaContext

Arguments

LPCTSTR szStateName

Return Value

True if success, false if fails

Example

```
rContext.SetState(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetStateName
- GetState
- GetStatId

bool SetState(HAROBJID iObjId)

Sets the specified State as the current context State.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetState(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetStateName
- GetState
- GetStatId

bool SetCheckout(LPCTSTR szCheckoutName)

Sets the specified Checkout as the current context Checkout.

Owning Class

CaContext

Arguments

LPCTSTR szCheckoutName

Return Value

True if success, false if fails

Example

```
rContext.SetCheckout(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetCheckoutName
- GetCheckout
- GetCheckoutId

bool SetCheckout(HAROBJID iObjId)

Sets the specified Checkout as the current context Checkout.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetCheckout(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetCheckoutName
- GetCheckout
- GetCheckoutId

bool SetCheckin(LPCTSTR szCheckinName)

Sets the specified Checkin as the current context Checkin.

Owning Class

CaContext

Arguments

LPCTSTR szCheckinName

Return Value

True if success, false if fails

Example

```
rContext.SetCheckin(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetCheckinName
- GetCheckin
- GetCheckinId

bool SetCheckin(HAROBJID iObjId)

Sets the specified Checkin as the current context Checkin.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetCheckin(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetCheckinName
- GetCheckin
- GetCheckinId

bool SetInteractiveMerge(LPCTSTR szInteractiveMergeName)

Sets the specified InteractiveMerge as the current context InteractiveMerge.

Owning Class

CaContext

Arguments

LPCTSTR szInteractiveMergeName

Return Value

True if success, false if fails

Example

```
rContext.SetInteractiveMerge(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetInteractiveMergeName
- GetInteractiveMerge

bool SetInteractiveMerge(HAROBJID iObjId)

Sets the specified InteractiveMerge as the current context InteractiveMerge.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetConcurrentMerge(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetInteractiveMergeName
- GetInteractiveMerge

bool SetPromote(LPCTSTR szPromoteName)

Sets the specified Promote as the current context Promote.

Owning Class

CaContext

Arguments

LPCTSTR szPromoteName

Return Value

True if success, false if fails

Example

```
rContext.SetPromote(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetPromoteName
- GetPromote
- GetPromoteId

bool SetPromote(HAROBJID iObjId)

Sets the specified Promote as the current context Promote.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetPromote(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetPromoteName
- GetPromote
- GetPromoteId

bool SetDemote(LPCTSTR szDemoteName)

Sets the specified Demote as the current context Demote.

Owning Class

CaContext

Arguments

LPCTSTR szDemoteName

Return Value

True if success, false if fails

Example

```
rContext.SetDemote(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetDemoteName
- GetDemote
- GetDemotId

bool SetDemote(HAROBJID iObjId)

Sets the specified Demote as the current context Demote.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetDemote(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetDemoteName
- GetDemote
- GetDemoteld

bool SetCrossProjectMerge(LPCTSTR szCrossProjectMergeName)

Sets the specified CrossProjectMerge as the current context CrossProjectMerge.

Owning Class

CaContext

Arguments

LPCTSTR szCrossProjectMerge

Return Value

True if success, false if fails

Example

```
rContext.CrossProjectMerge(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetCrossProjectMergeName
- GetCrossProjectMerge

bool SetCrossProjectMerge(HAROBJID iObjId)

Sets the specified CrossProjectMerge as the current context CrossProjectMerge.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.CrossProjectMerge( iObjId );
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetCrossProjectMergeName
- GetCrossProjectMerge

bool SetTakeSnapshot(LPCTSTR szTakeSnapshotName)

Sets the specified TakeSnapshot as the current context TakeSnapshot.

Owning Class

CaContext

Arguments

LPCTSTR szTakeSnapshotName

Return Value

True if success, false if fails

Example

```
rContext.SetTakeSnapshot(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetTakeSnapshotName
- GetTakeSnapshot

bool SetTakeSnapshot(HAROBJID iObjId)

Sets the specified TakeSnapshot as the current context TakeSnapshot.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetTakeSnapshot( iObjId );
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetTakeSnapshotName
- GetTakeSnapshot

bool SetUDP(LPCTSTR szUDPName)

Sets the specified UDP as the current context UDP.

Owning Class

CaContext

Arguments

LPCTSTR szUDPName

Return Value

True if success, false if fails

Example

```
rContext.SetUDP(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetUDPName
- GetUDP

bool SetUDP(HAROBJID iObjId)

Sets the specified UDP as the current context UDP.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetUDP( iObjId );
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetUDPName
- GetUDP

bool SetApprove(LPCTSTR szApproveName)

Sets the specified Approve as the current context Approve.

Owning Class

CaContext

Arguments

LPCTSTR szApproveName

Return Value

True if success, false if fails

Example

```
rContext.SetApprove(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetApproveName
- GetApprove
- GetApproveId

bool SetApprove(HAROBJID iObjId)

Sets the specified Approve as the current context Approve.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetApprove(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetApproveName
- GetApprove
- GetApproveId

void SetWebserverName(LPCTSTR szWebserverName)

Stores the address of the Web Interface server in the current context.

Owning Class

CaContext

Arguments

LPCTSTR szWebserverName

Return Value

None

Example

```
rContext.SetWebserverName(szWebserverName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetWebserverName

bool SetCreatePackage(LPCTSTR szCreatePackageName)

Sets the specified CreatePackage as the current context CreatePackage.

Owning Class

CaContext

Arguments

LPCTSTR szCreatePackageName

Return Value

True if success, false if fails

Example

```
rContext.SetCreatePackage(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetCreatePackageName
- GetCreatePackage
- GetCreatePackageld

bool SetCreatePackage(HAROBJID iObjId)

Sets the specified CreatePackage as the current context CreatePackage.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetCreatePackage(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetCreatePackageName
- GetCreatePackage
- GetCreatePackageld

bool SetDeleteVersion(LPCTSTR szDeleteVersionName)

Sets the specified DeleteVersion as the current context DeleteVersion.

Owning Class

CaContext

Arguments

LPCTSTR szDeleteVersionName

Return Value

True if success, false if fails

Example

```
rContext.SetDeleteVersion(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetDeleteVersionName
- GetDeleteVersion
- GetDeleteVersionId

bool SetDeleteVersion(HAROBJID iObjId)

Sets the specified DeleteVersion as the current context DeleteVersion.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetDeleteVersion(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetDeleteVersionName
- GetDeleteVersion
- GetDeleteVersionId

bool SetRemoveItem(LPCTSTR szRemoveItemName)

Sets the specified RemoveItem as the current context RemoveItem.

Owning Class

CaContext

Arguments

LPCTSTR szRemoveItemName

Return Value

True if success, false if fails

Example

```
rContext.SetRemoveItem(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetRemoveItemName
- GetRemoveItem
- GetRemoveItemId

bool SetRemoveItem(HAROBJID iObjId)

Sets the specified RemoveItem as the current context RemoveItem.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetRemoveItem(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetRemoveItemName
- GetRemoveItem
- GetRemoveItemId

bool SetConcurrentMerge(LPCTSTR szConcurrentMergeName)

Sets the specified ConcurrentMerge as the current context ConcurrentMerge.

Owning Class

CaContext

Arguments

LPCTSTR szConcurrentMergeName

Return Value

True if success, false if fails

Example

```
rContext.SetConcurrentMerge(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetConcurrentMergeName
- GetConcurrentMerge
- GetConcurrentMergeld

bool SetConcurrentMerge(HAROBJID iObjId)

Sets the specified ConcurrentMerge as the current context ConcurrentMerge.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetConcurrentMerge(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetConcurrentMergeName
- GetConcurrentMerge
- GetConcurrentMergeld

bool SetRenameItem(LPCTSTR szRenameItemName)

Sets the specified RenameItem as the current context RenameItem.

Owning Class

CaContext

Arguments

LPCTSTR szRenameItemName

Return Value

True if success, false if fails

Example

```
rContext.SetRenameItem(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetRenameItemName
- GetRenameItem

bool SetRenameItem(HAROBJID iObjId)

Sets the specified RenameItem as the current context RenameItem.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetRenameItem( iObjId );
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetRenameItemName
- GetRenameItem

bool SetMoveItem(LPCTSTR szMoveItemName)

Sets the specified MoveItem as the current context MoveItem.

Owning Class

CaContext

Arguments

LPCTSTR szMoveItemName

Return Value

True if success, false if fails

Example

```
rContext.SetMoveItem(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetMoveItemName
- GetMoveItem

bool SetMoveItem(HAROBJID iObjId)

Sets the specified MoveItem as the current context MoveItem.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetMoveItem( iObjId );
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetMoveItemName
- GetMoveItem

bool SetMovePath(LPCTSTR szMovePathName)

Sets the specified MovePath as the current context MovePath.

Owning Class

CaContext

Arguments

LPCTSTR szMovePathName

Return Value

True if success, false if fails

Example

```
rContext.SetMovePath(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetMovePathName
- GetMovePath

bool SetMovePath(HAROBJID iObjId)

Sets the specified MovePath as the current context MovePath.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetMovePath( iObjId );
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetMovePathName
- GetMovePath

bool SetRemovePath(LPCTSTR szRemovePathName)

Sets the specified RemovePath as the current context RemovePath.

Owning Class

CaContext

Arguments

LPCTSTR szRemovePathName

Return Value

True if success, false if fails

Example

```
rContext.SetRemovePath(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetRemovePathName
- GetRemovePath

bool SetRemovePath(HAROBJID iObjId)

Sets the specified RemovePath as the current context RemovePath.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetRemovePath( iObjId );
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetRemovePathName
- GetRemovePath

bool SetRenamePath(LPCTSTR szRenamePathName)

Sets the specified RenamePath as the current context RenamePath.

Owning Class

CaContext

Arguments

LPCTSTR szRenamePathName

Return Value

True if success, false if fails

Example

```
rContext.SetRenamePath(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetRenamePathName
- GetRenamePath

bool SetRenamePath(HAROBJID iObjId)

Sets the specified RenamePath as the current context RenamePath.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetRenamePath( iObjId );
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetRenamePathName
- GetRenamePath

bool SetPackage(LPCTSTR szPackageName)

Sets the specified Package as the current context Package.

Owning Class

CaContext

Arguments

LPCTSTR szPackageName

Return Value

True if success, false if fails

Example

```
rContext.SetPackage(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetPackageName
- GetPackage
- GetPackageld

bool SetPackage(HAROBJID iObjId)

Sets the specified Package as the current context Package.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

True if success, false if fails

Example

```
rContext.SetPackage(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetPackageName
- GetPackage
- GetPackageld

bool SetSnapshotView(LPCTSTR szSnapshotViewName)

Sets the specified SnapshotView as the current context SnapshotView.

Owning Class

CaContext

Arguments

LPCTSTR szSnapshotViewName

Return Value

True if success, false if fails

Example

```
rContext.SetSnapshotView(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetSnapshotViewName
- GetSnapshotViewId

bool SetSnapshotView(HAROBJID iObjid)

Sets the specified SnapshotView as the current context SnapshotView.

Owning Class

CaContext

Arguments

HAROBJID iObjid

Return Value

True if success, false if fails

Example

```
rContext.SetSnapshotView( iObjId );
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetSnapshotViewName
- GetSnapshotViewId

void SetClientPath(LPCTSTR szClientPath)

Sets the specified Client Path as the current context Client Path.

Owning Class

CaContext

Arguments

LPCTSTR szClientPath

Return Value

None

Exceptions

Throws CaException on error.

Example

```
rContext.SetClientPath(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetClientPath

void SetViewPath(LPCTSTR szViewPath)

Sets the specified View Path as the current context View Path.

Owning Class

CaContext

Arguments

LPCTSTR szViewPath

Return Value

None

Exceptions

Throws CaException on error.

Example

```
rContext.SetViewPath(szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- GetViewPath

void SetViewPathById(HAROBJID iObjId)

Sets the specified View Path as the current View Path.

Owning Class

CaContext

Arguments

HAROBJID iObjId

Return Value

None

Example

```
CaHarvest Harvest;  
CaContext& rContext = Harvest.GetContext();  
rContext.SetViewPathById(iObjId);
```

More information:

- HSDK Programming Fundamentals

bool IsPackageSet()

Determines if the context currently has a package set.

Owning Class

CaContext

Arguments

None

Return Value

bool-true if package set, false if not set

Exceptions

Throws CaException on error.

Example

```
bool bResult = rContext.IsPackageSet();
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetPackage
- GetPackage

CaCreatePackage

CaCreatePackage encapsulates the CreateChangePackage process. It cannot be instantiated directly. The CreatePackage process can be set in the CaContext object either by name or object ID. The CaCreatePackage object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

A Create Package process has a default name expression to generate the package name if none is supplied. The GetDefaultName() function returns the expression. If no expression then the default is "Package- number" where number is a sequential integer value.

The GetPackageName() function returns the caller specified name prior to an Execute call and the resulting package name after an Execute call. The name must be reset if additional Execute calls are to be made.

Usage

- Set the Project, State, and CreatePackage Context.
- Set the CreatePackage attributes.
- Invoke the Execute function.

Public Methods

- SetPackageName
- SetAssignedUserName
- SetDescription
- SetPriority
- GetPackageName
- GetAssignedUserName
- GetPriority
- GetDescription
- GetDefaultName
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

void SetPackageName(LPCTSTR szPackageName)

Sets a name for the new Package.

Note: If the package name attribute is set, then it is used without regard to the default name expression.

Owning Class

CaCreatePackage

Arguments

LPCTSTR szPackageName

Return Value

None

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage(); rCreatePackage.  
SetPackageName("Package-12");
```

- **More information:**

- HSDK Programming Fundamentals
- GetPackageName

void SetAssignedUserName(LPCTSTR szAssignedUserName)

Assigns a user to the new Package.

Owning Class

CaCreatePackage

Arguments

LPCTSTR szAssignedUserName-user name for the assigned user

Return Value

None

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage(); rCreatePackage.  
SetAssignedUserName ("George");
```

- **More information:**

- HSDK Programming Fundamentals
- GetAssignedUserName

void SetDescription(LPCTSTR szDescription)

Sets a description for the new package.

Owning Class

CaCreatePackage

Arguments

LPCTSTR szDescription.

Return Value

None

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage(); rCreatePackage.  
SetDescription("A good package");
```

■ More information:

- HSDK Programming Fundamentals
- GetDescription

void SetPriority(HINT32 iPriority)

Sets the Package priority attribute.

Note: Priorities are integers but the meaning of each value is user defined.

Owning Class

CaCreatePackage

Arguments

HINT32 iPriority-priorities value

Return Value

None

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage(); rCreatePackage.  
SetPriority (1);
```

More information:

- HSDK Programming Fundamentals
- GetPriority

LPCTSTR GetDefaultName (void)

Retrieves the default name expression for the new package.

Note: The CA Harvest SCM administrator sets the default name expression as an attribute of the Create Package process. The default name expression generates the package name if none is supplied with the SetPackageName() function. If there is no default expression then the default is “Package- number” where number is a sequential integer value.

Owning Class

CaCreatePackage

Arguments

None

Return Value

LPCTSTR-the name expression

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage();CaString szName =  
rCreatePackage. GetDefaultName();
```

■ More information:

- HSDK Programming Fundamentals
- SetPackageName
- GetDefaultName

LPCTSTR GetPackageName(void)

Retrieves the name of the new package.

Owning Class

CaCreatePackage

Arguments

None

Return Value

LPCTSTR-the name

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage();CaString szName =  
rCreatePackage. GetPackageName(true);
```

More information:

- HSDK Programming Fundamentals
- SetPackageName
- GetDefaultName

LPCTSTR GetAssignedUserName(void)

Retrieves the current value of the Assigned User Name attribute.

Owning Class

CaCreatePackage

Arguments

None

Return Value

LPCTSTR-current Assigned User Name value

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage();CaString szName =  
rCreatePackage. GetAssignedUserName();
```

More information:

- HSDK Programming Fundamentals
- SetItemName

LPCTSTR GetDescription(void)

Retrieves the current value of the description attribute.

Owning Class

CaCreatePackage

Arguments

None

Return Value

LPCTSTR-current description value

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage();CaString  
szDescription = rCreatePackage. GetDescription();
```

■ More information:

- HSDK Programming Fundamentals
- SetDescription

HINT32 GetPriority(void)

Retrieves the current value of the Priority attribute.

Owning Class

CaCreatePackage

Arguments

None

Return Value

HINT32- current Priority value

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage();HINT32 iPriority =  
rCreatePackage.GetPriority();
```

■ More information:

- HSDK Programming Fundamentals
- SetPriority

HINT32 Execute(void)

Executes the CreatePackage process on the Package list currently selected using the Package Chooser.

Owning Class

CaCreatePackage

Arguments

None

Return Value

HINT32-zero if success, nonzero if failure

Example

```
CaCreatePackage& rCreatePackage = rContext.GetCreatePackage(); HINT32 iResult =  
rCreatePackage.Execute();
```

■ More information:

- HSDK Programming Fundamentals
- CaContext

CaCrossProjectMerge

CaCrossProjectMerge encapsulates the ExecuteUDP process. It cannot be instantiated directly. The ExecuteUDP process can be set in the CaContext object either by name or object ID. The CaCrossProjectMerge object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, and ExecuteUDP Context.
- Set the ExecuteUDP attributes.
- Invoke the Execute function.

Public Methods

- SetFromState
- SetMergeRule
- SetPlacement
- SetVersionDescription
- GetFromProjectId
- GetFromProjectName
- GetFromStateId
- GetFromStateName
- GetMergeRule
- GetPlacement

- GetVersionDescription
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

void SetFromState(LPCTSTR szProjectName, LPCTSTR szStateName)

Sets the Project and State that contain the source packages to merge with. These must differ from the Project and State set in the Context.

Note: A list of available projects may be retrieved from a CaHarvest object using GetProjectList(). From that list, select a project by name and extract its ObjId. Then, using the CaHarvest object's GetStateList(ObjId) function, retrieve a list of available states in that project. The name or objid can then be used in SetFromState.

Owning Class

CaCrossPackageMerge

Arguments

- LPCTSTR szProjectName-name of the project to merge with
- LPCTSTR szStateName-name of a state in szProjectName that contains the desired packages

Return Value

None

Example

```
CaCrossProjectMerge& xprojmerge = CaContext.GetCrossProjectMerge();  
xprojmerge.SetFromState( szProject, szState );
```

More information:

- HSDK Programming Fundamentals
- CaHarvest
- GetProjectList
- GetStateList

void SetFromState(HAROBJID iProjectId, HAROBJID iStateId)

Sets the Project and State that contain the source packages to merge with. These must differ from the Project and State set in the Context.

Note: A list of available projects may be retrieved from a CaHarvest object using GetProjectList(). From that list, select a project by name and extract its ObjId. Then, using the CaHarvest object's GetStateList(ObjId) function, retrieve a list of available states in that project. The name or objid can then be used in SetFromState.

Owning Class

CaCrossPackageMerge

Arguments

- HAROBJID iProjectId - ID of the project to merge with
- HAROBJID iStateId - ID of a state in iProjectId that contains the desired packages

Return Value

None

Example

```
CaContainer bag = h.GetProjectList(); CaDataArray a=bag.FindData(CA_ATTRKEY_NAME,
(LPCTSTR)gszMergeProject ); HAROBJID iProject =
(HINT32)bag.Props(CA_ATTRKEY_OBJID,a[0]); bag = h.GetStateList( iProject ); a =
bag.FindData(CA_ATTRKEY_NAME, (LPCTSTR)gszMergeState ); HAROBJID iState =
(HINT32)bag.Props(CA_ATTRKEY_OBJID,a[0]); CaCrossProjectMerge& xprojmerge =
CaContext.GetCrossProjectMerge(); xprojmerge.SetFromState( iProject, iState);
```

More information:

- HSDK Programming Fundamentals
- CaHarvest
- GetProjectList
- GetStateList

void SetMergeRule(LPCTSTR szRule)

Sets Rule for how to merge the versions of files in the packages.

Note: The following are the allowable values for the Merge Rule. They are defined in an include file built into the project.

- MERGE_RULE_TRUNK_MERGE-Take the entire trunk version and ignore the version on the branch. This corresponds to the GUI TakeTrunk Version option.
- MERGE_RULE_BRANCH_MERGE-Take the entire branch version and ignore the version on the trunk. This corresponds to the GUI Take Branch Version option.
- MERGE_RULE_CONFLICT_TAG-Create a merge tag only if changes in branch and trunk conflict. This corresponds to the GUI Merge Aggressively option.
- MERGE_RULE_ALWAYS_TAG-Create a merge tag if there is any difference in the trunk and branch version at all. This corresponds to the GUI Merge Conservatively option.

Owning Class

CaCrossPackageMerge

Arguments

LPCTSTR szRule-the rule value

Return Value

None

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge();  
xprojmrg.SetMergeRule("MERGE_RULE_CONFLICT_TAG");
```

More information:

- HSDK Programming Fundamentals
- GetMergeRule

void SetPlacement(LPCTSTR szPlacement)

Sets Rule for how the merged version appears in the destination state.

Note: The following are the allowable values for the Placement. They are defined in an include file built into the project.

- MERGE_OPTION_BRANCH_ONLY-Creates a merge version on the target branch. This allows changes to be copied from the source project to the target project even if one or more target items are reserved for updates in the main trunk. With this option, a branch will be created to store the changes.
- MERGE_OPTION_TRUNK_ONLY-Creates a merge version on the target trunk.
- MERGE_OPTION_TRUNK_OR_BRANCH-Creates a merge version on the target trunk or branch. This allows changes to be copied from the source project to the target project even if one or more target items are reserved for update in the main trunk. If items are reserved for update on the trunk, a branch will be created to store the changes. If items are not reserved for update on the trunk, the items are simply copied to the trunk.

Owning Class

CaCrossPackageMerge

Arguments

LPCTSTR szRule-the rule value

Return Value

None

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge();  
xprojmrg.SetPlacement("MERGE_OPTION_TRUNK_ONLY");
```

More information:

- HSDK Programming Fundamentals
- GetPlacement

void SetVersionDescription(LPCTSTR szDescription)

Sets text description of merged version.

Owning Class

CaCrossPackageMerge

Arguments

LPCTSTR szDescription-the description itself

Return Value

None

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge();  
xprojmrg.SetVersionDescription("Version description");
```

■ More information:

- HSDK Programming Fundamentals
- GetVersionDescription

HAROBJID GetFromProjectId(void)

Returns the ObjID of the Project set by SetFromState.

Owning Class

CaCrossPackageMerge

Arguments

None

Return Value

HAROBJID-Project Id containing source state and packages

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge(); HAROBJID iProject  
= xprojmrg.GetFromProjectId();
```

■ More information:

- HSDK Programming Fundamentals
- SetFromState
- CaHarvest

LPCTSTR GetFromProjectName(void)

Returns the Name of the Project set by SetFromState.

Owning Class

CaCrossPackageMerge

Arguments

None

Return Value

LPCTSTR-Project Name containing source state and packages

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge(); LPCTSTR  
szProjectName = xprojmrg.GetFromProjectName();
```

More information:

- HSDK Programming Fundamentals
- SetFromState
- CaHarvest

HAROBJID GetFromStateId(void)

Returns the ObjID of the State set by SetFromState.

Owning Class

CaCrossPackageMerge

Arguments

None

Return Value

HAROBJID-State Id containing source state and packages

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge(); HAROBJID  
iFromState = xprojmrg.GetFromStateId();
```

More information:

- HSDK Programming Fundamentals
- SetFromState
- CaHarvest

LPCTSTR GetFromStateName(void)

Returns the Name of the State set by SetFromState.

Owning Class

CaCrossPackageMerge

Arguments

None

Return Value

LPCTSTR-State Name containing source state and packages

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge(); LPCTSTR  
szFromState = xprojmrg.GetFromStateName();
```

More information:

- HSDK Programming Fundamentals
- SetFromState
- CaHarvest

LPCTSTR GetMergeRule(void)

Gets Rule for how to merge the versions of files in the packages.

Note: The following are the allowable values for the Merge Rule. They are defined in an include file built into the project.

- MERGE_RULE_TRUNK_MERGE-Take the entire trunk version and ignore the version on the branch. This corresponds to the GUI TakeTrunk Version option.
- MERGE_RULE_BRANCH_MERGE-Take the entire branch version and ignore the version on the trunk. This corresponds to the GUI Take Branch Version option.
- MERGE_RULE_CONFLICT_TAG-Create a merge tag only if changes in branch and trunk conflict. This corresponds to the GUI Merge Aggressively option.
- MERGE_RULE_ALWAYS_TAG-Create a merge tag if there is any difference in the trunk and branch version at all. This corresponds to the GUI Merge Conservatively option.

Owning Class

CaCrossPackageMerge

Arguments

None

Return Value

LPCTSTR szRule-the rule value

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge(); LPCTSTR  
szMergeRule = xprojmrg.GetMergeRule();
```

More information:

- HSDK Programming Fundamentals
- SetMergeRule

LPCTSTR GetPlacement(void)

Gets Rule for how the merged version appears in the destination state.

Note: The following are the allowable values for the Placement. They are defined in an include file built into the project.

- MERGE_OPTION_BRANCH_ONLY-Creates a merge version on the target branch. This allows changes to be copied from the source project to the target project even if one or more target items are reserved for updates in the main trunk. With this option, a branch will be created to store the changes.
- MERGE_OPTION_TRUNK_ONLY-Creates a merge version on the target trunk.
- MERGE_OPTION_TRUNK_OR_BRANCH-Creates a merge version on the target trunk or branch. This allows changes to be copied from the source project to the target project even if one or more target items are reserved for update in the main trunk. If items are reserved for update on the trunk, a branch will be created to store the changes. If items are not reserved for update on the trunk, the items are simply copied to the trunk.

Owning Class

CaCrossPackageMerge

Arguments

None

Return Value

LPCTSTR szRule-the rule value

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge(); LPCTSTR  
szPlacement = xprojmrg.GetPlacement();
```

More information:

- HSDK Programming Fundamentals
- SetPlacement

LPCTSTR GetVersionDescription(void)

Gets text description of merged version.

Owning Class

CaCrossPackageMerge

Arguments

None

Return Value

LPCTSTR szDescription-the description itself

Example

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge(); LPCTSTR  
szDescription = xprojmrg.GetVersionDescription("Version description");
```

More information:

- HSDK Programming Fundamentals
- GetVersionDescription

HINT32 Execute(void)

Executes the CrossProjectMerge process on the Package in the current context with the Packages in the Package Chooser.

Note: Packages in Package Chooser must be from the Project and State specified by SetFromState(). This is accomplished by setting the Project and State. The CaPackageChooser will search with CaPackageChooser::SetProjectObjId(iProject) and CaPackageChooser::SetStateObjId(iState), and then CaPackageChooser::Execute (false) to allow searching outside of context.

Owning Class

CaCrossPackageMerge

Arguments

None

Return Value

HINT32-zero if success, nonzero if failure

Example

```
CaCrossProjectMerge& rConcurrentMerge = rContext.GetConcurrentMerge(); HINT32  
iResult = rConcurrentMerge.Execute();
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageChooser
- SetProjectObjId
- SetStatId
- CaHarvest
- GetProjectList
- GetStateList

CaDeletePackage

CaDeletePackage encapsulates the DeletePackage process. It cannot be instantiated directly. The DeletePackage process can be set in the CaContext object either by name or object ID. The CaDeletePackage object can only be retrieved from the CaContext after it has been set. Set the package either by object ID or by name and path.

Usage

- Set the Project, State, and DeletePackage Context.
- Set the DeletePackage attributes, including the package to be deleted.
- Invoke the Execute function.

Public Methods

- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

HINT32 Execute(LPCTSTR SourcePackage)

Executes the DeletePackage process.

Owning Class

CaDeletePackage

Arguments

SourcePackage-the source package for the delete package process

Return Value

HINT32-zero if success, non-zero if failure

Example

```
CaDeletePackage& rDeletePackage = rContext.GetDeletePackage();  
CaString szName = rDeletePackage. GetPackageName(true);  
HINT32 iResult = rDeletePackage.Execute(szName);
```

More information:

- HSDK Programming Fundamentals
- CaContext

HINT32 Execute(HAROBJID SourcePackage)

Executes the DeletePackage process.

Owning Class

CaDeletePackage

Arguments

SourcePackage-the source package ID for the delete package process

Return Value

HINT32-zero if success, non-zero if failure

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext();  
HAROBJID iObjId = rContext.GetPackageId();  
HINT32 iResult = rDeletePackage.Execute(iObjId);
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaDeleteVersion

CaDeleteVersion encapsulates the DeleteVersion process. It cannot be instantiated directly. The DeleteVersion process can be set in the CaContext object either by name or object ID. The CaDeleteVersion object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, and DeleteVersion Context
- Select a set of Versions to delete using the CaVersionChooser object
- Invoke the Execute function.

Public Methods

- Execute

More information:

- HSDK Programming Fundamentals
- CaContext
- CaVersionChooser

HINT32 Execute(void)

Executes the DeleteVersion process on the Package list currently selected using the Package Chooser.

Owning Class

CaDeleteVersion

Arguments

None

Return Value

HINT32-zero if success, nonzero if failure

Example

```
CaDeleteVersion& rDeleteVersion = rContext.GetDeleteVersion(); HINT32 iResult =  
rDeleteVersion.Execute();
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- CaVersionChooser

CaDemote

CaDemote encapsulates the Demote process. It cannot be instantiated directly. The Demote process can be set in the CaContext object either by name or object ID. The CaDemote object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, and Demote Context.
- Select a set of Change Packages to Demote using the CaPackageChooser object.
- Set the Demote attributes.
- Invoke the Execute function.

Public Methods

- SetCheckPackageDependencies
- SetCheckPackageGroupBind

- GetCheckPackageDependencies
- GetCheckPackageGroupBind
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageChooser

bool SetCheckPackageDependencies(bool bCheckPackageDependencies)

Sets Check Package Dependencies flag.

Owning Class

CaDemote

Arguments

bool bCheckPackageDependencies-true if verify dependencies prior to Demote, false if not

Return Value

bool-true if set succeeded, false if failed

Example

```
CaDemote& rDemote = rContext.GetDemote(); rDemote. SetCheckPackageDependencies  
(true);
```

More information:

- HSDK Programming Fundamentals
- GetCheckPackageDependencies

bool SetCheckPackageGroupBind(bool bFlag)

Specifies whether or not packages in a package group must all be demoted together.

Owning Class

CaDemote

Arguments

bool bFlag-true if yes, false if no

Return Value

bool-true if set succeeded, false if failed

Example

```
CaDemote& rDemote = rContext.GetDemote(); bool bFlag = rDemote.  
SetCheckPackageGroupBind(true);
```

More information:

- HSDK Programming Fundamentals
- GetCheckPackageGroupBind

bool GetCheckPackageDependencies (void)

Gets the current value of the Check Package Dependencies flag.

Owning Class

CaDemote

Arguments

None

Return Value

bool-current value of flag

Example

```
CaDemote& rDemote = rContext.GetDemote(); CaString = rDemote.  
GetCheckPackageDependencies ();
```

More information:

- HSDK Programming Fundamentals
- SetCheckPackageDependencies

bool GetCheckPackageGroupBind (void)

Gets the current value of the Check Package Group Bind flag.

Owning Class

CaDemote

Arguments

None

Return Value

bool-current value of flag

Example

```
CaDemote& rDemote = rContext.GetDemote(); CaString = rDemote.  
GetCheckPackageGroupBind ();
```

More information:

- HSDK Programming Fundamentals
- SetCheckPackageGroupBind

HINT32 Execute(void)

Executes the Demote process on the Package list currently selected using the Package Chooser.

Owning Class

CaDemote

Arguments

None

Return Value

HINT32-zero if success, nonzero if failure

Example

```
CaDemote& rDemote = rContext.GetDemote(); HINT32 iResult = rDemote.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageChooser

CaException

CaException provides keyed message generation and logging facilities. It uses the underlying CA Harvest SCM Error Message Handler to retrieve and/or log formatted message text passed into one of the constructors. Messages can be logged to the static ostream established with the CaHarvest::SetLog() function. The default constructor is declared private to ensure that a message is provided.

The Error Message Handler reads an input file that contains message format strings into a CaContainer. Each message in the file has a unique textual key, an error number, and a format string with an optional set of positional arguments. Message files can therefore be translated to other languages without affecting the code.

The CaException objects can be constructed with either a message key or with a literal text message depending on the constructor used.

Public Methods

- CaException
- GetReturnCode
- SetReturnCode
- GetMessage
- LogMessage

More information:

- HSDK Programming Fundamentals
- CaException
- CaHarvest
- CaContainer

CaException(bool bKey, LPCTSTR szMessage)

Constructs an exception object. The bKey argument determines whether szMessage is treated as a message key or as simple message text.

Note: This constructor can only be used for simple text or message keys that do not take optional arguments. If the message key is invalid then the object is created with a message-key error message.

Owning Class

CaException

Arguments

bool bKey-if true, szMessage is a message key; if false, plain text

LPCTSTR szMessage-either a message key or message text

Return Value

None

Example

```
CaException ex(false, "message text");
```

More information:

- HSDK Programming Fundamentals
- CaContainer

CaException(const CaException &Cright)

Copy constructor. Duplicates one exception object into another.

Owning Class

CaException

Arguments

const CaException &Cright-the object to be copied

Return Value

None

Example

```
CaException ex(false, "message text"); CaException excopy(ex);
```

More information:

- HSDK Programming Fundamentals
- CaContainer

CaException(LPCTSTR szKey, ...)

Constructs an exception object with the message indexed by the specified key.

Note: If the message key is invalid then the object is created with a message-key error message.

Owning Class

CaException

Arguments

LPCTSTR szKey-the message key

...-up to four optional positional arguments required by the message indexed by szKey. If optional arguments are used then they must be finalized with a NULL argument. Messages that require arguments may require either integer or string arguments.

Return Value

None

Example

```
CaException ex(HSDKKEYVALUE, 1, "arg2", "arg3", 4, NULL);
```

More information:

- HSDK Programming Fundamentals
- CaContainer

void SetReturnCode(HINT32 iReturnCode)

Sets an optional return code into the exception object.

Owning Class

CaException

Arguments

HINT32 iReturnCode-the value

Return Value

None

Example

```
CaException ex(HSDKKEYVALUE, 1, "arg2", "arg3", 4, NULL); ex.SetReturnCode(54);
```

More information:

- HSDK Programming Fundamentals

HINT32 GetReturnCode(void)

Retrieves the optional return code from the exception object.

Owning Class

CaException

Arguments

None

Return Value

Code value

Example

```
CaException ex(HSDKKEYVALUE, 1, "arg2", "arg3", 4, NULL); ex.GetReturnCode();
```

More information:

- HSDK Programming Fundamentals

CaString GetMessage(void)

Retrieves the full formatted message from the exception object.

Owning Class

CaException

Arguments

None

Return Value

Message value

Example

```
CaException ex(HSDKKEYVALUE, 1, "arg2", "arg3", 4, NULL); ex.GetMessage();
```

More information:

- HSDK Programming Fundamentals
- CaString

void LogMessage(void)

Logs the full formatted message to the static ostream set in the CaHarvest object.

Owning Class

CaException

Arguments

None

Return Value

None

Example

```
CaException ex(HSDKKEYVALUE, 1, "arg2", "arg3", 4, NULL); ex.LogMessage();
```

More information:

- HSDK Programming Fundamentals
- CaHarvest

CaForm

The CaForm object is used to retrieve and set values in a form, in addition to creating a new form based on an existing form type. After a new form is created, it can be associated with a package. In addition, form attachments may be added or removed from an existing form.

Public Methods

- CaForm
- GetHistory
- GetColumnHasValue
- GetFormType
- GetColumnValue
- SetColumnValue
- GetFormTypeName
- GetFormObjID
- AddPackage
- RemovePackage
- GetPackageList
- GetNotPackageInList
- GetFormAttachmentList
- AddFormAttachment
- RemoveFormAttachment
- GetFormAttachmentById
- Update
- SetFileAgent
- Delete
- Get
- SetName

More information:

- HSDK Programming Fundamentals
- CaFormType
- CaFormAttachment

CaForm(CaFormType& rFormType, LPCTSTR szName)

This constructor instantiates and creates a new CaForm object based on the form type object specified, optionally assigning the new form a name.

Owning Class

CaForm

Arguments

rFormType - form type to base new form on

szName - optional name of new form

Return Value

None

Exceptions

Throws CaException on error.

Example

```
CaForm rForm(rFormType, szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaFormType

CaForm(CaPackage& rPackage)

This constructor instantiates a new CaForm object based on the parent rPackage specified. This constructor does not create a new form; it is used to create a reference to an existing form.

Owning Class

CaForm

Arguments

rPackage - parent package form is based on

Return Value

None

Exceptions

Throws CaException on error.

Example

```
CaForm rForm(rFormType, szName);
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaFormType

CaContainer& GetHistory()

Returns a CaContainer containing form history.

Note: History attributes are stored with the following hash keys:

- CA_ATTRKEY_ACTION
- CA_ATTRKEY_USERNAME
- CA_ATTRKEY_EXECDCIME

Owning Class

CaForm

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = Form.GetHistory();
```

More information:

- HSDK Programming Fundamentals

bool GetColumnHasValue(LPCTSTR szColName)

Returns a boolean indicating if the given column has a value.

Owning Class

CaForm

Arguments

LPCTSTR - Name of column

Return Value

True if column has value, false otherwise

Exceptions

Throws CaException on error.

Example

```
bool bRet = Form.GetColumnHasValue("Name");
```

More information:

- HSDK Programming Fundamentals
- GetColumnValue
- SetColumnValue

CaFormType& GetFormType(void)

Retrieves the form type from the form.

Owning Class

CaForm

Arguments

None

Return Value

Returns a reference to the form type object

Exceptions

Throws CaException on error.

Example

```
CaFormType& FormType = Form.GetFormType();
```

More information:

- HSDK Programming Fundamentals
- CaFormType

CaData& GetColumnValue(LPCTSTR szColName)

Retrieves the data associated with the specified column.

Owning Class

CaForm

Arguments

LPCTSTR - name of column to retrieve

Return Value

Returns a reference to the column data

Exceptions

Throws CaException on error.

Example

```
CaData& Data = Form.GetColumnValue("Name");
```

More information:

- HSDK Programming Fundamentals
- GetColumnHasValue
- SetColumnValue

bool SetColumnValue(LPCTSTR szColName, const CaData& rValue)

Sets data into a column.

Owning Class

CaForm

Arguments

LPCTSTR - name of column to set

CaData - data to set

Return Value

True if success, false if fail

Exceptions

Throws CaException on error.

Example

```
bool bResult = Form.SetColumnValue("Name", rData);
```

More information:

- HSDK Programming Fundamentals
- GetColumnHasValue
- GetColumnValue

LPCTSTR GetFormTypeName(void)

Retrieves the form type name.

Owning Class

CaForm

Arguments

None

Return Value

LPCTSTR - Name of form type

Exceptions

Throws CaException on error.

Example

```
LPCTSTR szName = Form.GetFormTypeName();
```

More information:

- HSDK Programming Fundamentals
- CaFormType

HAROBJID GetFormTypeObjId(void)

Retrieves the Obj Id of the form type.

Owning Class

CaForm

Arguments

None

Return Value

HAROBJID - Obj Id of the form type

Exceptions

Throws CaException on error.

Example

```
HAROBJID iObjId = Form.GetFormTypeObjId();
```

More information:

- HSDK Programming Fundamentals
- CaFormType

bool AddPackage(CaPackage *pPackage)

Adds a package to this form.

Owning Class

CaForm

Arguments

CaPackage * to package to add

Return Value

True if success, false if fail

Exceptions

Throws CaException on error.

Example

```
bool bResult = Form.AddPackage(pPackage);
```

More information:

- HSDK Programming Fundamentals
- RemovePackage
- GetPackageList
- GetNotInPackageList

bool RemovePackage(CaPackage& rPackage)

Removes a package from this form.

Owning Class

CaForm

Arguments

CaPackage& of package to remove

Return Value

True if success, false if fail

Exceptions

Throws CaException on error.

Example

```
bool bResult = Form.RemovePackage(rPackage);
```

More information:

- HSDK Programming Fundamentals
- AddPackage
- GetPackageList
- GetNotInPackageList

CaPackageList& GetPackageList()

Retrieves package list associated with form.

Owning Class

CaForm

Return Value

Returns a reference to a package list

Exceptions

Throws CaException on error.

Example

```
CaPackageList& List = Form.GetPackageList();
```

More information:

- HSDK Programming Fundamentals
- AddPackage
- RemovePackage
- GetNotInPackageList

CaPackageList& GetNotInPackageList()

Retrieves package list not associated with form.

Owning Class

CaForm

Return Value

Returns a reference to a package list

Exceptions

Throws CaException on error.

Example

```
CaPackageList& List = Form.GetNotInPackageList();
```

More information:

- HSDK Programming Fundamentals
- AddPackage
- RemovePackage
- GetPackageList

CaContainer& GetFormAttachmentList()

Retrieves a list of form attachments.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID

- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME

Owning Class

CaForm

Return Value

Returns a CaContainer object.

Exceptions

Throws CaException on error.

Example

```
CaContainer& Container = Form.GetFormAttachmentList();
```

More information:

- HSDK Programming Fundamentals
- CaFormAttachment
- AddFormAttachment
- RemoveFormAttachment
- GetFormAttachmentById

bool AddFormAttachment(CaString szAttachmentName, int iType, bool bCompress)

Adds an attachment to the form.

Owning Class

CaForm

Arguments

CaString - attachment name

Type - attachment type

Compress - flag if attachment should be compressed

Return Value

Returns true on success, false on fail.

Exceptions

Throws CaException on error.

Example

```
bResult = Form.AddFormAttachment(szFile, ATTACHMENT_FILE, false);
```

More information:

- HSDK Programming Fundamentals
- GetFormAttachmentList
- RemoveFormAttachment
- GetFormAttachment
- GetFormAttachmentById
- SetFileAgent

bool RemoveFormAttachment(CaString szAttachmentName)

Removes an attachment associated with a form.

Owning Class

CaForm

Arguments

CaString - attachment name

Return Value

Returns true on success, false on fail

Exceptions

Throws CaException on error.

Example

```
bool bResult = Form.RemoveFormAttachment(szFile);
```

More information:

- HSDK Programming Fundamentals
- GetFormAttachmentList
- GetFormAttachment
- RemoveFormAttachment
- GetFormAttachmentById

bool RemoveFormAttachment(HAROBJID iObjId)

Removes an attachment associated with a form.

Owning Class

CaForm

Arguments

HAROBJID - Obj id of attachment to remove

Return Value

Returns true on success, false on fail

Exceptions

Throws CaException on error

Example

```
bool bResult = Form.RemoveFormAttachment(ObjId);
```

More information:

- HSDK Programming Fundamentals
- GetFormAttachmentList
- GetFormAttachment
- RemoveFormAttachment
- GetFormAttachmentById

CaFormAttachment* GetFormAttachmentById(HAROBJID ObjId)

Retrieves a form attachment by object ID.

Owning Class

CaForm

Arguments

HAROBJID - object ID of attachment to retrieve

Return Value

Returns form attachment pointer.

Exceptions

Throws CaException on error.

Example

```
CaFormAttachment *pFA = Form.GetformAttachmentById(ObjId);
```

More information:

- HSDK Programming Fundamentals
- GetFormAttachmentList
- GetFormAttachment
- RemoveFormAttachment

HINT32 Delete()

Deletes the form in the CA Harvest SCM database.

Owning Class

CaForm

Arguments

None

Return Value

Zero if success, non-zero if failure

Exceptions

Throws CaException on error.

Example

```
HINT32 iResult = Form.Delete();
```

More information:

- HSDK Programming Fundamentals
- Update

HINT32 Get(HAROBJID iObjId)

Loads the form data based upon the ObjId passed in.

Owning Class

CaForm

Arguments

Objid of form to load

Return Value

Zero if success, non-zero if failure

Exceptions

Throws CaException on error.

Example

```
HINT32 iResult = Form.Get(iObjId);
```

More information:

- HSDK Programming Fundamentals
- Update
- Delete

bool SetName(LPCTSTR szName)

Sets the name of the form

Owning Class

CaForm

Arguments

szName - name to set

Return Value

True on success, false on error

Exceptions

Throws CaException on error.

Example

```
bool bResult = Form.SetName("new name");
```

More information:

- HSDK Programming Fundamentals
- Update

HINT32 Update()

Updates the form in the CA Harvest SCM database.

Owning Class

CaForm

Return Value

Zero if success, non-zero if failure

Exceptions

Throws CaException on error.

Example

```
HINT32 iResult = Form.Update();
```

More information:

- HSDK Programming Fundamentals

void SetFileAgent(LPTSTR szRemoteMachineName, LPTSTR szRemoteUserName, LPTSTR szRemotePassword, LPTSTR szClientPath, LPTSTR szPortNumber)

Sets information used for accessing a remote system.

Owning Class

CaForm

Arguments

LPTSTR - remote machine name

LPTSTR - remote user name

LPTSTR - remote password

LPTSTR - client path

LPTSTR - agent port number

Return Value

None

Exceptions

Throws CaException on error.

Example

```
Form.SetFileAgent(szBroker, szUser, szPassword, szPath, "7001");
```

More information:

- HSDK Programming Fundamentals
- AddFormAttachment

CaFormType

The CaFormType object is used to retrieve default values associated with a CA Harvest SCM form type. The CaFormType can be retrieved from a specific CaForm, or a list of form types can be retrieved from the CaHarvest session.

The CaFormType object cannot be instantiated directly. It must be retrieved from an existing form or CA Harvest SCM session.

Public Methods

- GetColumnDefaultValue
- GetTableName
- GetColumnIndex
- GetColumnMaxLength
- GetColumnName
- GetColumnType
- GetColumnCount

More information:

- HSDK Programming Fundamentals

CaData GetColumnDefaultValue(HINT32 iIndex)

Retrieves the default value at the specified index.

Owning Class

CaFormType

Arguments

HINT32 - index to retrieve

Return Value

Reference to the column data

Exceptions

Throws CaException on error.

Example

```
CaData& rData = FormType.GetColumnDefaultValue(iIndex);
```

More information:

- HSDK Programming Fundamentals
- GetColumnType
- GetColumnCount
- GetColumnName

LPCTSTR GetTableName(void)

Retrieves the table name

Owning Class

CaFormType

Arguments

None

Return Value

LPCTSTR - name of table

Exceptions

Throws CaException on error.

Example

```
LPCTSTR szTable = FormType.GetTableName();
```

More information:

- HSDK Programming Fundamentals

HINT32 GetColumnIndex(LPCTSTR szColumnName)

Retrieves an index based on a column name.

Owning Class

CaFormType

Arguments

LPCTSTR - column name to use

Return Value

HINT32 - index of specified column name

Exceptions

Throws CaException on error.

Example

```
HINT32 iResult = FormType.GetColumnIndex(szColumnName);
```

More information:

- HSDK Programming Fundamentals
- GetColumnName
- GetColumnType
- GetColumnCount

HINT32 GetColumnMaxLength(HINT32 iIndex)

Retrieves the maximum length of the given column.

Owning Class

CaFormType

Arguments

HINT32 - index of column length to retrieve

Return Value

HINT32 - length of column at specified index

Exceptions

Throws CaException on error.

Example

```
HINT32 iResult = FormType.GetColumnMaxLength(iIndex);
```

More information:

- HSDK Programming Fundamentals
- GetColumnName
- GetColumnType
- GetColumnCount

LPCTSTR GetColumnName(HINT32 iIndex)

Retrieves the column name at the specified index.

Owning Class

CaFormType

Arguments

HINT32 - column index to retrieve name from

Return Value

LPCTSTR - name at specified index

Exceptions

Throws CaException on error.

Example

```
LPCTSTR szName = FormType.GetColumnName(iIndex);
```

More information:

- HSDK Programming Fundamentals
- GetColumnType
- GetColumnCount
- GetColumnMaxLength

CaData::Types GetColumnType(HINT32 iIndex)

Retrieves the type of a column.

Owning Class

CaFormType

Arguments

HINT32 - index of column type to retrieve

Return Value

CaData::Types - data type at index

Exceptions

Throws CaException on error.

Example

```
CaData::Types eType = FormType.GetColumnType(iIndex);
```

More information:

- HSDK Programming Fundamentals
- GetColumnName
- GetColumnCount

HUINT32 GetColumnCount(void)

Retrieves the number of columns available.

Owning Class

CaFormType

Arguments

None

Return Value

HUINT32 - number of columns available

Exceptions

Throws CaException on error

Example

```
HUINT32 uResult = FormType.GetColumnCount();
```

More information:

- HSDK Programming Fundamentals
- GetColumnName
- GetColumnType

CaFormAttachment

The CaFormAttachment object is used to get and set attributes associated with a form attachment, as well as retrieve the data associated with a form attachment. The form attachment values can be retrieved from the form with `CaForm::GetFormAttachmentList()`. Specific attachments can then be retrieved with `CaForm::GetFormAttachmentById()`.

The CaFormAttachment object cannot be instantiated directly, it must be retrieved from an existing form.

Public Methods

- GetCompression
- SetCompression
- GetAttachmentType
- SetAttachmentType
- SetFormObjId
- GetFileSize

- CopyFileData
- SetFileAgent

bool GetCompression(void)

Retrieves flag indicating if compression is used.

Owning Class

CaFormAttachment

Arguments

None

Return Value

Returns true on success, false on fail

Exceptions

Throws CaException on error.

Example

```
bool bResult = FormAttachment.GetCompression();
```

More information:

- HSDK Programming Fundamentals
- SetCompression

void SetCompression(bool bCompression)

Sets flag denoting if compression to be used.

Owning Class

CaFormAttachment

Arguments

True if compression used, false if not used

Return Value

None

Exceptions

Throws CaException on error.

Example

```
FormAttachment.SetCompression(false);
```

More information:

- HSDK Programming Fundamentals
- GetCompression

HINT32 GetAttachmentType(void)

Retrieves the attachment type.

Owning Class

CaFormAttachment

Arguments

None

Return Value

HINT32 - attachment type: ATTACHMENT_FILE, ATTACHMENT_URL

Exceptions

Throws CaException on error.

Example

```
HINT32 iResult = FormAttachment.GetAttachmentType();
```

More information:

- HSDK Programming Fundamentals
- SetAttachmentType

void SetAttachmentType(HINT32 iType)

Sets the form attachment type.

Owning Class

CaFormAttachment

Arguments

HINT32 - ATTACHMENT_FILE, ATTACHMENT_URL

Return Value

None

Exceptions

Throws CaException on error.

Example

```
FormAttachment.SetAttachmentType(ATTACHMENT_FILE);
```

More information:

- HSDK Programming Fundamentals
- GetAttachmentType

void SetFormObjId(HINT32 ObjId)

Sets the objid of the given attachment.

Owning Class

CaFormAttachment

Arguments

HINT32 - Object Id to set

Return Value

None

Exceptions

Throws CaException on error.

Example

```
FormAttachment.SetFormObjId(ObjId);
```

More information:

- HSDK Programming Fundamentals

HINT32 GetFileSize(void)

Retrieves the file size of the attachment.

Owning Class

CaFormAttachment

Arguments

None

Return Value

HINT32 - file size

Exceptions

Throws CaException on error.

Example

```
HINT32 iFileSize = FormAttachment.GetFileSize();
```

More information:

- HSDK Programming Fundamentals
- CopyFileData

bool CopyFileData()

Copies file data.

Owning Class

CaFormAttachment

Arguments

None

Return Value

Returns true on success, false on fail

Exceptions

Throws CaException on error.

Example

```
bool bResult = FormAttachment.CopyFileData();
```

More information:

- HSDK Programming Fundamentals
- GetFileSize

void SetFileAgent(LPTSTR szRemoteMachineName, LPTSTR szRemoteUserName, LPTSTR szRemotePassword, LPTSTR szClientPath, LPTSTR szPortNumber)

Sets information used for accessing a remote system.

Owning Class

CaFormAttachment

Arguments

LPTSTR - remote machine name

LPTSTR - remote user name

LPTSTR - remote password

LPTSTR - client path

LPTSTR - agent port number

Return Value

None

Exceptions

Throws CaException on error.

Example

```
FormAttachment.SetFileAgent(szBroker,szUser,szPassword,szPath, "7001");
```

More information:

- HSDK Programming Fundamentals
- CopyFileData

CaHarvest

CaHarvest is the top level HSDK object that provides the CA Harvest SCM login and initializes the HSDK and communications layers. It maintains a static log stream. It also provides access to the Project and User lists.

Public Methods

- CaHarvest
- GetHSDKRelease
- Login
- Logout
- GetAuthenticationSetting
- GetCaseSensLogin
- GetLastMessage
- ChangePassword
- GetProductInfo
- GetProjectList
- GetActiveProjectList
- GetInactiveProjectList
- GetStateList
- GetUserList
- SetLog
- SetStaticLog
- GetLog
- GetContext
- SetFileAccess
- GetBrokerName
- GetFormTypeList

More information:

- HSDK Programming Fundamentals
- CaException
- CaContext
- CaContainer

CaHarvest(void)

This constructor instantiates a CaHarvest.

Owning Class

CaHarvest

Arguments

None

Return Value

None

Exceptions

Throws CaException on error.

Example

```
CaHarvest Harvest;
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetBroker
- GetBrokerName

CaHarvest(LPCTSTR szBrokerName)

This constructor instantiates a CaHarvest object and specifies a CA Harvest SCM service broker.

Owning Class

CaHarvest

Arguments

LPCTSTR szBrokerName - Name of the CA Harvest SCM service broker for connections

Return Value

None

Exceptions

Throws CaException on error.

Example

```
CaHarvest Harvest("MyBrokerName");
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetBroker
- GetBrokerName

CaString GetHSDKRelease(void)

This method returns a string identifying the HSDK Release number for the executed HSDK library.

Owning Class

CaHarvest

Arguments

None

Return Value

CaString - String identifier of HSDK release number

Example

```
CaHarvest Harvest("MyBrokerName");  
cout << Harvest.GetHSDKRelesae() << endl;
```

More information:

- HSDK Programming Fundamentals

HINT32 SetBroker(LPCTSTR szBrokerName)

Connects to the CA Harvest SCM broker.

Note: SetBroker method must be called once after the CaHarvest object is created. It will establish a connection to the RTserver where the broker connects to. In multithreaded application, it must be called once in the main thread and the same connection will be used by all child threads.

Owning Class

CaHarvest

Arguments

LPCTSTR szBrokerName—The name of the CA Harvest SCM Broker.

Return Value

Returns zero if success, nonzero if failure.

Example

```
CaHarvest Harvest;  
if(Harvest.SetBroker("CMBroker") throw CaException(false, "Cannot connect to the  
broker.");
```

More information:

- HSDK Programming Fundamentals

HINT32 Login(LPCTSTR szBrokerName, LPCTSTR szUserName, LPCTSTR szPassword)

Logs into the specified CA Harvest SCM Broker.

Note: If password has expired, the return code will be nonzero and no valid context will be available. However, when password policy allows the changing of an expired password, the ChangePassword method will work. After the password is successfully changed, a valid context and session are available. GetLastMessage will retrieve the message from the server for why login failed

Owning Class

CaHarvest

Arguments

LPCTSTR szBrokerName - The name of the CA Harvest SCM Broker

LPCTSTR szUserName - The User Name

LPCTSTR szPassword - The Password

Return Value

Returns zero if success, nonzero if failure or password policy invocation.

Example

```
CaHarvest Harvest;  
if(Harvest.Login("CMBroker", "CMUser", "MyPassword")) throw CaException(false, "Bad  
Login");
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetBroker
- Logout
- GetLastMessage
- ChangePassword

HINT32 Login(LPCTSTR szUserName, LPCTSTR szPassword)

Logs in to the CA Harvest SCM Broker specified in constructor.

Note: If password has expired, the return code will be nonzero and no valid context will be available. However, when password policy allows the changing of an expired password, the ChangePassword method will work. After the password is successfully changed, a valid context and session are available. GetLastMessage will retrieve the message from the server for why login failed

Owning Class

CaHarvest

Arguments

LPCTSTR szUserName-the User Name

LPCTSTR szPassword-the Password

Return Value

Returns zero if success, nonzero if failure.

Example

```
CaHarvest Harvest;  
if(Harvest.Login "CMUser", "MyPassword")) throw CaException(false, "Bad Login");
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetBroker
- Logout
- GetLastMessage
- ChangePassword

bool GetAuthenticationSetting(enum tAuthenticationServerSettings nSetting)

Retrieves the authentication server flag for the requested setting.

Owning Class

CaHarvest

Arguments

enum tAuthenticationServerSettings nSetting-authentication server setting to retrieve. Possible values include:

- AUTHSERV_EDIT_PASSWORD
- AUTHSERV_EDIT_USERNAME
- AUTHSERV_EDIT_REALNAME
- AUTHSERV_EDIT_PHONE
- AUTHSERV_EDIT_EXT
- AUTHSERV_EDIT_FAX
- AUTHSERV_EDIT_EMAIL
- AUTHSERV_EDIT_NOTE
- AUTHSERV_GET_PASSPOLICY
- AUTHSERV_EDIT_PASSPOLICY
- AUTHSERV_AUTOCREATE_USERNAME

Return Value

bool - flag denoting authentication server setting

Example

```
bool bResult = harvest.GetAuthenticationSetting(AUTHSERV_EDIT_PASSWORD);
```

More information:

- HSDK Programming Fundamentals
- GetCaseSensLogin

bool GetCaseSensLogin()

Retrieves the authentication server flag, denoting if case sensitivity is used for operations pertaining to the login name.

Owning Class

CaHarvest

Arguments

None

Return Value

bool - flag denoting authentication server setting

Example

```
CaHarvest harvest;  
bool bResult = harvest.GetCaseSensLogin();
```

More information:

- HSDK Programming Fundamentals
- GetAuthenticationSetting
- ChangePassword

LPCTSTR GetLastMessage()

Returns the last message from the server for Login or ChangePassword requests.

Owning Class

CaHarvest

Arguments

None

Return Value

LPCTSTR-string message describing reason for login or change password failures or a success message

Example

```
CaHarvest Harvest;  
if(Harvest.Login "CMUser", "MyPassword")) cout << Harvest.GetLastMessage() << endl;
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetBroker
- Login
- Logout
- ChangePassword

HINT32 ChangePassword(LPCTSTR szPassword)

Changes the CA Harvest SCM password of the currently logged in user.

Notes:

- If new password does not match local password policy, a nonzero result code will be returned. Check with GetLastMessage to see why changing the password failed.
- If your site uses internal authentication (CA Harvest SCM authentication), changing the password is enabled.
- If your site uses external authentication, such as Microsoft Active Directory, this function is disabled; you must change the password using methods provided by the authentication server. For example, if using Microsoft Active Directory, you can change passwords by entering Ctrl+Alt+Del and using the Change Password option.
- If your site uses external authentication, changing passwords fails with an error message.

Owning Class

CaHarvest

Arguments

LPCTSTR szPassword-the Password

Return Value

Returns zero if success, nonzero if failure.

Exceptions

Throws CaException on error.

Example

```
CaHarvest Harvest;  
if( Harvest.ChangePassword( szNewPassword ) ) cout << Harvest.GetLastMessage() <<  
endl;
```

More information:

- HSDK Programming Fundamentals
- CaException
- SetBroker
- Login
- Logout
- GetLastMessage
- GetCaseSensLogin

void Logout(void)

Logs out of the CA Harvest SCM Broker. Clears the context.

Owning Class

CaHarvest

Arguments

None

Return Value

None

Example

```
Harvest.Logout();
```

More information:

- HSDK Programming Fundamentals
- Login

LPCTSTR GetBrokerName(void) const

The GetBrokerName function retrieves the name of the current CA Harvest SCM service broker.

Owning Class

CaHarvest

Arguments

None

Return Value

Returns Broker name, empty string if not set

Example

```
CaString szName = Harvest. GetBrokerName();
```

More information:

- HSDK Programming Fundamentals
- SetBroker

CaContext& GetContext(void) const

Retrieves the current context object.

Owning Class

CaHarvest

Arguments

None

Return Value

Returns a reference to the context object

Exceptions

Throws CaException on error.

Example

```
CaContext& rContext = Harvest.GetContext();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContext

CaContainer GetProductInfo(void) const

Retrieves a CaContainer object that contains the CA Harvest SCM product information attributes in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_CLIENT_VERSION
- CA_ATTRKEY_CLIENT_BUILD
- CA_ATTRKEY_CLIENT_PATCH

Owning Class

CaHarvest

Arguments

None

Return Value

Returns a CaContainer object.

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = Harvest.GetProductInfo ();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaContainer GetProjectList(void) const

Retrieves a CaContainer object that contains the attributes for each CA Harvest SCM Project in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME

Owning Class

CaHarvest

Arguments

None

Return Value

Returns a CaContainer object.

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = Harvest.GetProjectList ();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaContainer GetActiveProjectList(void) const

Retrieves a CaContainer object that contains the attributes for each active Project in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME

Owning Class

CaHarvest

Arguments

None

Return Value

Returns a CaContainer object.

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = Harvest.GetActiveProjectList ();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaContainer GetInactiveProjectList(void) const

Retrieve a CaContainer object that contains the attributes for each inactive Project in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME

Owning Class

CaHarvest

Arguments

None

Return Value

Returns a CaContainer object.

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = Harvest.GetInactiveProjectList ();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaContainer GetStateList(HAROBJID iProject) const

Retrieves a CaContainer object that contains the attributes for each State in the Project specified by the HAROBJID iProject argument.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME

Owning Class

CaHarvest

Arguments

HAROBJID iProject-ObjId of the desired project that contains this statelist

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = Harvest.GetStateList ( iProject );
```

CaContainer GetUserList(void) const

Retrieves a CaContainer object that contains the attributes for each User in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME
- CA_ATTRKEY_REALNAME
- CA_ATTRKEY_EMAIL
- CA_ATTRKEY_ACCOUNTLOCKED
- CA_ATTRKEY_ACCOUNTDISABLED
- CA_ATTRKEY_SWLOGIN
- CA_ATTRKEY_ACCOUNTEXTERNAL

Owning Class

CaHarvest

Arguments

None

Return Value

Returns a CaContainer object.

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = Harvest.GetUserList ();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

bool SetLog(ostream &rLogStream)

The SetLog function sets the specified ostream as the default log stream for the CA Harvest SCM program, both HSDK and HAPI instances.

Owning Class

CaHarvest

Arguments

ostream &rLogStream-reference to log stream

Return Value

True if success, false if fail

Example

```
ofstream log("HarvestLog.txt")  
bool bResult = Harvest. SetLog(log);  
if(!bResult) throw exception;
```

More information:

- HSDK Programming Fundamentals
- SetLog
- GetLog

static void SetStaticLog(ostream &rLogStream)

The SetStaticLog function sets the specified ostream as the default static log stream for the HSDK program but not for the HAPI instance.

Owning Class

CaHarvest

Arguments

ostream &rLogStream-reference to log stream

Return Value

None

Example

```
ofstream log("HarvestLog.txt")
Harvest. SetStaticLog(log);
```

More information:

- HSDK Programming Fundamentals
- SetLog
- GetLog

static ostream& GetLog(void)

The static GetLog function retrieves the static log ostream.

Owning Class

CaHarvest

Arguments

None

Return Value

Returns reference to the static log ostream.

Example

```
ofstream &rLog = Harvest.GetLog(log);  
if(!rLog) throw exception;
```

More information:

- HSDK Programming Fundamentals
- SetLog
- GetLog

HINT32 SetFileAccess(LPCTSTR szFullPath, LPCTSTR szAccess)

Sets the file access for the file at the specified path.

Owning Class

CaHarvest

Arguments

LPCTSTR szFullPath-location of the file

LPCTSTR szAccess-new access for file: "vexvexvex" v=read, e=edit, x=execute for Owner, Group, Other

Note: Use '-' for no access: for example, ("vexv--v--")

Return Value

Returns OK if success, nonzero number on error.

Example

```
CaHarvest harvest;  
harvest.SetFileAccess(_T("C:\\temp\\testfile.txt"), _T("vexv-v--"));
```

More information:

- HSDK Programming Fundamentals

CaFormTypeList GetFormTypeList(bool bRefresh)

Gets the list of form types associated with the CaHarvest object.

Owning Class

CaHarvest

Arguments

bRefresh - boolean indicating if list should be refreshed

Return Value

Returns CaFormTypeList with form types if successful, otherwise an error.

Example

```
CaHarvest harvest;  
CaFormTypeList &rFormTypeList;  
rFormTypeList = harvest.GetFormTypeList(true);
```

More information:

- HSDK Programming Fundamentals
- CaFormType

CaInteractiveMerge

CaInteractiveMerge encapsulates the InteractiveMerge process. It cannot be instantiated directly.

The InteractiveMerge process can be set in the CaContext object either by name or object ID. The CaInteractiveMerge object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, merged-tagged version item, and InteractiveMerge Context.

Note: Interactive Merge operators only on the merged-tagged version item in the current context. InteractiveMerge involves three steps: check out, resolve the conflicts with an extensive tool, and check in the resolved file.

Public Methods

- GetCompress
- GetDescription
- GetExecuteFlag
- GetParentPackageName

- GetParentStateName
- GetParentMappedVer
- GetMergedEnvName
- GetMergedStateName
- GetMergedPackageName
- GetMergedMappedVer
- GetExtMergeToolFlag
- GetMergedFile
- GetResolveFlag
- GetScriptFile
- SetDescription
- SetExecuteFlag
- SetExtMergeToolFlag
- SetMergedFile
- SetResolveFlag
- SetScriptFile
- Execute
- Resolve

LPCTSTR GetCompress(void)

Get the compress status of the merge-tagged file.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

Y if compressed, N if not

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szCompress = rInteractiveMerge.GetCompress();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetDescription(void)

Gets the current value of the Interactive Merge description.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-current value of description

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szDes = rInteractiveMerge.GetDescription();
```

More information:

- HSDK Programming Fundamentals
- SetDescription

HINT32 GetExecuteFlag(void)

Gets the interactive merge options: 3-way or 2-way.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

0-3-way
1-2-way

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
HINT32 iExeFlag = rInteractiveMerge.GetExecuteFlag();
```

More information:

- HSDK Programming Fundamentals
- SetExecuteFlag

LPCTSTR GetParentPackageName(void)

Gets the parent package name of the merge-tagged item.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-value of the parent package name

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szName = rInteractiveMerge.GetParentPackageName();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetParentStateName(void)

Gets the parent state name of the merge-tagged item.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-value of the parent state name

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szName = rInteractiveMerge.GetParentStateName();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetParentMappedVer(void)

Gets the parent mapped version of the merge-tagged item.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-value of the parent mapped version name

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szVer = rInteractiveMerge.GetParentMappedVer();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetMergedEnvName(void)

Gets the current merged environment name.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-value of the current merged environment name

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szName = rInteractiveMerge.GetMergedEnvName();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetMergedStateName(void)

Gets the merged state name.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-value of the merged state name

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szName = rInteractiveMerge.GetMergedStateName();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetMergedPackageName(void)

Gets the current merged package name.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-value of the current merged package name

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szName = rInteractiveMerge.GetMergedPackageName();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetMergedMappedVer(void)

Gets the current merged mapped version.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-value of the current merged mapped version

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szVer = rInteractiveMerge.GetMergedMappedVer();
```

More information:

- HSDK Programming Fundamentals

bool GetExtMergeToolFlag(void)

Gets the current merged tool flag.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

True if set, false if not set

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
bool bFlag = rInteractiveMerge.GetExtMergeToolFlag();
```

More information:

- HSDK Programming Fundamentals
- SetExtMergeToolFlag

LPCTSTR GetMergedFile(void)

Gets the merged file.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-value of the merged file name

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szName = rInteractiveMerge.GetMergedFile();
```

More information:

- HSDK Programming Fundamentals
- SetMergedFile

LPCTSTR GetResolveFlag(void)

Gets the resolve flag.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

T-trunk
B-branch
C-cancel
M-merge

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szFlag = rInteractiveMerge.GetResolveFlag();
```

More information:

- HSDK Programming Fundamentals
- SetResolveFlag

LPCTSTR GetScriptFile(void)

Gets the script file.

Owning Class

CaInteractiveMerge

Arguments

None

Return Value

LPCTSTR-value of the script file name

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
CaString szFile = rInteractiveMerge.GetScriptFile();
```

More information:

- HSDK Programming Fundamentals
- SetScriptFile

bool SetDescription(LPCTSTR szDesc)

Sets the description.

Owning Class

CaInteractiveMerge

Arguments

LPCTSTR szDesc

Return Value

True if successful, false if failed

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
bool bRel = rInteractiveMerge.SetDescription("My Description");
```

More information:

- HSDK Programming Fundamentals
- GetDescription

void SetExecuteFlag(HINT32 iExecuteFlag)

Sets the execute flag.

Owning Class

CaInteractiveMerge

Arguments

0-3-way

1-2-way

Return Value

None

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
rInteractiveMerge.SetExecuteFlag(0);
```

More information:

- HSDK Programming Fundamentals
- GetExecuteFlag

void SetExtMergeToolFlag(bool bExtTool)

Sets extensive merge tool.

Owning Class

CaInteractiveMerge

Arguments

true-set extensive merge tool

false-unset extensive merge tool

Return Value

None

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
rInteractiveMerge.SetExtMergeToolFlag(true);
```

More information:

- HSDK Programming Fundamentals
- GetExtMergeToolFlag

void SetMergedFile(LPCTSTR szMergedFile)

Sets merged file.

Owning Class

CaInteractiveMerge

Arguments

LPCTSTR szMergedFile

Return Value

None

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
rInteractiveMerge.SetMergedFile("My merged filename");
```

More information:

- HSDK Programming Fundamentals
- GetMergedFile

void SetResolveFlag(LPCTSTR szResolveFlag)

Sets resolved flag.

Owning Class

CaInteractiveMerge

Arguments

T-trunk

B-branch

C-cancel

M-merge

Return Value

None

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
rInteractiveMerge.SetResolveFlag("M");
```

More information:

- HSDK Programming Fundamentals
- GetResolveFlag

void SetScriptFile(LPCTSTR szScriptFile)

Sets script file.

Owning Class

CaInteractiveMerge

Arguments

LPCTSTR szScriptFile

Return Value

None

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
rInteractiveMerge.SetScriptFile("script");
```

More information:

- HSDK Programming Fundamentals
- GetScriptFile

HINT32 Execute(HINT32 iVersionId)

Executes an interactive merge.

Owning Class

CaInteractiveMerge

Arguments

Version ID of the merge tagged item

Return Value

0 if successful, non-zero if failed

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();  
rInteractiveMerge.SetScriptFile("script");  
rInteractiveMerge.SetExecuteFlag(0);  
rInteractiveMerge.Execute(81);
```

More information:

- HSDK Programming Fundamentals

HINT32 Resolve(HINT32 iVersionId)

Resolves the merge-tagged file.

Owning Class

CaInteractiveMerge

Arguments

Version ID of the merge tagged item

Return Value

0 if successful, non-zero if failed

Example

```
CaInteractiveMerge& rInteractiveMerge=rContext.GetInteractiveMerge();
rInteractiveMerge.SetScriptFile("script");
rInteractiveMerge.SetExecuteFlag(0);
rInteractiveMerge.Execute(81);
rInteractiveMerge.SetMergedFile( "Resolve" );
rInteractiveMerge.SetResolveFlag( "M" );
rInteractiveMerge.Resolve(81);
```

More information:

- HSDK Programming Fundamentals

CaItem

CaItem is the HSDK representation for the Items stored in a repository. This object is read-only, and is used primarily to provide information about the items available in the data view and navigate the repository file items. It is not possible to instantiate an item, but an item may be retrieved from either the Context object or from another item (to retrieve an item's children).

Public Methods

- CaItem
- GetCreationTime
- GetCreatorName
- GetCreatorObjId
- GetItemList
- GetItemPathList
- GetModifiedTime
- GetModifierName
- GetModifierObjId
- GetName
- GetObjId
- GetParentPathName
- GetParentRepositoryObjId
- GetPathFullName

- `GetVersionList`
- `GetViewObjId`
- `GetViewType`
- `IsItemPath`

More information:

- [HSDK Programming Fundamentals](#)
- [CaException](#)
- [CaContext](#)
- [CaContainer](#)

CaItem(const CaItem &src)

This constructor instantiates a `CaItem` from an existing `CaItem`.

Owning Class

`CaItem`

Arguments

`CaItem src`-existing `CaItem` to duplicate on construction

Return Value

None

Exceptions

Throws `CaException` on error.

Example

```
CaItem Item(MyItem);
```

More information:

- [HSDK Programming Fundamentals](#)
- [CaContext](#)
- [CaItemList](#)
- [CaVersion](#)

CaTimeStamp& GetCreationTime(void) const

Returns a reference to a CaTimeStamp that represents the creation time for the Item that this object represents.

Owning Class

CaItem

Arguments

None

Return Value

CaTimeStamp-item creation time

Example

```
CaItem& rItem = rContext.GetViewPathItem(); CaTimeStamp& rItemTime =  
rItem.GetCreationTime();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion

LPCTSTR GetCreatorName(void) const

Returns the string for the name of the CA Harvest SCM user who created this object.

Note: The returned LPCTSTR persists as long as the Item object is valid and in scope.

Owning Class

CaItem

Arguments

None

Return Value

LPCTSTR-item creator name

Example

```
CaItem& rItem = rContext.GetViewPathItem(); LPCTSTR szCreatorUser =  
rItem.GetCreatorName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaltemList
- GetUserName
- GetCreatorObjId

HAROBJID GetCreatorObjId(void) const

Returns the object ID of the CA Harvest SCM user who created this Item.

Owning Class

Caltem

Arguments

None

Return Value

HAROBJID-item creator object ID

Example

```
CaItem& rItem = rContext.GetViewPathItem(); HAROBJID idCreatorUser =  
rItem.GetCreatorObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaltemList
- GetUserName
- GetCreatorName

CaItemList& GetItemList(void) const

Returns a reference to a CaItemList that lists all of the file items and path items that are children in the path defined by this item.

Note: If this method is executed on a CaItem object that represents a file, it will return an empty item list. The results are only meaningful when the CaItem object represents a path.

Owning Class

CaItem

Arguments

None

Return Value

CaItemList-directories and files in this path item. Empty list when item is a file

Example

```
CaItem& rItem = rContext.GetViewPathItem(); CaItemList rItemList=  
rItem.GetItemList();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion
- CaVersionList

CaItemList& GetItemPathList(void) const

Returns a reference to CaItemList objects that lists all the path items that are children in the path defined by this item.

Note: If this method is executed on a CaItem object that represents a file, it will return an empty item list. The results are only meaningful when the CaItem object represents a path.

Owning Class

CaItem

Arguments

None

Return Value

CaItemList-directories beneath this path item. Empty list when item is a file

Example

```
CaItem& rItem = rContext.GetViewPathItem(); CaItemList rPathList =  
rItem.GetItemPathList();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion
- CaVersionList

CaTimeStamp& GetModifiedTime(void) const

Returns a reference to a CaTimeStamp that represents the last modified time for the Item that this object represents.

Owning Class

CaItem

Arguments

None

Return Value

CaTimeStamp-time when item was last modified

Example

```
CaItem& rItem = rContext.GetViewPathItem(); CaTimeStamp& rModifiedTime=  
rItem.GetModifiedTime();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion
- CaVersionList

LPCTSTR GetModifierName(void) const

Returns a string that is the name for the CA Harvest SCM User that last modified this Item.

Note: The returned LPCTSTR persists as long as the Item object is valid and in scope.

Owning Class

CaItem

Arguments

None

Return Value

LPCTSTR-name of last user to modify item

Example

```
CaItem& rItem = rContext.GetViewPathItem(); LPCTSTR szModifierName = rItem.  
GetModifierName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion
- CaVersionList
- GetUserName

HAROBJID GetModifierObjId(void) const

Returns the object ID of the CA Harvest SCM user who last modified this Item.

Owning Class

CaItem

Arguments

None

Return Value

HAROBJID-object ID of the last user to modify this item

Example

```
CaItem& rItem = rContext.GetViewPathItem(); HAROBJID iModifier =  
rItem.GetModifierObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaltemList
- CaVersion
- CaVersionList
- GetUserName

LPCTSTR GetName(void) const

Returns a string that is the filename of the Item in the data view; if the Item is a directory, the string returned is the last child directory of the complete path.

Note: On an Item which is a file, returns the filename. On an Item that is a directory, the last element in the path is returned. The returned LPCTSTR persists as long as the Item object is valid and in scope.

Owning Class

Caltem

Arguments

None

Return Value

LPCTSTR-name of this item

Example

```
CaItem& rItem = rContext.GetViewPathItem(); LPCTSTR szName = rItem.GetName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion
- CaVersionList

HAROBJID GetObjId(void) const

Returns an object ID for this Item.

Owning Class

CaItem

Arguments

None

Return Value

HAROBJID-object ID of the item

Example

```
CaItem& rItem = rContext.GetViewPathItem(); HAROBJID iItem = rItem.GetObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion
- CaVersionList

LPCTSTR GetParentPathName(void) const

Returns a LPCTSTR that is the name of the path in which this Item resides.

Note: The repository corresponds to the top level directory of the complete item path. The returned LPCTSTR persists as long as the Item object is valid and in scope.

Owning Class

CaItem

Arguments

None

Return Value

LPCTSTR-name of the item's parent path

Example

```
CaItem& rItem = rContext.GetViewPathItem(); LPCTSTR szParent =  
rItem.GetParentPathName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion
- CaVersionList

HAROBJID GetParentRepositoryObjId(void) const

Returns an object ID that is the id of the repository in which this Item resides.

Note: The repository corresponds to the top level directory of the complete item path.

Owning Class

CaItem

Arguments

None

Return Value

HAROBJID-object ID of the item's repository

Example

```
CaItem& rItem = rContext.GetViewPathItem(); LPCTSTR szParent =  
rItem.GetParentPathName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion
- CaVersionList

LPCTSTR GetPathFullName(void) const

Returns a string that is the complete path for this Item. It contains the repository name, all directories, and the file name (if the item is a file). Each path element is separated by a '\' character.

Note: Information about baseline and repository views is not accessible through HSDK Caltems.

Owning Class

CaItem

Arguments

None

Return Value

LPCTSTR-item's complete path

Example

```
CaItem& rItem = rContext.GetViewPathItem(); LPCTSTR szItemPath =  
rItem.GetPathFullName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaltemList
- CaVersion
- CaVersionList

HAROBJID GetViewObjId(void) const

Returns the object ID for the view in which the item resides. This view may be the working view associated with a state or a snapshot view.

Note: Information about baseline and repository views is not accessible through HSDK Caltems.

Owning Class

Caltem

Arguments

None

Return Value

HAROBJID-object ID of the view that the item is in

Example

```
CaItem& rItem = rContext.GetViewPathItem(); HAROBJID iView = rItem.GetViewObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaltemList
- CaVersion
- CaVersionList

CaVersionList& GetVersionList(void) const

Returns a reference to a CaVersionList that lists all of the versions available for this item.

Owning Class

CaItem

Arguments

None

Return Value

CaVersionList-versions of this item

Example

```
CaItem& rItem = rContext.GetViewPathItem(); CaVersionList& VersionList =  
rItem.GetVersionList();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaItemList
- CaVersion
- CaVersionList

LPCTSTR GetViewType(void) const

Returns a string describing the type of view in which the item resides. This view may be the working view associated with a state or a snapshot view. Valid types are defined as:

- VIEW_SNAPSHOT "Snapshot"
- VIEW_WORKING "Working"

Note: Information about baseline and repository views is not accessible through HSDK Caltems.

Owning Class

CaItem

Arguments

None

Return Value

HAROBJID-object ID of the view that the item is in

Example

```
CaItem& rItem = rContext.GetViewPathItem(); LPCTSTR szViewType =  
rItem.GetViewType();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaltemList
- CaVersion
- CaVersionList

bool IsItemPath(void) const

Returns a boolean value indicating whether this Item is a path or a file.

Owning Class

Caltem

Arguments

None

Return Value

boolean-true if item is a path; false if it is a file

Example

```
CaItem& rItem = rContext.GetViewPathItem(); bool bPath = rItem.IsItemPath();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaltemList

CaMoveItem

CaMoveItem encapsulates the MoveItem process. It cannot be instantiated directly. The MoveItem process can be set in the CaContext object either by name or object ID. The CaMoveItem object can only be retrieved from the CaContext after it has been set. Set the item either by object ID or by name and path. Set the target item path by Object ID or by path. This is a logical move that creates new versions of the source item (and any sub-items) on the new item path.

Usage

- Set the Project, State, and MoveItem Context.
- Set the MoveItem attributes, including the item to be moved and the path where the item is to be placed.
- Invoke the Execute function.

Public Methods

- setDescription
- SetItemObjId
- SetItemName
- SetItemPath
- SetTargetItemPath
- SetTargetItemPathObjId
- SetClientDir
- SetCreateOnBranch
- SetCreateOnTrunk
- SetBranchOut
- SetRefactorMode
- SetRemoteMachineName
- SetAgentPortNumber
- SetRemoteUserName
- SetRemotePassword
- SetCallBack
- SetCBUserData
- GetDescription
- GetItemObjId

- GetItemName
- GetItemPath
- GetTargetItemPath
- GetTargetItemPathObjId
- GetClientDir
- GetCreateOnBranch
- GetCreateOnTrunk
- GetBranchOut
- GetRefactorMode
- GetRemoteMachineName
- GetAgentPortNumber
- GetRemoteUserName
- GetRemotePassword
- GetCallBack
- GetCBUserData
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

bool SetDescription(LPCTSTR szDesc)

Sets a description to annotate why the item is being moved.

Owning Class

CaMoveItem

Arguments

LPCTSTR szDesc-the explanation

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem. SetDescription("New naming convention");
```

More information:

- HSDK Programming Fundamentals
- GetDescription

void SetItemObjId(HAROBJID iItemObjId)

Identifies the item to be moved by its object ID.

Note: Takes precedence over name. If present then item name and path are ignored.

Owning Class

CaMoveItem

Arguments

HAROBJID iItemObjId

Return Value

None

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem. SetItemObjId(12);
```

More information:

- HSDK Programming Fundamentals
- GetItemObjId
- SetItemName

void SetItemName(LPCTSTR szItemName)

Identifies the item to be moved by its name.

Note: The name is ignored if an item object ID is set. The name is used in conjunction with the path attribute.

Owning Class

CaMoveItem

Arguments

LPCTSTR szItemName

Return Value

None

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem.SetItemName("Item1.txt");
```

More information:

- HSDK Programming Fundamentals
- GetItemName
- SetItemObjId
- SetItemPath

void SetItemPath(LPCTSTR szItemPath)

Sets the view path of the item to be moved.

Owning Class

CaMoveItem

Arguments

LPCTSTR szItemPath-a valid view path

Return Value

None

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem. SetItemPath("\\rep\\code");
```

More information:

- HSDK Programming Fundamentals
- SetItemName
- SetItemObjId
- GetItemPath

bool SetTargetItemPath (LPCTSTR szTargetItemPathName)

Sets the new item path name for the item to be moved. Note that the target item path is set either by pathname or by item object ID.

Owning Class

CaMoveItem

Arguments

LPCTSTR szTargetItemPathName

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem. SetTargetItemPath ("\\rep\\code");
```

More information:

- HSDK Programming Fundamentals
- GetTargetItemPath
- SetTargetItemPathObjId

void SetTargetItemPathObjId(HAROBJID iTargetItemPathObjId)

Identifies the target item path of the item move by its object ID.

Note: Takes precedence over pathname. If present then target item path name are ignored.

Owning Class

CaMoveItem

Arguments

HAROBJID iTargetItemPathObjId

Return Value

None

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem ();  
rMoveItem.SetTargetItemPathObjId(12);
```

More information:

- HSDK Programming Fundamentals
- GetTargetItemPathObjId
- SetTargetItemPath

bool SetClientDir(LPCTSTR szClientDirectory)

Sets the client directory to use for refactoring check-in.

Owning Class

CaMoveItem

Arguments

LPCTSTR szClientDirectory

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem. SetClientDir("C:\myfiles");
```

More information:

- HSDK Programming Fundamentals
- GetClientDir

bool SetCreateOnBranch(bool bFlag)

Create refactored versions on a branch, identical to the *checkout for concurrent update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaMoveItem

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem.SetCreateOnBranch(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch
- GetBranchOut

bool SetCreateOnTrunk(bool bFlag)

Create refactored versions on the trunk, identical to the *checkout for update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaMoveItem

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem.SetCreateOnTrunk(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetBranchOut

bool SetBranchOut(bool bFlag)

On Execute, specify whether refactored versions can be created on the trunk or on a branch, depending on whether *CreateOnBranch* or *CreateOnTrunk* has been specified as true.

Owning Class

CaMoveItem

Arguments

BOOL bFlag-the flag value, true=create on branch, false=create on trunk

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem.SetBranchOut(true);
```

More information:

- HSDK Programming Fundamentals
- SetCreateOnBranch
- SetCreateOnTrunk

bool SetRefactorMode(HINT32 iRefactorMode)

Select the handling of version locks for refactored items.

Owning Class

CaMoveItem

Arguments

HINT32 iRefactorMode-refactor check-in mode values:

- REFACTOR_UPDATE
- REFACTOR_KEEP
- REFACTOR_UPDATE_AND_KEEP

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem.SetRefactorMode(REFACTOR_UPDATE);
```

More information:

- HSDK Programming Fundamentals
- GetRefactorMode

void SetRemoteMachineName(LPCTSTR szRemoteMachineName)

Sets the host name of the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine. A remote CA Harvest SCM File Agent must be running on the specified machine.

Owning Class

CaRenameItem

Arguments

LPCTSTR szRemoteMachineName-the machine name

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetRemoteMachineName("MyBuildMachine");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- GetRemoteMachineName
- SetViewPath
- SetPathOption
- SetAgentPortNumber
- SetRemoteUserName
- SetRemotePassword

void SetAgentPortNumber(LPCTSTR szPortNumber)

Sets the port number of the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine. A remote CA Harvest SCM File Agent must be running on the specified machine.

Owning Class

CaMoveItem

Arguments

LPCTSTR szPortNumber-the agent port number

Return Value

None

Example

```
CaMoveItem& rMoveItem =  
rContext.GetMoveItem(); rMoveItem.SetAgentPortNumber("7001");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool SetClientDir\(LPCTSTR szClientDirectory\)](#) (see page 290)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 294)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 81)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 79)

[LPCTSTR GetAgentPortNumber\(void\)](#) (see page 67)

[LPCTSTR GetRemoteUserName\(void\)](#) (see page 299)

[LPCTSTR GetRemotePassword\(void\)](#) (see page 299)

void SetRemoteUserName(LPCTSTR szRemoteUserName)

Sets the user name for the login to the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaRenameItem

Arguments

LPCTSTR szRemoteUserName

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetRemoteUserName("George");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- SetRemoteMachineName
- SetAgentPortNumber
- SetViewPath
- SetPathOption
- GetRemoteUserName
- SetRemotePassword

void SetRemotePassword(LPCTSTR szRemotePassword)

Sets the password for the username used to log in to the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaRenameItem

Arguments

LPCTSTR szRemotePassword

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetRemotePassword("georgespassword");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- SetRemoteMachineName
- SetAgentPortNumber
- SetViewPath
- SetPathOption
- SetRemoteUserName
- GetRemotePassword

LPCTSTR GetRemoteMachineName(void)

Gets remote machine name used for this process.

Owning Class

CaMoveItem

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
CaString szValue = rMoveItem.GetRemoteMachineName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetRemoteMachineName
- SetAgentPortNumber
- SetRemoteUserName
- SetRemotePassword

LPCTSTR GetAgentPortNumber(void)

Gets agent port number used for this process.

Owning Class

CaMoveItem

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem(); CaString szValue =  
rMoveItem.GetAgentPortNumber();
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 294)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 296)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 297)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 295)

LPCTSTR GetRemoteUserName(void)

Gets remote machine User Name used for this process.

Owning Class

CaMoveItem

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
CaString szValue = rMoveItem.GetRemoteUserName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetAgentPortNumber
- SetRemoteUserName
- SetRemoteMachineName
- SetRemotePassword

LPCTSTR GetRemotePassword(void)

Gets remote machine Password used for this process.

Owning Class

CaMoveItem

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
CaString szValue = rMoveItem.GetRemotePassword();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetAgentPortNumber
- SetRemotePassword
- SetRemoteUserName
- SetRemoteMachineName

void SetCallBack(PFN_CAMVICB pfCallBack)

Sets the callback function for the move item function that is called after each item is refactored.

Note: PFN_CAMVICB is a function pointer with the following return value and argument list: void (*PFN_CAMVICB)(int iBaseCBReturn, CaMoveItem* pMoveItem, void *pUserData, CaContainer *pBag = NULL);

pBag contains data from the checkout operation. The callback function has ownership of pBag and must delete the data it points to.

Owning Class

CaMoveItem

Arguments

PFN_CARNICB-function pointer to callback

Return Value

None

Example

```
void cbFunction(int iBaseCBReturn, CaMoveItem* pMoveItem, void *pUserData,
CaContainer *pBag = NULL)
{
    CaString* pString = (CaString*)pUserData;
    CaString szViewPath = (LPCTSTR)pBag->Props(ATTR_KEY_VIEWPATH, 0);
    cout << *pString << szViewPath << endl;
}

CaMoveItem& rMoveItem = rContext.GetMoveItem();
rMoveItem.SetCallBack( cbFunction );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCBUserData
- GetCallBack
- GetCBUserData

void SetCBUserData(void* pUserData)

Stores the user data that will be passed to the callback function on each callback execution.

Note: The callback function takes as an argument a void*, and during execution the data set by this method is passed to the callback function through this argument. The callback function must know how to interpret this data itself.

Owning Class

CaMoveItem

Arguments

void*-user defined data type that is expected by callback function

Return Value

None

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
rMoveItem.SetCBUserData( (void*) &usrData );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCallBack
- GetCallBack
- GetCBUserData

LPCTSTR GetDescription(void)

Retrieves the current value of the description attribute.

Owning Class

CaMoveItem

Arguments

None

Return Value

LPCTSTR-current description value

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
CaString szDesc = rMoveItem. GetDescription();
```

More information:

- HSDK Programming Fundamentals
- SetDescription

HAROBJID GetItemObjId(void)

Retrieves the current value of the item object ID attribute.

Owning Class

CaMoveItem

Arguments

None

Return Value

HAROBJID-minus one if not set otherwise the current id

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
HAROBJID iItemId = rMoveItem.GetItemObjId();
```

More information:

- HSDK Programming Fundamentals
- SetItemObjId

LPCTSTR GetItemName(void)

Retrieves the current value of the Item Name attribute.

Owning Class

CaMoveItem

Arguments

None

Return Value

LPCTSTR-current Item Name value

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
CaString szName = rMoveItem.GetItemName ();
```

More information:

- HSDK Programming Fundamentals
- SetItemName

LPCTSTR GetItemPath (void)

Retrieves the current value of the Item Path attribute.

Owning Class

CaMoveItem

Arguments

None

Return Value

LPCTSTR-current Item Path value

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
CaString szName = rMoveItem.GetItemPath ();
```

More information:

- HSDK Programming Fundamentals
- SetItemPath

LPCTSTR GetTargetItemPath (void)

Retrieves the current value of the target item path name attribute.

Owning Class

CaMoveItem

Arguments

None

Return Value

LPCTSTR-current target item path name value

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
CaString szPathName = rMoveItem.GetTargetItemPath();
```

More information:

- HSDK Programming Fundamentals
- SetTargetItemPath

HAROBJID GetTargetItemPathObjId(void)

Retrieves the current value of the target item path object ID attribute.

Owning Class

CaMoveItem

Arguments

None

Return Value

HAROBJID-minus one if not set otherwise the current id

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
HAROBJID iPathItemId = rMoveItem.GetTargetItemPathObjId();
```

More information:

- HSDK Programming Fundamentals
- SetTargetItemPathObjId

SetItemObjIdLPCTSTR GetClientDir(void)

Retrieves the current value of the client directory attribute.

Owning Class

CaMoveItem

Arguments

None

Return Value

LPCTSTR-current client directory value

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
CaString szClientDirectory = rMoveItem. GetClientDir();
```

More information:

- HSDK Programming Fundamentals
- SetClientDir

bool GetCreateOnBranch(void)

Retrieves the current value for creating refactored versions on a branch.

Owning Class

CaMoveItem

Arguments

None

Return Value

bool-current value for “CreateOnBranch” flag associated with the MoveItem process.

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
bool bCreateOnBranch = rMoveItem.GetCreateOnBranch();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk

bool GetCreateOnTrunk(void)

Retrieves the current value for creating refactored versions on the trunk.

Owning Class

CaMoveItem

Arguments

None

Return Value

bool-current value for “CreateOnTrunk” flag associated with the MoveItem process.

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
bool bCreateOnTrunk = rMoveItem.GetCreateOnTrunk();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch

bool GetBranchOut(void)

Retrieves the current value for whether the Execute will be allowed to create refactored versions on the trunk or on a branch.

Owning Class

CaMoveItem

Arguments

None

Return Value

bool-current value for “BranchOut” flag associated with the MoveItem process.

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
bool bBranchOutOnExecute = rMoveItem.GetBranchOut();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetCreateOnBranch

HINT32 GetRefactorMode(void)

Retrieves the current value for the handling of version locks for refactored items.

Owning Class

CaMoveItem

Arguments

None

Return Value

HINT32-current value for “RefactorMode” attribute associated with the MoveItem process.

- REFACTOR_UPDATE
- REFACTOR_KEEP
- REFACTOR_UPDATE_AND_KEEP

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
HINT32 iRefactorMode = rMoveItem.GetRefactorMode();
```

More information:

- HSDK Programming Fundamentals
- GetRefactorMode

PFN_CARNICB GetCallback(void);

Returns the callback function pointer for the move item function that is called after each item is refactored.

Note: PFN_CARNICB is a function pointer with the following return value and argument list: void (*PFN_CARNICB)(int iBaseCBReturn, CaMoveItem *pMoveItem, void *pUserData, CaContainer *pBag = NULL);

Owning Class

CaMoveItem

Arguments

None

Return Value

PFN_CARNICB-function pointer to callback

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
PFN_CARNICB pfCallback = rMoveItem.GetCallBack();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCBUserData
- SetCallBack
- GetCBUserData

void* GetCBUserData(void);

Retrieves the user data that will be passed to the callback function on each callback execution.

Note: The callback function takes as an argument a void*, and during execution the data set by this method is passed to the callback function through this argument. The callback function must know how to interpret this data itself.

Owning Class

CaMoveItem

Arguments

None

Return Value

void*-user defined data type that is expected by callback function

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
void*rMoveItem.GetCBUserData( (void*) &usrData );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCallBack
- GetCallBack
- GetCBUserData

HINT32 Execute(void)

Executes the MoveItem process on the Package currently selected using the Package Chooser and the itemname specified by the class methods. The item path named by the set functions will contain new version for moved items of this package.

Owning Class

CaMoveItem

Arguments

None

Return Value

HINT32

- 0 If success.
- 1 If it is a server side error, detailed information can be found in the log file if the log file is defined at the HSDK calling function.
- 2 If item is not valid.
- 3 If multiple matched items.
- 4 If package set is not correct.

Example

```
CaMoveItem& rMoveItem = rContext.GetMoveItem();  
HINT32 iResult = rMoveItem.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaMovePackage

CaMovePackage encapsulates the MovePackage process. It cannot be instantiated directly. The MovePackage process can be set in the CaContext object either by name or object ID. The CaMovePackage object can only be retrieved from the CaContext after it has been set. Set the package either by object ID or by project name. Set the target project by Object ID or by project name.

void SetToState(LPCTSTR szProjectName, LPCTSTR szStateName)

Sets the target state to which the package should be moved by specifying the project name and state name.

Owning Class

CaMovePackage

Arguments

LPCTSTR szProjectName, LPCTSTR szStateName

Return Value

None

Example

```
CaMovePackage& rMovePackage = rContext.GetMovePackage();  
rMovePackage.SetToState(szProjectName, szStateName);
```

More information:

- HSDK Programming Fundamentals
- GetToState

void SetToState(HAROBJID szProjectId, HAROBJID szStateId)

Sets the target state to which the package should be moved by specifying the project id and state id.

Owning Class

CaMovePackage

Arguments

HAROBJID szProjectId, HAROBJID szStateId

Return Value

None

Example

```
CaMovePackage& rMovePackage = rContext.GetMovePackage();  
rMovePackage.SetToState(szProjectId, szStateId);
```

More information:

- HSDK Programming Fundamentals
- GetToState

void SetIncludePkgHistory(bool bFlag)

Sets the include package history flag to specify whether to include package history or not.

Owning Class

CaMovePackage

Arguments

bool bflag – can be set to true or false

Return Value

None

Example

```
CaMovePackage& rMovePackage = rContext.GetMovePackage();  
rMovePackage.SetIncludePkgHistory(false);
```

More information:

- HSDK Programming Fundamentals
- GetIncludePkgHistory

HAROBJID GetToProjectId(void)

Gets the project id of the target project.

Owning Class

CaMovePackage

Arguments

None

Return Value

HAROBJID – Project id of the target project

Example

```
CaMovePackage& rMovePackage = rContext.GetMovePackage();  
HAROBJID prjid = rMovePackage.GetToProjectId();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetToProjectName(void)

Gets the project name of the target project.

Owning Class

CaMovePackage

Arguments

None

Return Value

LPCTSTR – Project name of the target project

Example

```
CaMovePackage& rMovePackage = rContext.GetMovePackage();  
LPCTSTR prjname = rMovePackage.GetToProjectName();
```

More information:

- HSDK Programming Fundamentals

HAROBJID GetToStateId(void)

Gets the state id of the target state.

Owning Class

CaMovePackage

Arguments

None

Return Value

HAROBJID – State id of the target state

Example

```
CaMovePackage& rMovePackage = rContext.GetMovePackage();  
HAROBJID stateid = rMovePackage.GetToStateId();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetToStateName(void)

Gets the state name of the target state.

Owning Class

CaMovePackage

Arguments

None

Return Value

LPCTSTR – State name of the target state

Example

```
CaMovePackage& rMovePackage = rContext.GetMovePackage();  
LPCTSTR statename = rMovePackage.GetToStateName();
```

More information:

- HSDK Programming Fundamentals

bool GetIncludepkgHistory(void)

Gets the state name of the target state.

Owning Class

CaMovePackage

Arguments

None

Return Value

bool – The value of include package history flag

Example

```
CaMovePackage& rMovePackage = rContext.GetMovePackage();  
bool inclpkgflag = rMovePackage.GetIncludeOkHistory();
```

More information:

- HSDK Programming Fundamentals

HINT32 Execute(void)

Executes the MovePackage process on the Package currently selected using the Package Chooser and the source and target project and state names specified by the class methods.

Owning Class

CaMovePackage

Arguments

None

Return Value

HINT32

-0 If success.

Example

```
CaMovePackage& rMovePackage = rContext.GetMovePackage();  
HINT32 iresult = rMovePackage.execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaMovePath

CaMovePath encapsulates the MovePath process. It cannot be instantiated directly. The MovePath process can be set in the CaContext object either by name or object ID. The CaMovePath object can only be retrieved from the CaContext after it has been set. Set the source item path either by object ID or by path name. Set the target item path by Object ID or by path name. This is a logical move that creates new versions of sourcepath item (and any sub-items of this path) on the new item path.

Usage

- Set the Project, State, and MovePath Context.
- Set the MovePath attributes, including the source and target item paths. The item paths can be set by Object ID or by path name.
- Invoke the Execute function.

Public Methods

- SetDescription
- SetItemPathObjId
- SetItemPath
- SetTargetItemPathObjId
- SetTargetItemPath
- SetCreateOnBranch
- SetCreateOnTrunk
- SetBranchOut
- SetRefactorMode
- GetDescription

- GetItemPathObjId
- GetItemPath
- GetTargetItemPathObjId
- GetTargetItemPath
- GetCreateOnBranch
- GetCreateOnTrunk
- GetBranchOut
- GetRefactorMode
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

bool SetDescription(LPCTSTR szDesc)

Sets a description to annotate why the path is being moved.

Owning Class

CaMovePath

Arguments

LPCTSTR szDesc-the explanation

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
rMovePath. SetDescription("New naming convention");
```

More information:

- HSDK Programming Fundamentals
- GetDescription

void SetItemPathObjId(HAROBJID iItemPathObjId)

Identifies the item path to be moved by its object ID.

Note: Takes precedence over name. If present then item path name is ignored.

Owning Class

CaMovePath

Arguments

HAROBJID iItemPathObjId

Return Value

None

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
rMovePath.SetItemPathObjId(12);
```

More information:

- HSDK Programming Fundamentals
- GetItemPathObjId
- SetItemPath

void SetItemPath(LPCTSTR szItemPathName)

Identifies the item path to be moved by its name.

Note: The name is ignored if an item path object ID is set.

Owning Class

CaMovePath

Arguments

LPCTSTR szItemPathName

Return Value

None

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
rMovePath. SetItemPath ("\\sourcebase\\code");
```

More information:

- HSDK Programming Fundamentals
- GetItemPath
- SetItemPathObjId

bool SetTargetItemPath (LPCTSTR szTargetItemPathName)

Sets the new item path name for the item to be moved. Note that the target item path is set either by path name or by item object ID.

Owning Class

CaMovePath

Arguments

LPCTSTR szTargetItemPathName

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
rMovePath. SetTargetItemPath ("\\rep\\code");
```

More information:

- HSDK Programming Fundamentals
- GetTargetItemPath
- SetTargetItemPathObjId

void SetTargetItemPathObjId(HAROBJID iTargetItemPathObjId)

Identifies the target item path of the item move by its object ID.

Note: Takes precedence over path name. If present then target item path name are ignored.

Owning Class

CaMovePath

Arguments

HAROBJID iTargetItemPathObjId

Return Value

None

Example

```
CaMovePath& rMovePath = rContext.GetMovePath ();  
rMovePath.SetTargetItemPathObjId(12);
```

More information:

- HSDK Programming Fundamentals
- GetTargetItemPathObjId
- SetTargetItemPath

bool SetCreateOnBranch(bool bFlag)

Create refactored versions on a branch, identical to the *checkout for concurrent update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaMovePath

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
rMovePath.SetCreateOnBranch(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch
- GetBranchOut

bool SetCreateOnTrunk(bool bFlag)

Create refactored versions on the trunk, identical to the *checkout for update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaMovePath

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
rMovePath.SetCreateOnTrunk(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetBranchOut

bool SetBranchOut(bool bFlag)

On Execute, allow refactored versions to be created on the trunk or on a branch, depending of whether *CreateOnBranch* or *CreateOnTrunk* has been specified as true.

Owning Class

CaMovePath

Arguments

BOOL bFlag-the flag value, true=create on branch, false=create on trunk

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
rMovePath.SetBranchOut(true);
```

More information:

- HSDK Programming Fundamentals
- SetCreateOnBranch
- SetCreateOnTrunk

bool SetRefactorMode(HINT32 iRefactorMode)

On Execute, specify the handling of version locks for refactored items.

Owning Class

CaMovePath

Arguments

HINT32 iRefactorMode-refactor check-in mode values:

- REFACTOR_UPDATE
- REFACTOR_KEEP
- REFACTOR_UPDATE_AND_KEEP

Return Value

bool-true if set succeeded, false if failed

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
rMovePath.SetRefactorMode(REFACTOR_UPDATE);
```

More information:

- HSDK Programming Fundamentals
- GetRefactorMode

LPCTSTR GetDescription(void)

Retrieves the current value of the description attribute.

Owning Class

CaMovePath

Arguments

None

Return Value

LPCTSTR-current description value

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
CaString szDesc = rMovePath.GetDescription();
```

More information:

- HSDK Programming Fundamentals
- SetDescription

HAROBJID GetItemPathObjId(void)

Retrieves the current value of the source item path object ID attribute.

Owning Class

CaMovePath

Arguments

None

Return Value

HAROBJID-minus one if not set otherwise the current id

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
HAROBJID iItemId = rMovePath.GetItemObjId();
```

More information:

- HSDK Programming Fundamentals
- SetItemObjId

LPCTSTR GetItemPath (void)

Retrieves the current value of the source item path name attribute.

Owning Class

CaMovePath

Arguments

None

Return Value

LPCTSTR-current source item path name value

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
CaString szName = rMovePath.GetItemPath ();
```

More information:

- HSDK Programming Fundamentals
- SetItemPath

LPCTSTR GetTargetItemPath (void)

Retrieves the current value of the target item path name attribute.

Owning Class

CaMovePath

Arguments

None

Return Value

LPCTSTR-current target name value

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
CaString szPathName = rMovePath.GetTargetItemPath();
```

More information:

- HSDK Programming Fundamentals
- SetTargetItemPath

HAROBJID GetTargetItemPathObjId(void)

Retrieves the current value of the target item path object ID attribute.

Owning Class

CaMovePath

Arguments

None

Return Value

HAROBJID-minus one if not set otherwise the current id

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
HAROBJID iPathItemId = rMovePath.GetTargetItemPathObjId();
```

More information:

- HSDK Programming Fundamentals
- SetTargetItemPathObjId

bool GetCreateOnBranch(void)

Retrieves the current value for creating refactored versions on a branch.

Owning Class

CaMovePath

Arguments

None

Return Value

bool-current value for “CreateOnBranch” flag associated with the MovePath process.

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
bool bCreateOnBranch = rMovePath.GetCreateOnBranch();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk

bool GetCreateOnTrunk(void)

Retrieves the current value for creating refactored items on the trunk.

Owning Class

CaMovePath

Arguments

None

Return Value

bool-current value for “CreateOnTrunk” flag associated with the MovePath process.

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
bool bCreateOnTrunk = rMovePath.GetCreateOnTrunk();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch

bool GetBranchOut(void)

Retrieves the current value for whether the Execute will be allowed to create refactored versions on the trunk or on a branch.

Owning Class

CaMovePath

Arguments

None

Return Value

bool-current value for “BranchOut” flag associated with the MovePath process.

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
bool bBranchOutOnExecute = rMovePath.GetBranchOut();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetCreateOnBranch

HINT32 GetRefactorMode(void)

Retrieves the current value for the handling of version locks for refactored items.

Owning Class

CaMovePath

Arguments

None

Return Value

HINT32-current value for “RefactorMode” attribute associated with the MovePath process.

- REFACTOR_UPDATE
- REFACTOR_KEEP
- REFACTOR_UPDATE_AND_KEEP

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
HINT32 iRefactorMode = rMovePath.GetRefactorMode();
```

More information:

- HSDK Programming Fundamentals
- GetRefactorMode

HINT32 Execute(void)

Executes the MovePath process on the Package currently selected using the Package Chooser and the source and target item path names specified by the class methods. The target item path named by the set functions will contain new versions for moved sub-items of this package.

Owning Class

CaMovePath

Arguments

None

Return Value

HINT32

- 0 If success.
- 1 If it is a server side error, detailed information can be found in the log file if the log file is defined at the HSDK calling function.
- 2 If item is not valid.
- 3 If multiple matched items.
- 4 If package set is not correct.

Example

```
CaMovePath& rMovePath = rContext.GetMovePath();  
HINT32 iResult = rMovePath.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaPackage

CaPackage is the HSDK representation for a CA Harvest SCM change package. This object is used primarily to provide metadata information about packages stored in the database. It is not possible to instantiate a package, but they may be retrieved from either the context object or from the package chooser.

Note: The direct deletion of CA Harvest SCM packages is not supported by the server. Use the CaDeletePackage process class to delete packages.

Public Methods

- CaPackage
- GetApprovalReport
- GetAssignedUserName
- GetAssignedUserObjId
- GetBindPackageGroupObjId
- GetCreationTime
- GetCreatorName
- GetCreatorObjId

- `GetDescription`
- `GetEnvironmentName`
- `GetEnvironmentObjId`
- `GetFormList`
- `GetFormListAsList`
- `GetHistory`
- `GetModifiedTime`
- `GetModifierName`
- `GetModifierObjId`
- `GetName`
- `GetObjId`
- `GetPriority`
- `GetStateEntryTime`
- `GetStateName`
- `GetStateObjId`
- `GetVersionList`
- `GetViewName`
- `GetViewObjId`

More information:

- [HSDK Programming Fundamentals](#)
- [CaException](#)
- [CaContext](#)
- [CaContainer](#)

CaPackage(const CaPackage &src)

This constructor instantiates a `CaPackage` from an existing `CaPackage`.

Owning Class

`CaPackage`

Arguments

`CaPackage src`-existing `CaPackage` to duplicate on construction

Return Value

None

Exceptions

Throws CaException on error.

Example

```
CaPackage packageCopy(originalPackage);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

CaFormList GetFormListAsList(bool bRefresh)

Returns a reference to a CaFormList object that contains the forms associated with this package.

Owning Class

CaPackage

Arguments

bRefresh - boolean indicating if list should be refreshed

Return Value

Returns a CaFormList object.

Exceptions

Throws CaException on error.

Example

```
CaFormList &FormList = package.GetFormListAsList(true);
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaForm
- GetFormList

CaContainer GetApprovalReport(void) const

Retrieve a CaContainer object that contains the attributes for the CA Harvest SCM approval report.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME

Owning Class

CaContainer

Arguments

None

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = package.GetApprovalReport();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

LPCTSTR GetAssignedUserName(void) const

Returns the string for the name of the CA Harvest SCM user that this change package is assigned to.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM package object is valid and in scope.

Owning Class

CaPackage

Arguments

None

Return Value

LPCTSTR-package assignee name

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
LPCTSTR szAssignedUser = pPackage->GetAssignedUserName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList
- GetAssignedUserObjId

HAROBJID GetAssignedUserObjId(void) const

Returns the object ID of the CA Harvest SCM user who is assigned to this package.

Owning Class

CaPackage

Arguments

None

Return Value

HAROBJID-package assignee object ID

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
HAROBJID iUser = pPackage->GetAssignedUserObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList
- GetAssignedUserName

HAROBJID GetBindPackageGroupObjId(void) const

Returns the object ID of the package group that this change package belongs to.

Owning Class

CaPackage

Arguments

None

Return Value

HAROBJID-package group object ID or NULL_HAROBJID if this package is not bound to a group

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
HAROBJID iGroup = pPackage->GetBindPackageGroupObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

CaTimeStamp& GetCreationTime(void) const

Returns a reference to a CaTimeStamp that represents the creation time for the CA Harvest SCM change package that this object represents.

Owning Class

CaPackage

Arguments

None

Return Value

CaTimeStamp-package creation time

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
CaTimeStamp& rTime = pPackage->GetCreationTime();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList
- CaVersion

LPCTSTR GetCreatorName(void) const

Returns the string for the name of the CA Harvest SCM user who created this object.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM package object is valid and in scope.

Owning Class

CaPackage

Arguments

None

Return Value

LPCTSTR-package creator name

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
LPCTSTR szCreatorUser = pPackage->GetCreatorName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList
- GetUserName
- GetCreatorObjId

HAROBJID GetCreatorObjId(void) const

Returns the object ID of the CA Harvest SCM user who created this CA Harvest SCM package.

Owning Class

CaPackage

Arguments

None

Return Value

HAROBJID-package creator object ID

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
HAROBJID idCreatorUser = pPackage->GetCreatorObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList
- GetUserName
- GetCreatorName

LPCTSTR GetDescription(void) const

Returns the description of this package.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM package object is valid and in scope.

Owning Class

CaPackage

Arguments

None

Return Value

LPCTSTR-package description, or an empty string if there is none

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
LPCTSTR szDescription = pPackage->GetDescription();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

LPCTSTR GetEnvironmentName(void) const

Returns a string that is the name of the CA Harvest SCM project (environment) that this change package is associated with.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM package object is valid and in scope.

Owning Class

CaPackage

Arguments

None

Return Value

LPCTSTR-name of this package's environment

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
LPCTSTR szEnvName = pPackage->GetEnvironmentName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

HAROBJID GetEnvironmentObjId(void) const

Returns the object ID of the Harves environment (project) that this change package is associated with.

Owning Class

CaPackage

Arguments

None

Return Value

HAROBJID-object ID of this package's environment

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
HAROBJID iID = pPackage->GetEnvironmentObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

CaContainer GetFormList(void) const

Retrieves a CaContainer object that contains the attributes for the CA Harvest SCM forms associated with this package.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME
- CA_ATTRKEY_FORMTYPEOBJID
- CA_ATTRKEY_FORMTYPEOBJID

Owning Class

CaContainer

Arguments

None

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = package.GetFormList();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaContainer GetHistory(void) const

Retrieves a CaContainer object that contains the attributes for the history associated with this package.

Owning Class

CaContainer

Arguments

None

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer container = package.GetHistory();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaTimeStamp& GetModifiedTime(void) const

Returns a reference to a CaTimeStamp that represents the last modified time for the CA Harvest SCM package that this object represents.

Owning Class

CaPackage

Arguments

None

Return Value

CaTimeStamp-time when package was last modified

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
CaTimeStamp& rModifiedTime = pPackage->GetModifiedTime();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList
- CaVersion
- CaVersionList

LPCTSTR GetModifierName(void) const

Returns a string that is the name for the CA Harvest SCM User that last modified this package.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM package object is valid and in scope.

Owning Class

CaPackage

Arguments

None

Return Value

LPCTSTR-name of last user who last modified the package

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
LPCTSTR szModName = pPackage->GetModifierName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

HAROBJID GetModifierObjId(void) const

Returns the object ID of the CA Harvest SCM user who last modified this package.

Owning Class

CaPackage

Arguments

None

Return Value

HAROBJID-object ID of the last user who last modified this package

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
HAROBJID iID = pPackage->GetModifierObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

LPCTSTR GetName(void) const

Returns the name of the change package represented by this object.

Owning Class

CaPackage

Arguments

None

Return Value

LPCTSTR-name of this package

Example

```
CaPackage * pPackage = rContext.GetPackageById(1234);
LPCTSTR szName = pPackage->GetName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

HAROBJID GetObjId(void) const

Returns the object ID for this CA Harvest SCM change package.

Owning Class

CaPackage

Arguments

None

Return Value

HAROBJID-object ID of the package

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
HAROBJID iID = pPackage->GetObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

HINT32 GetPriority(void) const

Returns the priority for this CA Harvest SCM change package.

Owning Class

CaPackage

Arguments

None

Return Value

HINT32-priority of the package

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
HINT32 iPriority = pPackage->GetPriority();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

CaTimeStamp& GetStateEntryTime(void) const

Returns a reference to a CaTimeStamp that represents the time that this change package entered its current state.

Owning Class

CaPackage

Arguments

None

Return Value

CaTimeStamp-state entry time

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
CaTimeStamp& rTime = pPackage->GetStateEntryTime();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

LPCTSTR GetStateName(void) const

Returns the name of the state that the change package represented by this object resides in.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM package object is valid and in scope.

Owning Class

CaPackage

Arguments

None

Return Value

LPCTSTR-state name

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
LPCTSTR szStateName = pPackage->GetStateName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList
- GetStateObjId

HAROBJID GetStateObjId(void) const

Returns the object ID of the thate that this change package resides in.

Owning Class

CaPackage

Arguments

None

Return Value

HAROBJID-state object ID

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
HAROBJID iState = pPackage->GetStateObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList
- GetStateName

CaVersionList& GetVersionList(void) const

Returns a reference to a CaVersionList that lists all the versions bound to this package.

Owning Class

CaPackage

Arguments

None

Return Value

CaVersionList-versions in this package

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
CaVersionList& VersionList = pPackage->GetVersionList();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList
- CaVersion
- CaVersionList

LPCTSTR GetViewName(void) const

Returns the name of the view that this change package resides in.

Owning Class

CaPackage

Arguments

None

Return Value

HAROBJID-object ID of the view that the package is in

Example

```
CaPackage * pPackage = rContext.GetPackageByName(_T("mypackage"));
LPCTSTR szName = pPackage->GetViewName();
```

More information:

- HSDK Programming Fundamentals
- CaContext

HAROBJID GetViewObjId(void) const

Returns the object ID for the view in which the item resides. This view may be the working view associated with a state or a snapshot view.

Note: Information about baseline and repository views is not accessible through HSDK Caltems.

Owning Class

Caltem

Arguments

None

Return Value

HAROBJID-object ID of the view that the item is in

Example

```
CaItem& rItem = rContext.GetViewPathItem(); HAROBJDI iView = rItem.GetViewObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageList

CaPackageChooser

CaPackageChooser cannot be instantiated directly. The CaPackageChooser object can only be retrieved from the CaContext.

The Package Chooser provides a mechanism to select a set of packages based upon filtered criteria. Each Package Chooser attribute adds an additional restriction on the selected package set. The resulting package list is available to CaApprove, CaPromote, and CaDemote processes in the current context. A list of attributes from the selected versions can be retrieved in the form of a CaContainer.

Usage

- Optionally set the Project and State Context.
- Set the desired Package Chooser attributes.
- Invoke the Execute function with the bUseCurrentContext set to either use the current context or override it.

Public Methods

- Clear
- SetPackageName
- SetPackageObjId
- SetProjectObjId
- SetStateObjId
- SetPackageGroupObjId
- SetAssigneeObjId
- SetFromDate
- SetToDate
- SetCreatorObjId
- SetModifierObjId
- SetApproveUserObjId
- SetPriority
- SetSubString
- SetPkgStatus
- SetPkgEntryFromDate
- SetPkgEntryToDate
- GetPackageList

- GetPackageListAsList
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext
- CaContainer
- CaCheckin
- CaCheckout
- CaDeleteVersion

void Clear(void)

Removes all selection criteria.

Owning Class

CaPackageChooser

Arguments

None

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.Clear();
```

More information:

- HSDK Programming Fundamentals
- CaContext

void SetPackageName(LPCTSTR szName, int iIndex = 0)

Sets a name expression for search by name pattern.

Owning Class

CaPackageChooser

Arguments

LPCTSTR szName - Name expression to search for package names

int iIndex - Index value for selecting multiple package names

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetPackageName("?QA: Test?", 0); rPackageChooser.SetPackageName("?QA:  
Document?", 1);
```

More information:

- HSDK Programming Fundamentals
- CaContext

void SetPackageObjId(HAROBJID iObjId, int iIndex = 0)

Sets a name expression for search by name pattern.

Owning Class

CaPackageChooser

Arguments

HAROBJID iObjId - Object Id to search for package with matching IDs

int iIndex - Index value for selecting multiple package IDs

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetPackageObjId(101, 0); rPackageChooser.SetPackageObjId(202, 1);
```

More information:

- HSDK Programming Fundamentals
- CaContext

void SetProjectObjId(HAROBJID iObjId)

Sets Project Object Id if not searching within current context.

Owning Class

CaPackageChooser

Arguments

HAROBJID iObjId-project Object Id

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetProjectObjId(10024);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Execute

void SetStateObjId (HAROBJID iObjId)

Sets the State Object Id if not searching within current context.

Owning Class

CaPackageChooser

Arguments

HAROBJID iObjId-State Id

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetStateObjId(10024);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- Execute

void SetPkgGroupObjId(HAROBJID iObjId)

Sets Package Group Object Id.

Owning Class

CaPackageChooser

Arguments

HAROBJID iObjId-Package Group Id

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetPkgGroupObjId(10024);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext

void SetAssigneeObjId (HAROBJID iObjId)

Sets Package Assignee Object Id.

Owning Class

CaPackageChooser

Arguments

HAROBJID iObjId-assignee Id

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetAssigneeObjId(10024);
```

void SetFromDate(CaTimeStamp& Time)

Sets the From Date value of the date range.

Owning Class

CaPackageChooser

Arguments

CaTimeStamp& Time-the time stamp value

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetFromDate(TimeStamp);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- CaTimeStamp
- SetToDate

void SetToDate(CaTimeStamp& Time)

Sets the To Date value of the date range.

Owning Class

CaPackageChooser

Arguments

CaTimeStamp& Time-the time stamp value

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetToDate(TimeStamp);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- CaTimeStamp
- SetFromDate

void SetCreatorObjId (HAROBJID iObjId)

Sets the Package Creator Object Id.

Owning Class

CaPackageChooser

Arguments

HAROBJID iObjId-creator Id

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetCreatorObjId(10024);
```

void SetModifierObjId (HAROBJID iObjId)

Sets the Package Modifier Object Id.

Owning Class

CaPackageChooser

Arguments

HAROBJID iObjId-most recent Modifier Id

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetModifierObjId(10024);
```

void SetApproveUserObjId (HAROBJID iObjId)

Sets the Package Approve User Object Id.

Owning Class

CaPackageChooser

Arguments

HAROBJID iObjId-ID of user who has approved packages

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetApproveUserObjId(10024);
```

void SetPriority(HINT32 iPriority)

Sets the Package Priority.

Owning Class

CaPackageChooser

Arguments

HINT32 iPriority-user-defined priority value to search for

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetPriority(1);
```

void SetSubString(LPCTSTR szString)

Sets a substring expression for searching the Note attribute of packages.

Owning Class

CaPackageChooser

Arguments

LPCTSTR szString-a substring expression

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetSubString("**required for Version 1.5*");
```

void SetPkgStatus(HINT32 iPkgStatus)

Sets the Package Status Option to search by Package approval status.

Note: The following are the allowable values for the Package Status Option. They are defined in an include file built into the project.

- APPROVE_STATUS_APPROVED-Get only approved packages.
- APPROVE_STATUS_REJECTED-Get only rejected packages.
- APPROVE_STATUS_NOT_APPROVED- Get packages that have not been approved or rejected.
- APPROVE_STATUS_ALL- Get packages with any approval status.

Owning Class

CaPackageChooser

Arguments

HINT32 iPkgStatus-Package Approval Status attribute

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetPkgStatus(APPROVE_STATUS_APPROVED);
```

More information:

- HSDK Programming Fundamentals
- CaContext

void SetPkgEntryFromDate(CaTimeStamp& Time)

Sets the State Entry From Date value of the date range.

Owning Class

CaPackageChooser

Arguments

CaTimeStamp& Time-the time stamp value

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetPkgEntryToDate(TimeStamp);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaTimeStamp
- SetPkgEntryToDate

void SetPkgEntryToDate(CaTimeStamp& Time)

Sets the State Entry To Date value of the date range.

Owning Class

CaPackageChooser

Arguments

CaTimeStamp& Time-the time stamp value

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetPkgEntryToDate(TimeStamp);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaTimeStamp
- SetPkgEntryFromDate

CaContainer GetPackageList(void) const

Retrieves a CaContainer object that contains the attributes for each CA Harvest SCM Package in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME, CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID

- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME
- CA_ATTRKEY_PROJECTID
- CA_ATTRKEY_STATEID
- CA_ATTRKEY_ASSIGNEEID
- CA_ATTRKEY_ASSIGNEE_NAME
- CA_ATTRKEY_PRIORITY

Owning Class

CaPackageChooser

Arguments

None

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = rPackageChooser.GetPackageList();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaPackageList& GetPackageListAsList(void)

Retrieves the package list.

Owning Class

CaPackageChooser

Arguments

None

Return Value

Returns a CaPackageList object

Exceptions

Throws CaException on error.

Example

```
CaPackageList& rList = rPackageChooser.GetPackageListAsList();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaPackageList
- CaPackage

void Execute(bool bUseCurrentContext)

Executes the PackageChooser based on the set of package selection criteria set in the package chooser attributes.

Default: true

Owning Class

CaPackageChooser

Arguments

bool bUseCurrentContext-if true, restrict package search to current context; if false, use Project Id and State Id if set

Return Value

None

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser(); HINT32 iResult =  
rPackageChooser.Execute(false);
```


CaPromote

CaPromote encapsulates the Promote process. It cannot be instantiated directly. The Promote process can be set in the CaContext object either by name or object ID. The CaPromote object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, and Promote Context
- Select a set of Change Packages to Promote using the CaPackageChooser object.
- Set the Promote attributes.
- Invoke the Execute function.

Public Methods

- SetMergedPkgsOnly
- SetCheckPackageDependencies
- SetCheckPackageGroupBind
- GetMergedPkgsOnly
- GetCheckPackageDependencies
- GetCheckPackageGroupBind
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageChooser

bool SetMergedPkgsOnly(bool bFlag)

Sets the Merged Packages Only flag.

Owning Class

CaPromote

Arguments

bool bFlag-if true, only merged packages can be promoted; if false, packages with branched versions can be promoted.

Return Value

bool-true if set succeeded, false if failed

Example

```
CaPromote& rPromote = rContext.GetPromote(); rPromote. SetMergedPkgsOnly(true);
```

More information:

- HSDK Programming Fundamentals
- GetMergedPkgsOnly

bool SetCheckPackageDependencies(bool bCheckPackageDependencies)

Sets Check Package Dependencies flag.

Owning Class

CaPromote

Arguments

bool bCheckPackageDependencies-true if verify dependencies prior to promote, false if not

Return Value

bool-true if set succeeded, false if failed

Example

```
CaPromote& rPromote = rContext.GetPromote(); rPromote. SetCheckPackageDependencies (true);
```

More information:

- HSDK Programming Fundamentals
- GetCheckPackageDependencies

bool SetCheckPackageGroupBind(bool bFlag)

Specifies whether or not packages in a package group must all be promoted together.

Owning Class

CaPromote

Arguments

bool bFlag-true if yes, false if no

Return Value

bool-true if set succeeded, false if failed

Example

```
CaPromote& rPromote = rContext.GetPromote(); bool bFlag = rPromote.  
SetCheckPackageGroupBind(true);
```

More information:

- HSDK Programming Fundamentals
- GetCheckPackageGroupBind

bool GetMergedPkgsOnly(void)

Gets the current value of the Merged Packages Only flag.

Owning Class

CaPromote

Arguments

None

Return Value

bool-current value of flag

Example

```
CaPromote& rPromote = rContext.GetPromote(); CaString = rPromote.  
GetMergedPkgsOnly();
```

More information:

- HSDK Programming Fundamentals
- SetMergedPkgsOnly

bool GetCheckPackageDependencies (void)

Gets the current value of the Check Package Dependencies flag.

Owning Class

CaPromote

Arguments

None

Return Value

bool-current value of flag

Example

```
CaPromote& rPromote = rContext.GetPromote(); CaString = rPromote.  
GetCheckPackageDependencies ();
```

More information:

- HSDK Programming Fundamentals
- SetCheckPackageDependencies

bool GetCheckPackageGroupBind (void)

Gets the current value of the Check Package Group Bind flag.

Owning Class

CaPromote

Arguments

None

Return Value

bool-current value of flag

Example

```
CaPromote& rPromote = rContext.GetPromote(); CaString = rPromote.  
GetCheckPackageGroupBind ();
```

More information:

- HSDK Programming Fundamentals
- SetCheckPackageGroupBind

HINT32 Execute(void)

Executes the Promote process on the Package list currently selected using the Package Chooser.

Owning Class

CaPromote

Arguments

None

Return Value

HINT32-zero if success, non-zero if failure

Example

```
CaPromote& rPromote = rContext.GetPromote(); HINT32 iResult = rPromote.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageChooser

CaRemoveItem

CaRemoveItem encapsulates the RemoveItem process. It cannot be instantiated directly. The RemoveItem process can be set in the CaContext object either by name or object ID. The CaRemoveItem object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values. Set the item either by object ID or by name and path. If name is used then the recursive flag is relevant. This is a logical remove that creates a version with a Delete tag.

Usage

- Set the Project, State, and RemoveItem Context.
- Set the RemoveItem attributes, including the item to be removed.
- Invoke the Execute function.

Public Methods

- SetDescription
- SetItemObjId
- SetItemName
- SetItemPath
- SetRecursive
- SetCreateOnBranch
- SetCreateOnTrunk
- SetBranchOut
- GetDescription
- GetItemObjId

- GetItemName
- GetItemPath
- GetRecursive
- GetCreateOnBranch
- GetCreateOnTrunk
- GetBranchOut
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

bool SetDescription(LPCTSTR szDesc)

Sets a description to annotate why the item is being removed.

Owning Class

CaRemoveItem

Arguments

LPCTSTR szDesc-the explanation

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem(); rRemoveItem.  
SetDescription("Obsolete");
```

More information:

- HSDK Programming Fundamentals
- GetDescription

void SetItemObjId(HAROBJID iItemObjId)

Identifies the item to be removed by its object ID.

Note: Takes precedence over name. If present then name, path, and recursive are ignored.

Owning Class

CaRemoveItem

Arguments

HAROBJID iItemObjId

Return Value

None

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem(); rRemoveItem. SetItemObjId(12);
```

More information:

- HSDK Programming Fundamentals
- GetItemObjId
- SetItemName

void SetItemName(LPCTSTR szItemName)

Identifies the item to be removed by its name.

Note: The name is ignored if an item object ID is set. The name is used in conjunction with the path and recursive attributes.

Owning Class

CaRemoveItem

Arguments

LPCTSTR szItemName

Return Value

None

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem(); rRemoveItem.  
SetItemName("Item1.txt");
```

More information:

- HSDK Programming Fundamentals
- GetItemName
- SetItemObjId
- SetItemPath
- SetRecursive

void SetItemPath(LPCTSTR szItemPath)

Sets the view path of the item to be removed.

Owning Class

CaRemoveItem

Arguments

LPCTSTR szItemPath-a valid view path

Return Value

None

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem(); rRemoveItem.  
SetItemPath("\\rep\\code");
```

More information:

- HSDK Programming Fundamentals
- SetItemName
- SetItemObjId
- GetItemPath
- SetRecursive

void SetRecursive(bool bRecursive)

Specifies whether or not to look recursively below the current item path for the item.

Note: Used in conjunction with SetItemName and SetItemPath.

Owning Class

CaRemoveItem

Arguments

bool bRecursive-true if yes, false if no

Return Value

None

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem(); rRemoveItem.  
SetRecursive(true);
```

More information:

- HSDK Programming Fundamentals
- SetItemName
- SetItemObjId
- SetItemPath
- GetRecursive

bool SetCreateOnBranch(bool bFlag)

Create refactored versions on a branch, identical to the *checkout for concurrent update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaRemoveItem

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
rRemoveItem.SetCreateOnBranch(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch
- GetBranchOut

bool SetCreateOnTrunk(bool bFlag)

Create refactored versions on the trunk, identical to the *checkout for update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaRemoveItem

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
rRemoveItem.SetCreateOnTrunk(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetBranchOut

bool SetBranchOut(bool bFlag)

On Execute, create refactored versions on the trunk or a branch version, depending of whether *CreateOnBranch* or *CreateOnTrunk* has been specified as true.

Owning Class

CaRemoveItem

Arguments

BOOL bFlag-the flag value, true=create on branch, false=create on trunk

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
rRemoveItem.SetBranchOut(true);
```

More information:

- HSDK Programming Fundamentals
- SetCreateOnBranch
- SetCreateOnTrunk

LPCTSTR GetDescription(void)

Retrieves the current value of the description attribute.

Owning Class

CaRemoveItem

Arguments

None

Return Value

LPCTSTR-current description value

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
CaString szDesc = rRemoveItem.  
GetDescription();
```

More information:

- HSDK Programming Fundamentals
- SetDescription

HAROBJID GetItemObjId(void)

Retrieves the current value of the item object ID attribute.

Owning Class

CaRemoveItem

Arguments

None

Return Value

HAROBJID-minus one if not set, otherwise the current ID

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();HAROBJID iItemId = rRemoveItem.  
GetItemObjId();
```

More information:

- HSDK Programming Fundamentals
- SetItemObjId

LPCTSTR GetItemName(void)

Retrieves the current value of the Item Name attribute.

Owning Class

CaRemoveItem

Arguments

None

Return Value

LPCTSTR-current Item Name value

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();CaString szName = rRemoveItem.  
GetItemName ();
```

More information:

- HSDK Programming Fundamentals
- SetItemName

LPCTSTR GetItemPath (void)

Retrieves the current value of the Item Path attribute.

Owning Class

CaRemoveItem

Arguments

None

Return Value

LPCTSTR-current Item Path value

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();CaString szName = rRemoveItem.  
GetItemPath ();
```

More information:

- HSDK Programming Fundamentals
- SetItemPath

bool GetRecursive(void)

Retrieves the current value of the Recursive attribute.

Owning Class

CaRemoveItem

Arguments

None

Return Value

bool-current Recursive value

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();bool bRecursive = rRemoveItem.  
GetRecursive();
```

More information:

- HSDK Programming Fundamentals
- SetRecursive

bool GetCreateOnBranch(void)

Retrieves the current value for creating refactored version on a branch.

Owning Class

CaRemoveItem

Arguments

None

Return Value

bool-current value for “CreateOnBranch” flag associated with the RemoveItem process.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
bool bCreateOnBranch = rRemoveItem.GetCreateOnBranch();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch

bool GetCreateOnTrunk(void)

Retrieves the current value for creating refactored versions on the trunk.

Owning Class

CaRemoveItem

Arguments

None

Return Value

bool-current value for “CreateOnTrunk” flag associated with the RemoveItem process.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
bool bCreateOnTrunk = rRemoveItem.GetCreateOnTrunk();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk

bool GetBranchOut(void)

Retrieves the current value for whether the Execute will create refactored versions on the trunk or on a branch.

Owning Class

CaRemoveItem

Arguments

None

Return Value

bool-current value for “BranchOut” flag associated with the RemoveItem process.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();  
bool bBranchOutOnExecute = rRemoveItem.GetBranchOut();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetCreateOnBranch

HINT32 Execute(void)

Executes the RemoveItem process on the Package list currently selected using the Package Chooser.

Owning Class

CaRemoveItem

Arguments

None

Return Value

HINT32

- 0 If success.
- 1 If it is a server side error, detailed information can be found in the log file if the log file is defined at the HSDK calling function.
- 2 If item is not valid.
- 3 If multiple matched items.
- 4 If package set is not correct.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem(); HINT32 iResult =  
rRemoveItem.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageChooser

CaRemovePath

CaRemovePath encapsulates the RemovePath process. It cannot be instantiated directly. The RemovePath process can be set in the CaContext object either by name or object ID. The CaRemovePath object can only be retrieved from the CaContext after it has been set. Set the source item path either by object ID or by path name. This is a logical remove that creates new D-tag versions of the specified path item (and any sub-items of this path).

Usage

- Set the Project, State, and RemovePath Context.
- Set the RemovePath attributes, including the source item path. The item path can be set by Object ID or by path name.
- Invoke the Execute function.

Public Methods

- SetDescription
- SetItemPathObjId
- SetItemPath
- SetCreateOnBranch

- SetCreateOnTrunk
- SetBranchOut
- GetDescription
- GetItemPathObjId
- GetItemPath
- GetCreateOnBranch
- GetCreateOnTrunk
- GetBranchOut
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

bool SetDescription(LPCTSTR szDesc)

Sets a description to annotate why the path is being removed.

Owning Class

CaRemovePath

Arguments

LPCTSTR szDesc-the explanation

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
rRemovePath.SetDescription("New naming convention");
```

More information:

- HSDK Programming Fundamentals
- GetDescription

void SetItemPathObjId(HAROBJID iItemPathObjId)

Identifies the item path to be removed by its object ID.

Note: Takes precedence over name. If present then item path name is ignored.

Owning Class

CaRemovePath

Arguments

HAROBJID iItemPathObjId

Return Value

None

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
rRemovePath.SetItemPathObjId(12);
```

More information:

- HSDK Programming Fundamentals
- GetItemPathObjId
- SetItemPath

void SetItemPath(LPCTSTR szItemPathName)

Identifies the item path to be removed by its name.

Note: The name is ignored if an item path object ID is set.

Owning Class

CaRemovePath

Arguments

LPCTSTR szItemPathName

Return Value

None

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
rRemovePath.SetItemPath (“\sourcebase\code”);
```

More information:

- HSDK Programming Fundamentals
- GetItemPath
- SetItemPathObjId

bool SetCreateOnBranch(bool bFlag)

Create refactored versions on a branch, identical to the *checkout for concurrent update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true, not both.

Owning Class

CaRemovePath

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
rRemovePath.SetCreateOnBranch(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch
- GetBranchOut

bool SetCreateOnTrunk(bool bFlag)

Create refactored versions on the trunk, identical to the *checkout for update* process. Only one *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaRemovePath

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
rRemovePath.SetCreateOnTrunk(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetBranchOut

bool SetBranchOut(bool bFlag)

On Execute, will refactoring be allowed create versions on the trunk or on a branch version, depending of whether *CreateOnBranch* or *CreateOnTrunk* has been specified as true.

Owning Class

CaRemovePath

Arguments

BOOL bFlag-the flag value, true=create on branch, false=create on trunk

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
rRemovePath.SetBranchOut(true);
```

More information:

- HSDK Programming Fundamentals
- SetCreateOnBranch
- SetCreateOnTrunk

LPCTSTR GetDescription(void)

Retrieves the current value of the description attribute.

Owning Class

CaRemovePath

Arguments

None

Return Value

LPCTSTR-current description value

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
CString szDesc = rRemovePath. GetDescription();
```

More information:

- HSDK Programming Fundamentals
- SetDescription

HAROBJID GetItemPathObjId(void)

Retrieves the current value of the source item path object ID attribute.

Owning Class

CaRemovePath

Arguments

None

Return Value

HAROBJID-minus one if not set otherwise the current id

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
HAROBJID iItemId = rRemovePath.GetItemObjId();
```

More information:

- HSDK Programming Fundamentals
- SetItemObjId

LPCTSTR GetItemPath (void)

Retrieves the current value of the source item path name attribute.

Owning Class

CaRemovePath

Arguments

None

Return Value

LPCTSTR-current source item path name value

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
CaString szName = rRemovePath.GetItemPath ();
```

More information:

- HSDK Programming Fundamentals
- SetItemPath

bool GetCreateOnBranch(void)

Retrieves the current value for creating refactored versions on a branch.

Owning Class

CaRemovePath

Arguments

None

Return Value

bool-current value for “CreateOnBranch” flag associated with the RemovePath process.

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
bool bCreateOnBranch = rRemovePath.GetCreateOnBranch();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk

bool GetCreateOnTrunk(void)

Retrieves the current value for creating refactored versions on the trunk.

Owning Class

CaRemovePath

Arguments

None

Return Value

bool-current value for “CreateOnTrunk” flag associated with the RemovePath process.

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
bool bCreateOnTrunk = rRemovePath.GetCreateOnTrunk();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch

bool GetBranchOut(void)

Retrieves the current value for whether the Execute will be allowed to create a refactored versions on the trunk or on a branch.

Owning Class

CaRemovePath

Arguments

None

Return Value

bool-current value for “BranchOut” flag associated with the RemovePath process.

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
bool bBranchOutOnExecute = rRemovePath.GetBranchOut();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetCreateOnBranch

HINT32 Execute(void)

Executes the RemovePath process on the Package currently selected using the Package Chooser and the source item path specified by the class methods. New D-tag versions will be created for sub-items of the path item being removed.

Owning Class

CaRemovePath

Arguments

None

Return Value

HINT32

- 0 If success.
- 1 If it is a server side error, detailed information can be found in the log file if the log file is defined at the HSDK calling function.
- 2 If item is not valid.
- 3 If multiple matched items.
- 4 If package set is not correct.

Example

```
CaRemovePath& rRemovePath = rContext.GetRemovePath();  
HINT32 iResult = rRemovePath.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaRenameItem

CaRenameItem encapsulates the RenameItem process. It cannot be instantiated directly. The RenameItem process can be set in the CaContext object either by name or object ID. The CaRenameItem object can only be retrieved from the CaContext after it has been set. Set the item either by object ID or by name and path. This is a logical rename that creates a new version of an item with a new item name.

Usage

- Set the Project, State, and RenameItem Context.
- Set the RenameItem attributes, including the item to be renamed and the new item name.
- Invoke the Execute function.

Public Methods

- SetDescription
- SetItemObjId
- SetItemName
- SetItemPath
- SetNewItemName
- SetClientDir

- SetCreateOnBranch
- SetCreateOnTrunk
- SetBranchOut
- SetRefactorMode
- SetRemoteMachineName
- SetRemoteUserName
- SetRemotePassword
- SetCallBack
- SetCBUserData
- GetDescription
- GetItemObjId
- GetItemName
- GetItemPath
- GetNewItemName
- GetClientDir
- GetCreateOnBranch
- GetCreateOnTrunk
- GetBranchOut
- GetRefactorMode
- GetRemoteMachineName
- GetRemoteUserName
- GetRemotePassword
- GetCallBack
- GetCBUserData
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

bool SetDescription(LPCTSTR szDesc)

Sets a description to annotate why the item is being renamed.

Owning Class

CaRenameItem

Arguments

LPCTSTR szDesc-the explanation

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetDescription("New naming convention");
```

More information:

- HSDK Programming Fundamentals
- GetDescription

void SetItemObjId(HAROBJID iItemObjId)

Identifies the item to be renamed by its object ID.

Note: Takes precedence over name. If present then name and path are ignored.

Owning Class

CaRenameItem

Arguments

HAROBJID iItemObjId

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetItemObjId(12);
```

More information:

- HSDK Programming Fundamentals
- GetItemObjId
- SetItemName

void SetItemName(LPCTSTR szItemName)

Identifies the item to be renamed by its name.

Note: The name is ignored if an item object ID is set. The name is used in conjunction with the path attribute.

Owning Class

CaRenameItem

Arguments

LPCTSTR szItemName

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetItemName("Item1.txt");
```

More information:

- HSDK Programming Fundamentals
- GetItemName
- SetItemObjId
- SetItemPath
- SetRecursive

void SetItemPath(LPCTSTR szItemPath)

Sets the view path of the item to be renamed.

Owning Class

CaRenameItem

Arguments

LPCTSTR szItemPath-a valid view path

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetItemPath("\\rep\\code");
```

More information:

- HSDK Programming Fundamentals
- SetItemName
- SetItemObjId
- GetItemPath
- SetRecursive

bool SetNewItemName(LPCTSTR szNewItemName)

Sets a new name for the item to be renamed.

Owning Class

CaRenameItem

Arguments

LPCTSTR szNewItemName

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetNewItemName("newfilename.txt");
```

More information:

- HSDK Programming Fundamentals
- GetDescription

bool SetClientDir(LPCTSTR szClientDirectory)

Sets the client directory to use for refactoring check-in.

Owning Class

CaRenameItem

Arguments

LPCTSTR szClientDirectory

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetClientDir("C:\\myfiles");
```

More information:

- HSDK Programming Fundamentals
- GetClientDir

bool SetCreateOnBranch(bool bFlag)

Create refactored versions on a branch, identical to the *checkout for concurrent update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaRenameItem

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetCreateOnBranch(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch
- GetBranchOut

bool SetCreateOnTrunk(bool bFlag)

Create refactored versions on the trunk, identical to the *checkout for update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaRenameItem

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetCreateOnTrunk(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetBranchOut

bool SetBranchOut(bool bFlag)

On Execute, create refactored versions on the trunk or on a branch, depending of whether *CreateOnBranch* or *CreateOnTrunk* has been specified as true.

Owning Class

CaRenameItem

Arguments

BOOL bFlag-the flag value, true=create on branch, false=create on trunk

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetBranchOut(true);
```

More information:

- HSDK Programming Fundamentals
- SetCreateOnBranch
- SetCreateOnTrunk

bool SetRefactorMode(HINT32 iRefactorMode)

On Execute, select how the version locks will be handled for refactored items.

Owning Class

CaRenameItem

Arguments

HINT32 iRefactorMode-refactor check-in mode values:

- REFACTOR_UPDATE-Check in the file and release the lock.
- REFACTOR_UPDATE AND _KEEP-Check in the file and reserve it again for update.
- REFACTOR_KEEP -Retain the lock but do not check in the file.

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetRefactorMode(REFACTOR_UPDATE);
```

More information:

- HSDK Programming Fundamentals
- GetRefactorMode

void SetRemoteMachineName(LPCTSTR szRemoteMachineName)

Sets the host name of the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine. A remote CA Harvest SCM File Agent must be running on the specified machine.

Owning Class

CaRenameItem

Arguments

LPCTSTR szRemoteMachineName-the machine name

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetRemoteMachineName("MyBuildMachine");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- GetRemoteMachineName
- SetViewPath
- SetPathOption
- SetAgentPortNumber
- SetRemoteUserName
- SetRemotePassword

void SetAgentPortNumber(LPCTSTR szPortNumber)

Sets the port number of the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine. A remote CA Harvest SCM File Agent must be running on the specified machine.

Owning Class

CaCheckin

Arguments

LPCTSTR szPortNumber-the agent port number

Return Value

None

Example

```
CaCheckin& rCheckin = rContext.GetCheckin();rCheckin.SetAgentPortNumber("7001");
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[bool SetClientDir\(LPCTSTR szCDir\)](#) (see page 52)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[bool SetViewPath\(LPCTSTR szVPath\)](#) (see page 52)

[bool SetPathOption\(HINT32 iOption\)](#) (see page 50)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

void SetRemoteUserName(LPCTSTR szRemoteUserName)

Sets the user name for the login to the remote machine where the files are located.

Note: If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaRenameItem

Arguments

LPCTSTR szRemoteUserName

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetRemoteUserName("George");
```

More information:

- [HSDK Programming Fundamentals](#)
- [CaContext](#)
- [SetClientDir](#)
- [SetRemoteMachineName](#)
- [SetViewPath](#)
- [SetPathOption](#)
- [GetRemoteUserName](#)
- [SetRemotePassword](#)

void SetRemotePassword(LPCTSTR szRemotePassword)

Sets the password for the username used to log in to the remote machine where the files are located.

Note: If a remote machine is set, then a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Owning Class

CaRenameItem

Arguments

LPCTSTR szRemotePassword

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetRemotePassword("georgespassword");
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetClientDir
- SetRemoteMachineName
- SetViewPath
- SetPathOption
- SetRemoteUserName
- GetRemotePassword

void SetCallBack(PFN_CARMICB pfCallBack)

Sets the callback function for the rename item function that is called after each item is refactored.

Note: PFN_CARMICB is a function pointer with the following return value and argument list: void (*PFN_CARNICB)(int iBaseCBReturn, CaRenameItem* pRenameItem, void *pUserData, CaContainer *pBag = NULL);

pBag contains data from the checkout operation. The callback function has ownership of pBag and must delete the data it points to.

Owning Class

CaRenameItem

Arguments

PFN_CARNICB-function pointer to callback

Return Value

None

Example

```
void cbFunction(int iBaseCBReturn, CaRenameItem* pRenameItem, void *pUserData,
CaContainer *pBag = NULL)
{
    CaString* pString = (CaString*)pUserData;
    CaString szViewPath = (LPCTSTR)pBag->Props(ATTR_KEY_VIEWPATH, 0);
    cout << *pString << szViewPath << endl;
}

CaRenameItem& rRenameItem = rContext.GetRenameItem();
rRenameItem.SetCallBack( cbFunction );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCBUserData
- GetCallBack
- GetCBUserData

void SetCBUserData(void* pUserData)

Stores the user data that will be passed to the callback function on each callback execution.

Note: The callback function takes as an argument a void*, and during execution the data set by this method is passed to the callback function through this argument. The callback function must know how to interpret this data itself.

Owning Class

CaRenameItem

Arguments

void*-user defined data type that is expected by callback function

Return Value

None

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
rRenameItem.SetCBUserData( (void*) &usrData );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCallBack
- GetCallBack
- GetCBUserData

LPCTSTR GetDescription(void)

Retrieves the current value of the description attribute.

Owning Class

CaRenameItem

Arguments

None

Return Value

LPCTSTR-current description value

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
CaString szDesc = rRenameItem. GetDescription();
```

More information:

- HSDK Programming Fundamentals
- SetDescription

HAROBJID GetItemObjId(void)

Retrieves the current value of the item object ID attribute.

Owning Class

CaRenameItem

Arguments

None

Return Value

HAROBJID-minus one if not set otherwise the current id

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
HAROBJID iItemId = rRenameItem. GetItemObjId();
```

More information:

- HSDK Programming Fundamentals
- SetItemObjId

LPCTSTR GetItemName(void)

Retrieves the current value of the Item Name attribute.

Owning Class

CaRenameItem

Arguments

None

Return Value

LPCTSTR-current Item Name value

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
CaString szName = rRenameItem.GetItemName ();
```

More information:

- HSDK Programming Fundamentals
- SetItemName

LPCTSTR GetItemPath (void)

Retrieves the current value of the Item Path attribute.

Owning Class

CaRenameItem

Arguments

None

Return Value

LPCTSTR-current Item Path value

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
CaString szName = rRenameItem.GetItemPath ();
```

More information:

- HSDK Programming Fundamentals
- SetItemPath

LPCTSTR GetNewItemName(void)

Retrieves the current value of the newitemname attribute.

Owning Class

CaRenameItem

Arguments

None

Return Value

LPCTSTR-current newitemname value

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
CaString szNewName = rRenameItem. GetNewItemName();
```

More information:

- HSDK Programming Fundamentals
- setDescription

LPCTSTR GetClientDir(void)

Retrieves the current value of the client directory attribute.

Owning Class

CaRenameItem

Arguments

None

Return Value

LPCTSTR-current client directory value

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
CaString szClientDirectory = rRenameItem. GetClientDir();
```

More information:

- HSDK Programming Fundamentals
- SetClientDir

bool GetCreateOnBranch(void)

Retrieves the current value for creating refactored versions on a branch.

Owning Class

CaRenameItem

Arguments

None

Return Value

bool-current value for “CreateOnBranch” flag associated with the RenameItem process.

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
bool bCreateOnBranch = rRenameItem.GetCreateOnBranch();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk

bool GetCreateOnTrunk(void)

Retrieves the current value for creating refactored versions on the trunk.

Owning Class

CaRenameItem

Arguments

None

Return Value

bool-current value for “CreateOnTrunk” flag associated with the RenameItem process.

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
bool bCreateOnTrunk = rRenameItem.GetCreateOnTrunk();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch

bool GetBranchOut(void)

Retrieves the current value for allowing refactored versions to be created on the trunk or on a branch.

Owning Class

CaRenameItem

Arguments

None

Return Value

bool-current value for “BranchOut” flag associated with the RenameItem process.

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
bool bBranchOutOnExecute = rRenameItem.GetBranchOut();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetCreateOnBranch

HINT32 GetRefactorMode(void)

Retrieves the current value for the handling of version locks for refactored items.

Owning Class

CaRenameItem

Arguments

None

Return Value

HINT32-current value for “RefactorMode” attribute associated with the RenameItem process.

- REFACTOR_UPDATE
- REFACTOR_KEEP
- REFACTOR_UPDATE_AND_KEEP

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
HINT32 iRefactorMode = rRenameItem.GetRefactorMode();
```

More information:

- HSDK Programming Fundamentals
- GetRefactorMode

LPCTSTR GetRemoteMachineName(void)

Gets remote machine name used for this process.

Owning Class

CaRenameItem

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
CString szValue = rRenameItem.GetRemoteMachineName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetRemoteMachineName
- SetRemoteUserName
- SetRemotePassword

LPCTSTR GetAgentPortNumber(void)

Gets agent port number used for this process.

Owning Class

CaCheckin

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaCheckin& rCheckin = rContext.GetCheckin(); CaString szValue =  
rCheckin.GetAgentPortNumber();
```

More information:

[HSDK Programming Fundamentals](#) (see page 36)

[CaContext](#) (see page 106)

[void SetRemoteMachineName\(LPCTSTR szRemoteMachineName\)](#) (see page 54)

[void SetAgentPortNumber\(LPCTSTR szPortNumber\)](#) (see page 55)

[void SetRemoteUserName\(LPCTSTR szRemoteUserName\)](#) (see page 56)

[void SetRemotePassword\(LPCTSTR szRemotePassword\)](#) (see page 57)

LPCTSTR GetRemoteUserName(void)

Gets remote machine User Name used for this process.

Owning Class

CaRenameItem

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem(); CaString szValue =  
rRenameItem.GetRemoteUserName();
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- SetAgentPortNumber
- SetRemoteUserName
- SetRemoteMachineName
- SetRemotePassword

LPCTSTR GetRemotePassword(void)

Gets remote machine Password used for this process.

Owning Class

CaRenameItem

Arguments

None

Return Value

LPCTSTR-the value

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
CaString szValue = rRenameItem.GetRemotePassword();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetAgentPortNumber
- SetRemotePassword
- SetRemoteUserName
- SetRemoteMachineName

PFN_CARNICB GetCallback(void);

Returns the callback function pointer for the rename item function that is called after each item is refactored.

Note: PFN_CARNICB is a function pointer with the following return value and argument list: void (*PFN_CARNICB)(int iBaseCBReturn, CaRenameItem *pRenameItem, void *pUserData, CaContainer *pBag = NULL);

Owning Class

CaRenameItem

Arguments

None

Return Value

PFN_CARNICB-function pointer to callback

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
PFN_CARNICB pfCallback = rRenameItem.GetCallback();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCBUserData
- SetCallback
- GetCBUserData

void* GetCBUserData(void);

Retrieves the user data that will be passed to the callback function on each callback execution.

Note: The callback function takes as an argument a void*, and during execution the data set by this method is passed to the callback function through this argument. The callback function must know how to interpret this data itself.

Owning Class

CaRenameItem

Arguments

None

Return Value

void*-user defined data type that is expected by callback function

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
void*rRenameItem.GetCBUserData( (void*) &usrData );
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetCallBack
- GetCallBack
- GetCBUserData

HINT32 Execute(void)

Executes the RenameItem process on the Package currently selected using the Package Chooser and the itemname specified by the class methods. The item named by the set functions will produce new versions with the new name within this package.

Owning Class

CaRenameItem

Arguments

None

Return Value

HINT32

- 0 If success.
- 1 If it is a server side error, detailed information can be found in the log file if the log file is defined at the HSDK calling function.
- 2 If item is not valid.
- 3 If multiple matched items.
- 4 If package set is not correct.

Example

```
CaRenameItem& rRenameItem = rContext.GetRenameItem();  
HINT32 iResult = rRenameItem.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaRenamePath

CaRenamePath encapsulates the RenamePath process. It cannot be instantiated directly. The RenamePath process can be set in the CaContext object either by name or object ID. The CaRenamePath object can only be retrieved from the CaContext after it has been set. Set the item path to be renamed either by object ID or by path name. This is a logical rename that creates new versions of an path item (and any sub-items of the source path) with the new item path name.

Usage

- Set the Project, State, and RenamePath Context.
- Set the RenamePath attributes, including the item path to be renamed.
- Invoke the Execute function.

Public Methods

- SetDescription
- SetItemPathObjId
- SetItemPath
- SetNewItemPathName
- SetCreateOnBranch
- SetCreateOnTrunk

- SetBranchOut
- SetRefactorMode
- GetDescription
- GetItemPathObjId
- GetItemPath
- GetNewItemPathName
- GetCreateOnBranch
- GetCreateOnTrunk
- GetBranchOut
- GetRefactorMode
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

bool SetDescription(LPCTSTR szDesc)

Sets a description to annotate why the item is being renamed.

Owning Class

CaRenamePath

Arguments

LPCTSTR szDesc-the explanation

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
rRenamePath.SetDescription("New naming convention");
```

More information:

- HSDK Programming Fundamentals
- GetDescription

void SetItemPathObjId(HAROBJID iItemPathObjId)

Identifies the item path to be renamed by its object ID.

Note: Takes precedence over item path name. If present then the path name specification is ignored.

Owning Class

CaRenamePath

Arguments

HAROBJID iItemObjId

Return Value

None

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
rRenamePath.SetItemObjId(12);
```

More information:

- HSDK Programming Fundamentals
- GetItemPathObjId
- SetItemPath

void SetItemPath(LPCTSTR szItemPath)

Sets the name of the view path to be renamed.

Owning Class

CaRenamePath

Arguments

LPCTSTR szItemPath-a valid view path

Return Value

None

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
rRenamePath.SetItemPath("\\rep\\code");
```

More information:

- HSDK Programming Fundamentals
- SetItemPathObjId
- GetItemPath

bool SetNewItemPathName(LPCTSTR szNewItemPathName)

Sets the new name for the view path to be renamed.

Owning Class

CaRenamePath

Arguments

LPCTSTR szNewItemPathName

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
rRenamePath.SetNewItemName("\\sourcebase\\code");
```

More information:

- HSDK Programming Fundamentals
- GetNewItemPathName

bool SetCreateOnBranch(bool bFlag)

Create refactored versions on a branch, identical to the *checkout for concurrent update* process. Only one of *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaRenamePath

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
rRenamePath.SetCreateOnBranch(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch
- GetBranchOut

bool SetCreateOnTrunk(bool bFlag)

Create refactored versions on the trunk, identical to the *checkout for update* process. Only one *CreateOnBranch* or *CreateOnTrunk* can be specified as true; not both.

Owning Class

CaRenamePath

Arguments

BOOL bFlag-the flag value

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
rRenamePath.SetCreateOnTrunk(true);
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetBranchOut

bool SetBranchOut(bool bFlag)

On Execute, create refactored versions on the trunk or on a branch, depending of whether *CreateOnBranch* or *CreateOnTrunk* has been specified as true.

Owning Class

CaRenamePath

Arguments

BOOL bFlag-the flag value, true=create on branch, false=create on trunk

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
rRenamePath.SetBranchOut(true);
```

More information:

- HSDK Programming Fundamentals
- SetCreateOnBranch
- SetCreateOnTrunk

bool SetRefactorMode(HINT32 iRefactorMode)

On Execute, specifies the handling of version locks for refactored items.

Owning Class

CaRenamePath

Arguments

HINT32 iRefactorMode-refactor check-in mode values:

- REFACTOR_UPDATE
- REFACTOR_KEEP
- REFACTOR_UPDATE_AND_KEEP

Return Value

bool-true if set succeeded, false if failed

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
rRenamePath.SetRefactorMode(REFACTOR_UPDATE);
```

More information:

- HSDK Programming Fundamentals
- GetRefactorMode

LPCTSTR GetDescription(void)

Retrieves the current value of the description attribute.

Owning Class

CaRenamePath

Arguments

None

Return Value

LPCTSTR-current description value

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
CaString szDesc = rRenamePath. GetDescription();
```

More information:

- HSDK Programming Fundamentals
- SetDescription

HAROBJID GetItemPathObjId(void)

Retrieves the current value of the view path object ID attribute.

Owning Class

CaRenamePath

Arguments

None

Return Value

HAROBJID-minus one if not set otherwise the current id

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
HAROBJID iItemPathId = rRenamePath. GetItemPathObjId();
```

More information:

- HSDK Programming Fundamentals
- SetItemPathObjId

LPCTSTR GetItemPath (void)

Retrieves the current name of the view path attribute.

Owning Class

CaRenamePath

Arguments

None

Return Value

LPCTSTR-current view path value

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
CaString szName = rRenamePath.GetItemPath ();
```

More information:

- HSDK Programming Fundamentals
- SetItemPath

LPCTSTR GetNewItemPathName(void)

Retrieves the current value of the new view path name attribute.

Owning Class

CaRenamePath

Arguments

None

Return Value

LPCTSTR-current new view path value

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
CaString szNewName = rRenamePath. GetNewItemName();
```

More information:

- HSDK Programming Fundamentals
- SetNewItemPathName

bool GetCreateOnBranch(void)

Retrieves the current value for creating refactored versions on a branch.

Owning Class

CaRenamePath

Arguments

None

Return Value

bool-current value for “CreateOnBranch” flag associated with the RenamePath process.

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
bool bCreateOnBranch = rRenamePath.GetCreateOnBranch();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk

bool GetCreateOnTrunk(void)

Retrieves the current value for creating refactored versions on the trunk.

Owning Class

CaRenamePath

Arguments

None

Return Value

bool-current value for “CreateOnTrunk” flag associated with the RenamePath process.

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
bool bCreateOnTrunk = rRenamePath.GetCreateOnTrunk();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnBranch

bool GetBranchOut(void)

Retrieves the current value for whether the Execute can create refactored versions on the trunk or on a branch, depending on the setting of *CreateOnBranch* and *CreateOnTrunk*.

Owning Class

CaRenamePath

Arguments

None

Return Value

bool-current value for “BranchOut” flag associated with the RenamePath process.

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
bool bBranchOutOnExecute = rRenamePath.GetBranchOut();
```

More information:

- HSDK Programming Fundamentals
- GetCreateOnTrunk
- GetCreateOnBranch

HINT32 GetRefactorMode(void)

Retrieves the current value for handling of version locks for refactored items.

Owning Class

CaRenamePath

Arguments

None

Return Value

HINT32-current value for “RefactorMode” attribute associated with the RenamePath process.

- REFACTOR_UPDATE
- REFACTOR_KEEP
- REFACTOR_UPDATE_AND_KEEP

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
HINT32 iRefactorMode = rRenamePath.GetRefactorMode();
```

More information:

- HSDK Programming Fundamentals
- GetRefactorMode

HINT32 Execute(void)

Executes the RenamePath process on the Package currently selected using the Package Chooser and the view path specified by the class methods. New versions will be create for sub-items of specified view path named by the set functions.

Owning Class

CaRenamePath

Arguments

None

Return Value

HINT32

- 0 If success.
- 1 If it is a server side error, detailed information can be found in the log file if the log file is defined at the HSDK calling function.
- 2 If item is not valid.
- 3 If multiple matched items.
- 4 If package set is not correct.

Example

```
CaRenamePath& rRenamePath = rContext.GetRenamePath();  
HINT32 iResult = rRenamePath.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaSQL

CaSQL encapsulates the ExecuteSQL process. It cannot be instantiated directly. The CaSQL object can only be retrieved from the CaContext, but no other setup is necessary. The CaSQL object is instantiated when the CaContext is created.

Usage

- Set the SQL string value.
- Invoke the Execute function.

Public Methods

- SetSQLStatement
- GetSQLResult
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

void SetSQLStatement(LPCTSTR szSQLStr)

Sets the SQL text to execute in the CA Harvest SCM database.

Owning Class

CaSQL

Arguments

LPCTSTR szSQLStr-the SQL statement

Return Value

None

Example

```
CaSQL& rSQL= rContext.GetSQL();  
rSQL.SetSQLStatement("Select * from HARUSER");  
rSQL.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaString

CaContainer GetSQLResult(void)

Retrieves the results from the execution of the SQL statement.

Owning Class

CaSQL

Arguments

None

Return Value

CaContainer-results

Example

```
CaSQL& rSQL= rContext.GetSQL();  
rSQL.SetSQLStatement("Select * from HARUSER");  
rSQL.Execute();  
CaContainer sqldata = rSQL.GetSQLResult();
```

More information:

- HSDK Programming Fundamentals
- CaContainer

HINT32 Execute(void)

Executes the SQL statement supplied by SetSQLStatement against the CA Harvest SCM database.

Owning Class

CaSQL

Arguments

None

Return Value

HINT32-zero if success, nonzero if failure

Example

```
CaSQL& rSQL= rContext.GetSQL();  
rSQL.SetSQLStatement("Select * from HARUSER");  
rSQL.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaSwitchPackage

CaSwitchPackage encapsulates the SwitchPackage process. It cannot be instantiated directly. The SwitchPackage process can be set in the CaContext object either by name or object ID. The CaSwitchPackage object can only be retrieved from the CaContext after it has been set. Set the package either by object ID or by name and path.

Usage

- Set the Project, State, and SwitchPackage Context.
- Set the SwitchPackage attributes, including the package to be switched and the target package name.
- Invoke the Execute function.

If you need to select a subset of versions from the source package, these versions must be selected in to the version chooser. If the version list is empty, then *all* versions in the source package will be switched to the target package.

Public Methods

- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

HINT32 Execute(LPCTSTR SourcePackage, LPCTSTR TargetPackage)

Executes the SwitchPackage process. To select a subset of the versions from the source package, use the version chooser.

Owning Class

CaSwitchPackage

Arguments

SourcePackage-the source package for the switch package process

TargetPackage-the target package for the switch package process

Return Value

HINT32-zero if success, non-zero if failure

Example

```
CaSwitchPackage& rSwitchPackage = rContext.GetSwitchPackage();  
CaString szName = rSwitchPackage. GetPackageName(true);
```

More information:

- HSDK Programming Fundamentals
- CaContext

HINT32 Execute(HAROBJID SourcePackage, HAROBJID TargetPackage)

Executes the SwitchPackage process. To select a subset of the versions from the source package, use the version chooser.

Owning Class

CaSwitchPackage

Arguments

SourcePackage-the source package ID for the switch package process

TargetPackage-the target package ID for the switch package process

Return Value

HINT32-zero if success, non-zero if failure

Example

```
CaHarvest Harvest; CaContext& rContext = Harvest.GetContext();  
HAROBJID iObjId = rContext.GetPackageId();
```

More information:

- HSDK Programming Fundamentals
- CaContext

CaTakeSnapshot

CaTakeSnapshot encapsulates the TakeSnapshot process. It cannot be instantiated directly. The TakeSnapshot process can be set in the CaContext object either by name or object ID. The CaTakeSnapshot object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, and TakeSnapshot Context.

Three types of snapshots are possible. The first creates a snapshot view of the current contents of the view in selected state in the context. The second uses a passed in CaTimeStamp to take a snapshot of the contents matching that time. The third combines selected packages from Package Chooser and the name of a reference snapshotview to create a snapshot where the contents of the packages are overlayed on the view of the reference snapshot.

- Invoke the Execute function with the appropriate arguments for the desired behavior.

Public Methods

- SetCanViewExternallyFlag
- GetCanViewExternally
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext
- CaTimeStamp
- CaPackageChooser

bool SetCanViewExternallyFlag(bool bFlag)

Sets the CanViewExternally flag true or false.

Note: The CanViewExternally flag defaults to true.

Owning Class

CaTakeSnapshot

Arguments

bool bFlag-if true, snapshot is visible to other projects; if false, only current project has visibility

Return Value

bool-value that was set

Example

```
CaTakeSnapshot& rTakeSnapshot = rContext.GetTakeSnapshot();  
rTakeSnapshot.SetCanViewExternallyFlag(true);
```

More information:

- HSDK Programming Fundamentals
- GetCanViewExternallyFlag

bool GetCanViewExternallyFlag(bool bFlag)

Gets the CanViewExternally flag true or false.

Note: The CanViewExternally flag defaults to true.

Owning Class

CaTakeSnapshot

Arguments

None

Return Value

bool bFlag-if true, snapshot is visible to other projects; if false, only current project has visibility

Example

```
CaTakeSnapshot& rTakeSnapshot = rContext.GetTakeSnapshot();  
bool bViewExternal = rTakeSnapshot.GetCanViewExternallyFlag(true);
```

More information:

- HSDK Programming Fundamentals
- SetCanViewExternallyFlag

HINT32 Execute(LPCTSTR szViewName, CaTimeStamp snapshottime=NULL_CATIMESTAMP, LPCTSTR szReferenceSnapshot = NULL)

Executes the TakeSnapshot process on one of the following:

- The State selected in the CaContext.
- The State selected in the CaContext at a given time.
- The packages selected in the package chooser overlaid on a previous Snapshot view

Owning Class

CaTakeSnapshot

Arguments

LPCTSTR szViewName, CaTimeStamp snapshottime, LPCTSTR szReferenceSnapshot

Return Value

HINT32-zero if success, non-zero if failure

Example

```
CaApprove& rApprove = rContext.GetApprove();  
HINT32 iResult = rApprove.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- CaPackageChooser

CaUDP

CaUDP encapsulates the ExecuteUDP process. It cannot be instantiated directly. The ExecuteUDP process can be set in the CaContext object either by name or object ID. The CaUDP object can only be retrieved from the CaContext after it has been set. The administrator sets default attribute values.

Usage

- Set the Project, State, and ExecuteUDP Context.
- Set the ExecuteUDP attributes.
- Invoke the Execute function.

Public Methods

- SetUsePackageList
- SetUseVersionList

- SetInput
- SetAddCmdLineArg
- GetUDPTType
- GetProgramName
- GetDescription
- GetUserCanModifyInput
- GetUsePackageList
- GetUseVersionList
- GetAddCmdLineArg
- GetInput
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext

bool SetUsePackageList(bool bUsePackageChooser)

Sets a boolean value for whether to pass the UDP process a Package List which would be expanded by the [PACKAGE] system variable.

Note: To pass a list of packages to a UDP process, this boolean is only part of the process. The following must happen:

- The actual UDP command must make use as an argument the system variable [PACKAGE]. The hserver will replace this variable name with the names of the packages sent by the client.
- This boolean must be set before Execute() is called.
- The package chooser must be used to select the packages and executed. The contents of the package chooser must match the contents of the desired package list.

Owning Class

CaUDP

Arguments

bool bUsePackageChooser-to use packages in package chooser or not

Return Value

bool-what was set for this value

Example

```
CaUDP& rUDP = rContext.GetUDP(); rUDP.SetUsePackageList(true); CaPackageChooser  
&rPackageUDP = rContext.GetPackageChooser(); rPackageUDP.Clear();  
rPackageUDP.SetPackageName( rContext.GetPackageName() ); rPackageUDP.Execute();  
rUDP.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaPackageChooser
- GetUsePackageList

bool SetUseVersionList(bool bUseVersionChooser)

Sets a boolean value for whether to pass the UDP process a Version List which would be expanded by the [VERSION] system variable.

Note: To pass a list of versions to a UDP process, this boolean is only part of the process. The following must happen:

- The actual UDP command must make use as an argument the system variable [VERSION]. The hserver will replace this variable name with the names and version number of the versions sent by the client.
- This boolean must be set before Execute() is called.
- The version chooser must be used to select the versions of items and executed. The contents of the version chooser must match the contents of the desired version list.

Owning Class

CaUDP

Arguments

bool bUseVersionChooser-to use versions in version chooser or not

Return Value

bool-what was set for this value

Example

```
CaUDP& rUDP = rContext.GetUDP(); rUDP.SetUseVersionList(true); CaVersionChooser  
&rVersionUDP = rContext.GetVersionChooser(); rVersionUDP.Clear();  
rVersionUDP.SetItemName( "*.cpp" ); rVesrionUDP.Execute(); rUDP.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaVersionChooser
- GetUserVersionList

void SetInput(LPCTSTR szAddInput)

Sets the string that will be piped to stdin for the UDP process.

Note: Permission must be given by the CA Harvest SCM administrator to allow user-modifiable input. See GetUserCanModifyInput() to determine if the administrator has secured input.

SetInput replaces any input that was specified by the administrator for that UDP.

Owning Class

CaUDP

Arguments

LPCTSTR szAddInput-the string to be passed to stdin

Return Value

None

Example

```
CaUDP& rUDP = rContext.GetUDP(); if( rUDP.GetUserCanModifyInput() == true )  
rUDP.SetInput("Input string"); rUDP.Execute();
```

More information:

- HSDK Programming Fundamentals
- GetUserCanModifyInput
- GetInput

void SetAddCmdLineArg(LPCTSTR szCmdLineArg)

Sets additional command line arguments to be passed to the UDP executing process.

Note: SetAddCmdLineArg appends the arguments specified to the entire UDP command. Any arguments specified by the administrator during the setup of the UDP are retained.

Arguments that are specified are prepended with a whitespace character.

Owning Class

CaUDP

Arguments

LPCTSTR szCmdLineArg-the argument's value

Return Value

None

Example

```
CaUDP& rUDP = rContext.GetUDP();  
rUDP.SetAddCmdLineArg("-v -h");  
rUDP.Execute();
```

More information:

- HSDK Programming Fundamentals
- GetAddCmdLineArg

LPCTSTR GetUDPType(void)

Gets the string describing the type of UDP that this process defines.

Note: There are two possible types of UDP:

- **Client**-client-side UDP execution (local machine)
- **Server**-server-side UDP execution (CA Harvest SCM server)

Owning Class

CaUDP

Arguments

None

Return Value

LPCTSTR-string description of UDP type

Example

```
CaUDP& rUDP = rContext.GetUDP();  
LPCTSTR szUDPTYPE = rUDP.GetUDPTYPE();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetProgramName(void)

Gets the complete program name and argument for the UDP process specified by the CA Harvest SCM administrator.

Owning Class

CaUDP

Arguments

None

Return Value

LPCTSTR-string containing command and arguments for UDP process

Example

```
CaUDP& rUDP = rContext.GetUDP();  
LPCTSTR szProgramName = rUDP.GetProgramName();
```

More information:

- HSDK Programming Fundamentals

LPCTSTR GetDescription(void)

Returns the description for the UDP process set by the CA Harvest SCM administrator.

Owning Class

CaUDP

Arguments

LPCTSTR szDescription-the description for this process

Return Value

None

Example

```
CaUDP& rUDP = rContext.GetUDP();  
LPCTSTR szDescription = rUDP.GetDescription();
```

More information:

- HSDK Programming Fundamentals

bool GetUserCanModifyInput(void)

Returns the boolean value that specifies whether “Secure Input” was set for the UDP process by the CA Harvest SCM administrator.

Notes:

- **true**-Using SetInput(), the user can replace the default values for the string that is passed to stdin of the executed UDP.
- **false**-User cannot modify input, using SetInput() will fail.

Owning Class

CaUDP

Arguments

bool-whether input is user modifiable

Return Value

None

Example

```
CaUDP& rUDP = rContext.GetUDP();  
if( rUDP.GetUserCanModifyInput() == true ) rUDP.SetInput("Input String");
```

More information:

- HSDK Programming Fundamentals
- SetInput
- GetInput

bool GetUserPackageList(void)

Returns the current boolean setting for whether on not the packages in the PackageChooser will be sent to the UDP process during Execute().

Note: To pass a list of packages to a UDP process, this boolean is only part of the process. The following must happen:

- The actual UDP command must make use as an argument the system variable [PACKAGE]. The hserver will replace this variable name with the names of the packages sent by the client.
- This boolean must be set before Execute() is called.
- The package chooser must be used to select the packages and executed. The contents of the package chooser must match the contents of the desire package list.

Owning Class

CaUDP

Arguments

None

Return Value

bool-the current setting for using the package list

Example

```
CaUDP& rUDP = rContext.GetUDP();  
bool usepackage = rUDP.GetUsePackageList();
```

More information:

- HSDK Programming Fundamentals
- SetUsePackageList
- CaPackageChooser

bool GetUseVersionList(void)

Returns the current boolean setting for whether or not the versions in the VersionChooser will be sent to the UDP process during Execute().

Note: To pass a list of versions to a UDP process, this boolean is only part of the process. The following must happen:

- The actual UDP command must make use as an argument the system variable [VERSION]. The hserver will replace this variable name with the names and version number of the versions sent by the client.
- This boolean must be set before Execute() is called.
- The version chooser must be used to select the versions of items and executed. The contents of the version chooser must match the contents of the desired version list.

Owning Class

CaUDP

Arguments

None

Return Value

bool-the current setting for using the version list

Example

```
CaUDP& rUDP = rContext.GetUDP(); bool useversion = rUDP.GetUseVersionList();
```

More information:

- HSDK Programming Fundamentals
- SetUsePackageList
- CaVersionChooser

LPCTSTR GetAddCmdLineArg(void)

Retrieves the additional command line arguments that will be passed to the executing UDP.

Note: SetAddCmdLineArg appends the arguments specified to the entire UDP command. Any arguments specified by the administrator during the setup of the UDP are retained.

Arguments that are specified are prepended with a white space character.

Owning Class

CaUDP

Arguments

None

Return Value

LPCTSTR-string of arguments

Example

```
CaUDP& rUDP = rContext.GetUDP(); LPCTSTR szCmdArg = rUDP.GetAddCmdLineArg();
```

More information:

- HSDK Programming Fundamentals
- SetAddCmdLineArg

LPCTSTR GetInput(void)

Returns the current string that will be passed to the stdin of the process on execution.

Note: Permission must be given by the CA Harvest SCM administrator to allow user-modifiable input. See `GetUserCanModifyInput()` to determine if the administrator has secured input.

`GetInput` returns the default string specified by the CA Harvest SCM administrator if not yet modified by `SetInput()`; otherwise returns the current value for the input.

Output from the process will be sent to the CaHarvest log.

Owning Class

CaUDP

Arguments

None

Return Value

LPCTSTR-string that will be passed to input of executing process

Example

```
CaUDP& rUDP = rContext.GetUDP(); LPCTSTR szInput = rUDP.GetInput();
```

More information:

- HSDK Programming Fundamentals
- `SetInput`
- CaHarvest
- `GetLog`

HINT32 Execute(void)

Executes the UDP process.

Note: If using Version Lists or Package Lists, their appropriate chooser must be run Execute() before running CaUDP.Execute().

Output from the process will be sent the CA Harvest SCM object log.

Owning Class

CaUDP

Arguments

None

Return Value

HINT32-zero if success, non-zero if failure

Example

```
CaUDP& rUDP = rContext.GetUDP(); HINT32 iResult = rUDP.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaHarvest
- GetLog
- CaPackageChooser
- CaVersionChooser

CaVersion

Caltem is the HSDK representation for the CA Harvest SCM versions stored on the server. This object is read-only. It is not possible to instantiate a version, but a version may be retrieved from either the Context object or from the version chooser.

Public Methods

- CaVersion
- GetClientMachine
- GetClientPath
- GetCreationTime
- GetCreatorName
- GetCreatorObjId

- `GetDescription`
- `GetItemObjId`
- `GetItemType`
- `GetMappedVersionName`
- `GetMergedVersionId`
- `GetModifiedTime`
- `GetModifierName`
- `GetModifierObjId`
- `GetName`
- `GetObjId`
- `GetPackageName`
- `GetPackageObjId`
- `GetParentVersionObjId`
- `GetPathFullName`
- `GetVersionDataObjId`
- `GetVersionStatus`
- `GetViewObjId`
- `GetRefactorType`
- `GetRefactorByPathVersionObjId`
- `GetPathVersionObjId`

More information:

- [HSDK Programming Fundamentals](#)
- [CaException](#)
- [CaContext](#)
- [CaContainer](#)

CaVersion(const CaVersion &src)

This constructor instantiates a CaVersion from an existing CaVersion.

Owning Class

CaVersion

Arguments

CaVersion src-existing CVersion to duplicate on construction

Return Value

None

Exceptions

Throws CaException on error.

Example

```
CaVersion VersionCopy(MyVersion);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser

LPCTSTR GetClientMachine(void) const

Returns the name of the client machine that this version was checked out to.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM version object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-client machine name

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szMachine pVersion->GetClientMachine();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser

LPCTSTR GetClientPath(void) const

Returns the full path name for the version's checked out location.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM version object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-the client path

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szPath pVersion->GetClientPath();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser

CaTimeStamp& GetCreationTime(void) const

Returns a reference to a CaTimeStamp that represents the creation time for the CA Harvest SCM version that this object represents.

Owning Class

CaVersion

Arguments

None

Return Value

CaTimeStamp-version creation time

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
CaTimeStamp &rTime = pVersion->GetCreationTime();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser

LPCTSTR GetCreatorName(void) const

Returns the string for the name of the CA Harvest SCM user who created this object.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM version object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-version creator name

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szName pVersion->GetCreatorName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- GetCreatorObjId

HAROBJID GetCreatorObjId(void) const

Returns the object ID of the CA Harvest SCM user who created this CA Harvest SCM version.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-version creator object ID

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iCreatorId = pVersion->GetCreatorObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- GetCreatorName

LPCTSTR GetDescription(void) const

Returns the description/comment for this version.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM version object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-version description if one exists, an empty string otherwise

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szDesc pVersion->GetDescription();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaCheckin
- CaVersionChooser

HAROBJID GetItemObjId(void) const

Returns the object ID of the item that this version corresponds to.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-item object ID

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iItemId = pVersion->GetItemObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser

HINT32 GetItemType(void) const

Returns whether or not this version is a file or folder.

Owning Class

CaVersion

Arguments

None

Return Value

HINT32-the item type

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HINT32 iType = pVersion->GetItemType();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser

LPCTSTR GetMappedVersionName(void) const

Returns the mapped version name for this version (for example, “1.2.1”).

Note: The returned LPCTSTR persists as long as the CA Harvest SCM version object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-the mapped version name

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szVersion = pVersion->GetMappedVersionName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

HAROBJID GetMergedVersionId(void) const

Returns the object ID of the trunk merged version for a 3-way merge.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-object ID of the merged version

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iID = pVersion->GetMergedVersionId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

CaTimeStamp& GetModifiedTime(void) const

Returns a reference to a CaTimeStamp that represents the last modified time for the CA Harvest SCM Item that this object represents.

Owning Class

CaVersion

Arguments

None

Return Value

CaTimeStamp-time when version was last modified

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
CaTimeStamp &rTime = pVersion->GetModifiedTime();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

LPCTSTR GetModifierName(void) const

Returns a string that is the name for the CA Harvest SCM User that last modified this CA Harvest SCM version.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM version object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-name of last user to modify the version

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
pVersion->GetModifierName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

HAROBJID GetModifierObjId(void) const

Returns the object ID of the CA Harvest SCM user who last modified this version.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-object ID of the last user to modify this Version

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iUserID = pVersion->GetModifierObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

LPCTSTR GetName(void) const

Returns a string that is the filename of the version in the data view; if the version is a directory, the string returned is the last child directory of the complete path.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM version object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-name of this version

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szName = pVersion->GetName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

HAROBJID GetObjId(void) const

Returns an object ID for this CA Harvest SCM version.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-object ID of the version

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iID = pVersion->GetObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList GetName

LPCTSTR GetPackageName(void) const

Returns the name of the change package that this version is associated with.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM version object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-package name

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szName pVersion->GetPackageName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- GetPackageObjId

HAROBJID GetPackageObjId(void) const

Returns the object ID of the change package that this version is associated with.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-package object ID

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iPackageId = pVersion->GetPackageObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- GetPackageName

LPCTSTR GetParentPathName(void) const

Returns a LPCTSTR that is the name of the parent path to this version.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM Item object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-name of the version's parent path

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szPath = pVersion->GetParentPathName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

HAROBJID GetParentVersionObjId(void) const

Returns an object ID for the version that is the parent to this version in a 3-way merge.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-object ID of the version's parent

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iID = pVersion->GetParentVersionObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

LPCTSTR GetPathFullName(void) const

Returns a string that is the complete path for this version. It contains the repository name, all directories, and the file name (if the item is a file). Each path element is separated by a '\' character.

Note: Information about baseline and repository views is not accessible through HSDK Caltems.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-the complete path

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szFullPath = pVersion->GetPathFullName();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

HAROBJID GetVersionDataObjId(void) const

Returns the object ID of the version data in the database.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-version data object ID

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iId = pVersion->GetVersionDataObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- GetCreatorName

LPCTSTR GetVersionStatus(void) const

Returns the status for this version.

Note: The returned LPCTSTR persists as long as the CA Harvest SCM version object is valid and in scope.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-version status. The returned status will be one of the following:

- NORMAL_TAG_VERSION
- RESERVE_TAG_VERSION
- MERGE_TAG_VERSION
- DELETED_TAG_VERSION

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szStatus = pVersion->GetVersionStatus();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- GetCreatorObjId

HAROBJID GetViewObjId(void) const

Returns the object ID for the view in which the version resides. This view may be the working view associated with a state or a snapshot view.

Note: Information about baseline and repository views is not accessible through HSDK Caltems.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-object ID of the view that the version is in

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iID = pVersion->GetViewObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

LPCTSTR GetRefactorType(void) const

Returns the refactor type indicator.

Owning Class

CaVersion

Arguments

None

Return Value

LPCTSTR-refactor type status. The returned status will be one of the following:

- RENAME - 'A'
- REMOVE - 'D'
- MOVE - 'V'
- By Its Parent - 'S'

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
LPCTSTR szStatus = pVersion->GetRefactorType();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

HAROBJID GetRefactorByPathVersionObjId(void) const

Returns the object ID of the original refactored path version in the case that the Refactor Type is 'S'.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-object ID original refactored path version

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iID = pVersion->GetRefactorByPathVersionObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

HAROBJID GetPathVersionObjId(void) const

Returns the object ID of the parent path version for this version.

Owning Class

CaVersion

Arguments

None

Return Value

HAROBJID-object ID parent path version

Example

```
CaVersion* pVersion = GetVersionById(iVersionId);  
HAROBJID iID = pVersion->GetPathVersionObjId();
```

More information:

- HSDK Programming Fundamentals
- CaContext
- Caltem
- CaVersionChooser
- CaVersionList

CaVersionChooser

CaVersionChooser cannot be instantiated directly. The CaVersionChooser object can only be retrieved from the CaContext after the project and state have been correctly added to context.

The Version Chooser provides a mechanism to select a set of versions based upon filtered criteria. Each Version Chooser attribute adds an additional restriction on the selected version set. The resulting version list is available to CaDeleteVersion, CaCheckin and CaCheckout processes in the current context. A list of attributes from the selected versions can be retrieved in the form of a CaContainer or a CaVersionList.

Usage

- Set the Project and State Context.
Note: If the State is a Snapshot State then set a Snapshot View in the current context. If the State is not a Snapshot State and there is no Working View then the Version Chooser is not active.
- Set the desired Version Chooser attributes.
- Invoke the Execute function.

Public Methods

- Clear
- SetParentPath
- SetExcludePaths
- RemoveVersionFromList
- SetItemObjId
- SetItemName
- SetItemPackageObjId
- SetPackageObjId
- SetRecursive
- SetBranchOption
- SetVersionNumber
- SetVersionOption
- SetVersionItemOption
- SetVersionStatusOption
- SetVersionDateOption
- SetVersionObjIdSetFromDate

- SetToDate
- GetVersionList
- GetVersionListAsList
- Execute

More information:

- HSDK Programming Fundamentals
- CaContext
- CaContainer
- CaCheckin
- CaCheckout
- CaDeleteVersion

void Clear(void)

Removes all selection criteria.

Owning Class

CaVersionChooser

Arguments

None

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.Clear();
```

More information:

- HSDK Programming Fundamentals
- CaContext

void SetParentPath(LPCTSTR szParentPath)

Removes all selection criteria.

Owning Class

CaVersionChooser

Arguments

LPCTSTR szParentPath-parent repository view path to begin search

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetParentPath("\\repl\\mydir");
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- SetRecursive

void SetExcludePaths(bool bFlag)

Excludes all item paths when set to true.

Owning Class

CaVersionChooser

Arguments

bool bFlag-flag to exclude paths

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetExcludePaths(true);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext

void RemoveVersionFromList(HAROBJID iVersionObjId)

Excludes all item paths when set to true.

Note: This function must be called after the Execute function.

Owning Class

CaVersionChooser

Arguments

bool bFlag-flag to exclude paths

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.RemoveVersionFromList(12345);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- Execute

void SetItemObjId(HAROBJID iItemObjId, int iIndex)

Sets one in a set of Item Object Ids whose versions will be searched for.

Note: The set of ids is stored in an ordered array. The index value sets a specific object ID value location. Index locations must be set in numerical order.

Owning Class

CaVersionChooser

Arguments

HAROBJID iItemObjId-item Object Id

int iIndex-the set of IDs is stored in an indexed array. Defaults to zero.

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetItemObjId(10024, 1);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext

void SetItemName(LPCTSTR szItemName)

Sets a name expression to search for versions.

Owning Class

CaVersionChooser

Arguments

LPCTSTR szItemName-a name expression

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetItemName("ca*.h");
```

■ More information:

- HSDK Programming Fundamentals
- CaContext

void SetItemPackageObjId(HAROBJID iPackageObjId, int iIndex)

Sets one in a set of Package Object Ids whose items will be searched for, and returns the latest version in the view.

Note: The set of ids is stored in an ordered array. The index value sets a specific object ID value location. Index locations must be set in numerical order.

Owning Class

CaVersionChooser

Arguments

HAROBJID iPackageObjId - Package Id whose items to search for

int iIndex - The set of ids is stored in an indexed array. Defaults to zero.

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetItemPackageObjId(10024, 1);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext

void SetPackageObjId(HAROBJID iPackageObjId, int iIndex)

Sets one in a set of Package Object Ids whose versions will be searched for.

Note: The set of ids is stored in an ordered array. The index value sets a specific object ID value location. Index locations must be set in numerical order.

Owning Class

CaVersionChooser

Arguments

HAROBJID iPackageObjId-package Id whose versions to search for.

int iIndex-the set of ids is stored in an indexed array. Defaults to zero.

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetPackageObjId(10024, 1);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext

void SetRecursive(bool bFlag)

Sets recursive flag to search below the parent path.

Owning Class

CaVersionChooser

Arguments

bool bFlag-true if recursive, false if no

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetItemName("ca*.h");
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- SetItemName

void SetBranchOption(HINT32 iOptions)

Sets the Branch option to specify whether trunk versions, branch versions, or both are to be retrieved.

Note: The following are the allowable values for the Branch Option. They are defined in an include file built into the project.

- `BRANCH_FILTER_TRUNK_ONLY`-only get trunk versions
- `BRANCH_FILTER_BRANCH_ONLY`-only get branch versions
- `BRANCH_FILTER_UNMERGED_ONLY`-only get unmerged branches
- `BRANCH_FILTER_MERGED_ONLY`-only get merged branches
- `BRANCH_FILTER_TRUNK_AND_BRANCH`-get trunk and branched versions

Owning Class

CaVersionChooser

Arguments

HINT32 iOption-the Branch Option

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetBranchOption(BRANCH_FILTER_TRUNK_AND_BRANCH);
```

More information:

- HSDK Programming Fundamentals
- CaContext

void SetVersionOption(HINT32 iOptions)

Sets the Version Option to specify whether to get only the latest or multiple versions of the same item.

Note: The following are the allowable values for the Version Option. They are defined in an include file built into the project.

- VERSION_FILTER_ALL_IN_VIEW-get all version of an item in the view
- VERSION_FILTER_LATEST_IN_VIEW-get only the latest version of an item in the view
- VERSION_FILTER_LATEST_IN_PACKAGE-get only the latest version of an item in the package, set the package ID before calling this option
- VERSION_FILTER_ALL_IN_PACKAGE-get all versions of an item in the package, set the package ID before calling this option

Owning Class

CaVersionChooser

Arguments

HINT32 iOption-the Version Option

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetVersionOption(VERSION_FILTER_LATEST_IN_VIEW);
```

More information:

- HSDK Programming Fundamentals
- CaContext

void SetVersionItemOption(HINT32 iOptions)

Sets the Version Item Option to specify whether to base versions of an item, modified versions of an item, or both.

Note: The following are the allowable values for the Version Item Option. They are defined in an include file built into the project.

- VERSION_FILTER_ITEM_MODIFIED-get only modified versions of an item in the view
- VERSION_FILTER_ITEM_BASELINE-get only base versions of an item in the view
- VERSION_FILTER_ITEM_BOTH-get both base and modified versions of an item in the view

Owning Class

CaVersionChooser

Arguments

HINT32 iOption-the Version Item Option

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetVersionItemOption(VERSION_FILTER_ITEM_BOTH);
```

More information:

- HSDK Programming Fundamentals
- CaContext

void SetVersionNumber(LPCTSTR szVersion)

Sets the version number to be selected by version chooser.

Note: Must be used with SetVersionOption(VERSION_FILTER_ALL_IN_VIEW).

Owning Class

CaVersionChooser

Arguments

LPCTSTR szVersion - The version number as a string

Return Value

None

Example

```
CaVersionChooser& rVerChoose= rContext.GetVersionChooser();  
rVerChoose.SetVersionNumber("3.1.1");
```

More information:

- HSDK Programming Fundamentals
- SetVersionOption.

void SetVersionStatusOption(HINT32 iOptions)

Sets the Version Status Option to specify what types of versions to retrieve.

Note: The following are the allowable values for the Version Status Option. They are defined in an include file built into the project.

- VERSION_FILTER_NORMAL_VERSION-get only versions with a Normal tag
- VERSION_FILTER_RESERVED_VERSION-get only versions with a Reserve tag
- VERSION_FILTER_MERGED_VERSION-get only versions with a Merge tag
- VERSION_FILTER_DELETED_VERSION-get only versions with a Delete tag
- VERSION_FILTER_FROZEN_VERSION-get only versions with a Frozen tag
- VERSION_FILTER_ALL_TAG-get versions with any tag
- VERSION_FILTER_ANY_TAG-get versions with any tag except a Normal tag

Owning Class

CaVersionChooser

Arguments

HINT32 iOption-the Version Status Option

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetVersionStatusOption(VERSION_FILTER_ALL_TAG);
```

More information:

- HSDK Programming Fundamentals
- CaContext

void SetVersionDateOption(HINT32 iOptions)

Sets the Version Date Option to specify date ranges.

Note: The following are the allowable values for the Version Date Option. They are defined in an include file built into the project.

- VERSION_OPTION_DATE_ALL-don't filter on dates
- VERSION_OPTION_DATE_BETWEEN-get only versions created between the from and to dates

Owning Class

CaVersionChooser

Arguments

HINT32 iOption-the Version Date Option

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetVersionDateOption(VERSION_OPTION_DATE_BETWEEN);
```

More information:

- HSDK Programming Fundamentals
- CaContext
- SetFromDate
- SetToDate

void SetVersionObjId(HAROBJID iVersionObjId, int iIndex)

Sets one in a set of Version Object Ids to search for.

Note: The set of ids is stored in an ordered array. The index value sets a specific object ID value location. Index locations must be set in numerical order.

Owning Class

CaVersionChooser

Arguments

HAROBJID iVersionObjId-version Object Id

int iIndex-the set of IDs is stored in an indexed array. Defaults to zero

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser(); rVersionChooser.  
SetVersionObjId (12345);
```

More information:

- HSDK Programming Fundamentals
- CaContextvoid
- SetFromDate(CaTimeStamp& Time)

CaContextvoid SetFromDate(CaTimeStamp& Time

Sets the From Date value of the date range.

Note: Only used if the Version Date Option is set.

Owning Class

CaVersionChooser

Arguments

CaTimeStamp& Time-the time stamp value

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetFromDate(TimeStamp);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- CaTimeStamp
- SetVersionDateOption
- SetToDate

void SetToDate(CaTimeStamp& Time)

Sets the To Date value of the date range.

Note: Only used if the Version Date Option is set.

Owning Class

CaVersionChooser

Arguments

CaTimeStamp& Time-the time stamp value

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetToDate(TimeStamp);
```

■ More information:

- HSDK Programming Fundamentals
- CaContext
- CaTimeStamp
- SetVersionDateOption
- SetFromDate

CaContainer GetVersionList(void) const

Retrieves a CaContainer object that contains the attributes for each CA Harvest SCM Version in the list.

Note: List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME
- CA_ATTRKEY_PACKAGEID
- CA_ATTRKEY_ITEMID
- CA_ATTRKEY_ITEM_TYPE
- CA_ATTRKEY_PARENT_VERSIONID
- CA_ATTRKEY_MERGED_VERSIONID
- CA_ATTRKEY_MAPPED_VERSION_NAME
- CA_ATTRKEY_VERSION_STATUS
- CA_ATTRKEY_PACKAGE_NAME

- CA_ATTRKEY_FULL_PATH_NAME
- CA_ATTRKEY_FULL_NAME
- CA_ATTRKEY_CLIENT_MACHINE
- CA_ATTRKEY_CLIENT_PATH
- CA_ATTRKEY_DESCRIPTION
- CA_ATTRKEY_VERSIONDATASIZE

Owning Class

CaVersionChooser

Arguments

None

Return Value

Returns a CaContainer object

Exceptions

Throws CaException on error.

Example

```
CaContainer Container = rVersionChooser.GetVersionList ();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaContainer

CaVersionList& GetVersionListAsList(void)

Retrieves the version list.

Owning Class

CaVersionChooser

Arguments

None

Return Value

Returns a CaVersionList object.

Exceptions

Throws CaException on error.

Example

```
CaVersionList& rList = rVersionChooser.GetVersionListAsList();
```

More information:

- HSDK Programming Fundamentals
- CaException
- CaVersionList
- CaVersion

void Execute(void)

Executes the VersionChooser based on the set of version selection criteria set in the version chooser attributes.

Owning Class

CaVersionChooser

Arguments

None

Return Value

None

Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.Execute();
```

More information:

- HSDK Programming Fundamentals
- CaContext

Chapter 3: HUtilities Class Descriptions

This section contains the following topics:

[HUtilities Programming Fundamentals](#) (see page 477)

[CaBlob](#) (see page 478)

[CaContainer](#) (see page 482)

[CaData](#) (see page 491)

[CaDataArray](#) (see page 500)

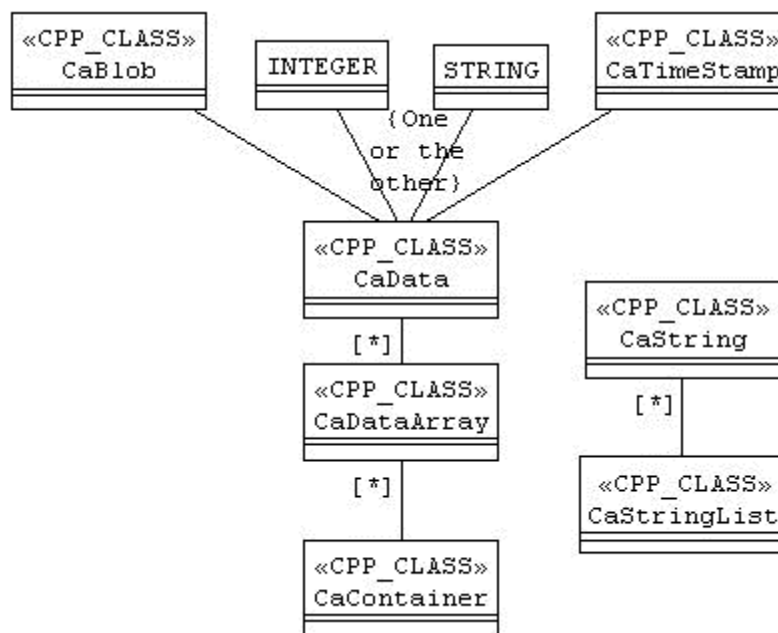
[CaString](#) (see page 503)

[CaStringList](#) (see page 516)

[CaTimeStamp](#) (see page 528)

HUtilities Programming Fundamentals

HUtilities is a lower-level CA Harvest SCM module. It defines a set of cross-platform classes that encapsulate data and data containers used throughout CA Harvest SCM. A subset of that class set is exported as a set of encapsulation classes. The encapsulation removes the need for HSDK users to include the Standard Template Library (STL) in HSDK applications.



More information:

For more information, see HSDK Programming Fundamentals in the chapter “Using the Software Development Kit.”

CaBlob

CaBlob objects contain binary data. A blob does not own its data. The default constructor is private so that an empty blob cannot be created.

Public Methods

- CaBlob
- GetSize
- GetData
- Operator=
- Operator==
- Operator!=
- operator<<

void CaBlob(const CaBlob& rBlob)

Copy constructor sets one blob to another. They both point to the same data.

Owning Class

CaBlob

Arguments

const CaBlob& rBlob - the blob to copy

Return Value

None

Notes

Blob does not own this data.

Example

```
CaBlob b1(b2);
```

More information:

- HUtilities Programming Fundamentals

void CaBlob(HINT32 nSize, void * lpData)

This method is a constructor.

Owning Class

CaBlob

Arguments

HINT32 nSize - size in bytes of data to be pointed to

void * lpData - data pointer to be stored

Return Value

None

Notes

Blob does not own this data.

Example

```
CaBlob b1(5, ptr);
```

More information:

- HUtilities Programming Fundamentals

HINT32 GetSize(void) const

Retrieves size in bytes of data stored.

Owning Class

CaBlob

Arguments

None

Return Value

HINT32 - number of bytes stored.

Notes

Blob does not own this data.

Example

```
CaBlob b1(5, ptr); HINT32 iSize = b1.GetSize();
```

More information:

- HUtilities Programming Fundamentals
- GetData

void * GetData(void) const

Retrieves the data pointer stored in the object.

Owning Class

CaBlob

Arguments

None

Return Value

void* - pointer stored in blob

Notes

Blob does not own this data.

Example

```
CaBlob b1(5, ptr); void* pData = b1.GetData();
```

More information:

- HUtilities Programming Fundamentals
- GetSize

CaBlob& operator=(const CaBlob &blob)

Assignment operator. Copies one data object to another.

Owning Class

CaBlob

Arguments

const CaBlob &aData - the object to copy

Return Value

CaBlob & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals

bool operator==(const CaBlob &rSrc)

Compares the blob with the specified object.

Owning Class

CaBlob

Arguments

const CaBlob &rSrc

Return Value

bool - true if equal, false if not

More information:

- HUtilities Programming Fundamentals

bool operator!=(const CaBlob &rSrc)

Compares the blob with the specified object.

Owning Class

CaBlob

Arguments

const CaBlob &rSrc

Return Value

bool - false if equal, true if not

More information:

- HUtilities Programming Fundamentals

friend ostream& operator<<(ostream &o, const CaBlob &rBlob)

Writes a description of the contents onto an ostream.

Owning Class

CaBlob

Arguments

ostream &o - Ostream on which to write

CaBlob &Blob - reference to a blob

Return Value

ostream & - reference to the ostream

Notes

This is normally used for debugging purposes.

Example

```
CaBlob b(5, ptr); cout << b;
```

More information:

- HUtilities Programming Fundamentals

CaContainer

CaContainer is an STL hash-map of CaDataArray objects. Strings are used as the hash-map key values. CaData objects are stored as elements in the CaDataArray. CaData objects provide a type-safe container for LPCTSTR, CaTimeStamp, HINT32, and CaBlob data types.

Public Methods

- Clear
- Props
- GetKeyCount
- GetKeyList

- GetKeyElementCount
- IsKeyExist
- IsEmpty
- FindData
- IsValid
- operator=
- operator+=
- operator<<

More information:

- CaDataArray
- CaData
- CaString

void Clear(void)

Empties the container.

Owning Class

CaContainer

Arguments

None

Return Value

None

Example

```
CaContainer bag; bag.clear();
```

More information:

- HUtilities Programming Fundamentals

CaData & Props(LPCTSTR strKey, HINT32 iIndex)

Retrieves a referent to an object within the container. The contents of the object can be read or altered.

Owning Class

CaContainer

Arguments

LPCTSTR strKey - the key value of the CaDataArray to locate

HINT32 iIndex- the location within the CaDataArray where the CaData object resides

Return Value

CPtrHData & - a referent to the located object

Notes

iIndex defaults to zero.

A lower-level exception occurs if the key is not found or if the index is out of range.

Example

```
CaContainer bag; CaString = (LPCTSTR)bag.Props("MyKey", 2);
```

More information:

- HUtilities Programming Fundamentals
- IsKeyExist
- GetKeyElementCount
- CaData

HUINT32 GetKeyCount(void) const

Retrieves the quantity of hash-map keys currently in the container.

Owning Class

CaContainer

Arguments

None

Return Value

HUINT32 - number of keys as an unsigned 32-bit integer

Example

```
CaContainer bag; HUINT32 iNumKeys = bag.GetKeyCount();
```

More information:

- HUtilities Programming Fundamentals
- IsKeyExist
- GetKeyElementCount

CaStringList GetKeyList(void) const

Retrieves a list of the hash-map keys currently in the container.

Owning Class

CaContainer

Arguments

None

Return Value

CaStringList - The key list

Example

```
CaContainer bag; CaStringList list = bag.GetKeyList();
```

More information:

- HUtilities Programming Fundamentals
- CaStringList
- CaString

HUINT32 GetKeyElementCount(LPCTSTR strKey) const

Retrieves the quantity of CaData objects in the CaDataArray indexed by the specified hash-map key value.

Owning Class

CaContainer

Arguments

LPCTSTR strKey - the key value to check

Return Value

HUINT32 - number of CaData objects as an unsigned 32-bit integer.

Example

```
CaContainer bag; HUINT32 iNumElements = bag.GetKeyElementCount("MyKey");
```

More information:

- HUtilities Programming Fundamentals
- IsKeyExist
- CaDataArray
- CaData

bool IsKeyExist(LPCTSTR strKey) const

Checks for the existence of the specified key value.

Owning Class

CaContainer

Arguments

LPCTSTR strKey - the key value to check

Return Value

bool - true if it exists, false if not

Example

```
CaContainer bag; bool bHaveKey = bag.IsKeyExist("MyKey");
```

More information:

- HUtilities Programming Fundamentals
- Props

bool IsEmpty(void) const

Checks if the container is empty.

Owning Class

CaContainer

Arguments

None

Return Value

bool - true if it empty, false if not

Example

```
CaContainer bag; bool bHaveKey = bag.IsEmpty();
```

More information:

- HUtilities Programming Fundamentals

CaDataArray FindData(LPCTSTR strKey, LPCTSTR strValue) const

Searches all elements indexed by the specified key value for any whose value matches the supplied string. Returns a set of indexes to CaData objects that match the values.

Owning Class

CaContainer

Arguments

LPCTSTR strKey - Key value

LPCTSTR strValue - string to search for

Return Value

CaDataArray - array of matching indexes

Example

```
CaContainer bag; CaDataArray a = bag.FindData(CA_ATTRKEY_NAME,  
(LPCTSTR)gszProjectName); HAROBJID iObjid =  
(HINT32)bag.Props(CA_ATTRKEY_OBJID,a[0]);
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaDataArray
- CaData

CaDataArray FindData(LPCTSTR strKey, HINT32 iValue) const

Searches all elements indexed by the specified key value for any whose value matches the supplied string. Returns a set of indexes to CaData objects that match the values.

Owning Class

CaContainer

Arguments

LPCTSTR strKey - key value

HINT32 iValue - integer value to search for

Return Value

CaDataArray - array of matching indexes

Example

```
CaContainer bag; CaDataArray a = bag.FindData(CA_ATTRKEY_PRIORITY, 1); HINT32  
iPriority = (HINT32)bag.Props(CA_ATTRKEY_PRIORITY,a[0]);
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaDataArray
- CaData

CaContainer & operator=(const CaContainer &CRight)

Clears the left-hand container and then copies all data from the right-hand container into it.

Owning Class

CaContainer

Arguments

CaContainer &CRight - reference to the right-hand container

Return Value

CaContainer & - reference to the left-hand container

Example

```
CaContainer bag; CaContainer bag2; bag2.Props("mykey", 0) = 3; bag = bag2;
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaDataArray
- CaData

CaContainer & operator+=(const CaContainer &CRight)

Appends the right-hand container to the left-hand container by copying all data from the right-hand container into it. Any left key values not in the right container are added. Any duplicate keys have the right elements added after the left elements.

Owning Class

CaContainer

Arguments

CaContainer &CRight - reference to the right-hand container

Return Value

CaContainer & - reference to the left-hand container

Example

```
CaContainer bag; CaContainer bag2; bag2.Props("mykey", 0) = 3; bag += bag2;
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaDataArray
- CaData

friend ostream& operator<<(ostream &o, const CaContainer &c)

Writes the key values and their associated data values onto an ostream.

Owning Class

CaContainer

Arguments

ostream &o - Ostream on which to write

CaContainer &c - reference to a container

Return Value

ostream & - reference to the ostream

Notes

This is normally used for debug purposes.

Example

```
CaContainer bag; bag.Props("mykey", 0) = 3; cout << bag;
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaDataArray
- CaData

CaData

CaData objects provide a type-safe container for LPCTSTR, CaTimeStamp, HINT32, and CaBlob data types. They have no type until set. After they are set the value can only be retrieved as that same type. After set they can be overwritten by the same data type but no other.

Public Methods

- GetType
- IsNull
- SetNull
- Operator=
- Operator==
- Operator!=
- Operator HINT32()
- Operator LPCTSTR()
- Operator CaString()
- Operator (CaTimeStamp&)()
- Operator (CaBlob&)()
- operator<<

More information:

- CaString
- CaTimeStamp
- CaBlob

void SetNull(void)

Sets the Data object's value to NULL but does not change its type.

Owning Class

CaData

Arguments

None

Return Value

None

Example

```
CaData d; d.SetNull();
```

More information:

- HUtilities Programming Fundamentals
- CaString
- CaTimeStamp
- CaBlob
- IsNull

bool IsNull(void) const

Checks to see whether the value of the CaData object is NULL.

Owning Class

CaData

Arguments

None

Return Value

bool - true if NULL, false if not

Example

```
CaData d; bool bResult = d.IsNull();
```

More information:

- HUtilities Programming Fundamentals
- CaString
- CaTimeStamp
- CaBlob
- SetNull

CaData::Types GetType(void) const

Retrieves the data type of the object.

Owning Class

CaData

Arguments

None

Return Value

CaData::Types - data type of object

Notes

CaData::Types is an enumeration that can have the following values:

NO_TYPE

Value has not been set.

INTEGER

Integer value accessed as HINT32.

BLOB

Binary Large Object value accessed as CaBlob.

TIMESTAMP

TimeStamp object accessed as CaTimeStamp, LPCTSTR, or CaString.

Example

```
CaData a; a = 3; CaData::Types iType = a.GetType();
```

More information:

- HUtilities Programming Fundamentals
- CaString
- CaTimeStamp
- CaBlob

CaData& operator=(const CaData &aData)

Copies one data object to another.

Owning Class

CaData

Arguments

const CaData &aData - the object to copy

Return Value

CaData & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals

CaData& operator=(HINT32 nValue)

Sets the data value to the specified integer value.

Owning Class

CaData

Arguments

HINT32 nValue

Return Value

CaData & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals

CaData& operator=(LPCTSTR szValue)

Sets the data value to the specified string value.

Owning Class

CaData

Arguments

LPCTSTR szValue

Return Value

CaData & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals

CaData& operator=(const CaBlob& rBlob)

Sets the data value to the specified blob value.

Owning Class

CaData

Arguments

const CaBlob& rBlob

Return Value

CaData & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals
- CaBlob

CaData& operator=(const CaTimeStamp& rTimeStamp)

Sets the data value to the specified time stamp value.

Owning Class

CaData

Arguments

const CaTimeStamp& rTimeStamp

Return Value

CaData & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals
- CaTimeStamp

bool operator==(const CaData &rSrc)

Compares the data object with the specified object.

Owning Class

CaData

Arguments

const CaData &rSrc

Return Value

bool - true if equal, false if not

More information:

- HUtilities Programming Fundamentals

bool operator!=(const CaData &rSrc)

Compares the data object with the specified object.

Owning Class

CaData

Arguments

const CaData &rSrc

Return Value

bool - false if equal, true if not

More information:

- HUtilities Programming Fundamentals

operator HINT32()

Conversion operator that returns the value of the CaData object as an HINT32 object.

Owning Class

CaData

Arguments

None

Return Value

None

Notes

Errors if CaData value is not of type INTEGER.

More information:

- HUtilities Programming Fundamentals

operator LPCTSTR()

Conversion operator that returns the value of the CaData object as an LPCTSTR object.

Owning Class

CaData

Arguments

None

Return Value

None

Notes

Errors if CaData value is not of type STRING or TIMESTAMP.

More information:

- HUtilities Programming Fundamentals

operator CaString()

Conversion operator that returns the value of the CaData object as a CaString object.

Owning Class

CaData

Arguments

None

Return Value

None

Notes

Errors if CaData value is not of type STRING or TIMESTAMP.

More information:

- HUtilities Programming Fundamentals
- CaString

operator (CaBlob&)()

Conversion operator that returns the value of the CaData object as blob object.

Owning Class

CaData

Arguments

None

Return Value

None

Notes

Errors if CaData value is not of type BLOB.

More information:

- HUtilities Programming Fundamentals
- CaBlob

operator (CaTimeStamp&)()

Conversion operator that returns the value of the CaData object as timestamp object.

Owning Class

CaData

Arguments

None

Return Value

None

Notes

Errors if CaData value is not of type TIMESTAMP.

More information:

- HUtilities Programming Fundamentals
- CaTimeStamp

friend ostream& operator<<(ostream &o, const CaData &d)

Writes the array values onto an ostream.

Owning Class

CaData

Arguments

ostream &o - ostream on which to write
CaData &d - reference to a Data object

Return Value

ostream & - reference to the ostream

Notes

This is normally used for debugging purposes.

Example

```
CaData d; d = 3; cout << d;
```

More information:

- HUtilities Programming Fundamentals

CaDataArray

CaDataArray is an STL vector of CaData objects. CaData objects are stored as elements in the CaDataArray. CaData objects provide a type-safe DataArray for LPCTSTR, CaTimeStamp, HINT32, and CaBlob data types.

Public Methods

- clear
- size
- Find
- Operator[]
- operator<<

More information:

- CaData

void Clear(void)

Empties the DataArray.

Owning Class

CaDataArray

Arguments

None

Return Value

None

Example

```
CaDataArray bag; bag.clear();
```

More information:

- HUtilities Programming Fundamentals

int GetSize(void) const

Retrieves the quantity of CaData objects in the array.

Owning Class

CaDataArray

Arguments

None

Return Value

int - number of objects in array

Example

```
CaDataArray bag; int iSize = bag.size();
```

More information:

- HUtilities Programming Fundamentals

bool Find (const CaData& rData, HUINT32& rIndex) const

Searches all elements for one whose value matches the supplied object. Returns the array index of the found object in an argument.

Owning Class

CaDataArray

Arguments

const CaData& rData - object to search for

HUINT32& rIndex - location to place array index if found

Return Value

bool - true if found, false if not

Notes

If not found, then rIndex is not changed.

Example

```
CaDataArray bag; HUINT32 iIndex; bag[0] = 3; bool bResult = bag.Find(bag[0],iIndex);
```

More information:

- HUilities Programming Fundamentals
- Props
- CaData

CaData & operator[](HUINT32 index)

Retrieves the indexed CaData object.

Owning Class

CaDataArray

Arguments

HUINT32 index - index value

Return Value

CaData & - reference to the indexed object

Example

```
CaDataArray bag; bag[0] = 3; CaData& rData = bag[0];
```

More information:

- HUilities Programming Fundamentals
- Props
- CaDataArray
- CaData

friend ostream& operator<<(ostream &o, const CaDataArray &a)

Writes the key values and their associated data values onto an ostream.

Owning Class

CaDataArray

Arguments

ostream &o - ostream on which to write

CaDataArray &a - reference to a DataArray

Return Value

ostream & - reference to the ostream

Notes

This is normally used for debugging purposes.

Example

```
CaDataArray bag; bag[0] = 3; cout << bag;
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaData

CaString

The CaString class provides various string manipulation routines. An important feature is that strings encapsulate conversion between Unicode and non-Unicode strings when compiled with Unicode turned on.

Public Methods

- CaString
- Append
- ToUpper
- ToLower
- IsEmpty
- MakeEmpty
- GetLength
- Trim

- IsValid
- GetMultiByteString
- Operator=
- Operator+=
- Operator-=
- Operator==
- Operator!=
- Operator LPCTSTR()
- Operator char* ()
- operator<<

More information:

- HUtilities Programming Fundamentals

CaString(void)

Default constructor. Initialize new string value to NULL.

Owning Class

CaString

Arguments

None

Return Value

None

Notes

Value is set to NULL.

Example

```
CaString d;
```

More information:

- HUtilities Programming Fundamentals

CaString(const CaString &src)

Copy Constructor - Initializes new string with source string.

Owning Class

CaString

Arguments

const CaString &src - value for new string

Return Value

None

Example

```
CaString d("My String"); CaString newString(d);
```

More information:

- HUtilities Programming Fundamentals

CaString(LPCTSTR lpStr)

Constructor - Initialize new string with this value.

Owning Class

CaString

Arguments

LPCTSTR lpStr - value for new string

Return Value

None

Notes

LPCTSTR is a Unicode string if compiled with Unicode and non-Unicode if not.

Example

```
CaString d(_T("My String"));
```

More information:

- HUtilities Programming Fundamentals

CaString(char* lpStr)

Constructor - Initializes new string with this value.

Owning Class

CaString

Arguments

char* lpStr - value for new string

Return Value

None

Notes

char* is a non-Unicode string regardless of how it is compiled.

Example

```
CaString d("My String");
```

More information:

- HUtilities Programming Fundamentals

CaString& Append(LPCTSTR src)

Appends a string to this one.

Owning Class

CaString

Arguments

LPCTSTR src - string to append

Return Value

CaString& - reference to this string

Example

```
CaString d("My String"); d.Append(_T("Appended String"));
```

More information:

- HUtilities Programming Fundamentals

CaString& ToUpper (void)

Converts this string to upper case.

Owning Class

CaString

Arguments

None

Return Value

CaString& - reference to this string

Example

```
CaString d("My String"); d.ToUpper();
```

More information:

- HUtilities Programming Fundamentals

CaString& ToLower (void)

Converts this string to lower case.

Owning Class

CaString

Arguments

None

Return Value

CaString& - reference to this string

Example

```
CaString d("My String"); d.ToLower();
```

More information:

- HUtilities Programming Fundamentals

bool IsEmpty(void) const

Determines if value is set to NULL, an empty string, or data.

Owning Class

CaString

Arguments

None

Return Value

bool - true if NULL or empty, false if length > 0

Example

```
CaString d("My String"); bool bResult = d.IsEmpty();
```

More information:

- HUtilities Programming Fundamentals

CaString& MakeEmpty(void)

Sets string value to NULL.

Owning Class

CaString

Arguments

None

Return Value

CaString& - reference to this string

Example

```
CaString d("My String"); d.MakeEmpty();
```

More information:

- HUtilities Programming Fundamentals

long GetLength(void) const

Returns the number of characters in this string.

Owning Class

CaString

Arguments

None

Return Value

long - string length

Notes

This function returns number characters not number of bytes. If Unicode, then each character is two bytes wide.

If string value is NULL or the empty string then return value is zero.

Example

```
CaString d("My String"); long length = d.GetLength();
```

More information:

- HUtilities Programming Fundamentals

CaString& Trim(void)

Removes white space from both ends of the string.

Owning Class

CaString

Arguments

None

Return Value

CaString& - reference to this string

Example

```
CaString d(" My String "); d.Trim();
```

More information:

- HUtilities Programming Fundamentals

bool IsValid(void) const

Determines whether the pointer to this string is NULL or not.

Owning Class

CaString

Arguments

None

Return Value

bool - true if valid, false if not

Notes

This function checks the value of the “this” pointer, not the string value.

Example

```
CaString d(" My String "); bool bResult = d.IsValid();
```

More information:

- HUtilities Programming Fundamentals

const char* GetMultiByteString(void)

If compiled with Unicode this returns the non-Unicode version of the data.

Owning Class

CaString

Arguments

None

Return Value

const char* - pointer to converted string

Notes

If not compiled with Unicode, no conversion is required and result is same.

Example

```
CaString d(_T(" My String ")); const char* p = d.GetMultiByteString();
```

More information:

- HUtilities Programming Fundamentals

CaString& operator=(const CaString &src)

Copies one object to another.

Owning Class

CaString

Arguments

const CaString &src - the object to copy

Return Value

CaString & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals

CaString& operator=(LPCTSTR szSrc)

Sets the data value to the specified string value.

Owning Class

CaString

Arguments

LPCTSTR szSrc

Return Value

CaString & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals

CaString& operator=(char* szSrc)

Sets the data value to the specified string value.

Owning Class

CaString

Arguments

LPCTSTR szSrc

Return Value

CaString & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals

bool operator==(const CaString &rSrc)

Compares the data object with the specified object.

Owning Class

CaString

Arguments

const CaString &rSrc

Return Value

bool - true if equal, false if not

More information:

- HUtilities Programming Fundamentals

bool operator==(LPCTSTR src) const

Compares the string value with the specified string.

Owning Class

CaString

Arguments

LPCTSTR src

Return Value

bool - true if equal, false if not

More information:

- HUtilities Programming Fundamentals

bool operator!=(const CaString &rSrc) const

Compares the object with the specified object.

Owning Class

CaString

Arguments

const CaString &rSrc

Return Value

bool - false if equal, true if not

More information:

- HUtilities Programming Fundamentals

bool operator!=(LPCTSTR src) const

Compares the string value with the specified string.

Owning Class

CaString

Arguments

LPCTSTR src

Return Value

bool - false if equal, true if not

More information:

- HUtilities Programming Fundamentals

CaString& operator+=(LPCTSTR src)

Appends a string to this one.

Owning Class

CaString

Arguments

LPCTSTR src - string to append

Return Value

CaString& - reference to this string

Example

```
CaString d("My String"); d += _T("Appended String");
```

More information:

- HUtilities Programming Fundamentals

CaString& operator-=(LPCTSTR src)

Subtracts a string from this one.

Owning Class

CaString

Arguments

LPCTSTR src - string to subtract

Return Value

CaString& - reference to this string

Notes

Only removes string if it is at the end of the left-hand string.

Example

```
CaString d("My String"); d -= _T("String");
```

More information:

- HUtilities Programming Fundamentals

operator LPCTSTR()

Conversion operator that returns the value of the CaString object as an LPCTSTR object.

Owning Class

CaString

Arguments

None

Return Value

LPCTSTR

Notes

If value is NULL returns empty string.

More information:

- HUtilities Programming Fundamentals

operator char *()

Conversion operator that returns the value of the CaString object as a char* object.

Owning Class

CaString

Arguments

None

Return Value

char*

Notes

If value is NULL, returns empty string.

More information:

- HUtilities Programming Fundamentals
- CaString

friend ostream& operator<<(ostream &o, const CaString &rString)

Writes string onto an ostream.

Owning Class

CaString

Arguments

ostream &o - ostream on which to write

CaString &rString - reference to a String object

Return Value

ostream & - reference to the ostream

Notes

If compiled in Unicode, this uses char* operator.

Example

```
CaString d("My String"); cout << d;
```

More information:

- HUtilities Programming Fundamentals

CaStringList

The CaStringList class is a container class to hold CaString objects.

Public Methods

- CaStringList
- Add

- Find
- Remove
- Clean
- GetSize
- IsEmpty
- GetAllNames
- GetIndex
- Operator=
- Operator+=
- Operator==
- Operator!=
- operator bool()
- operator[]
- operator<<

More information:

- HUtilities Programming Fundamentals
- CaString

CaStringList(void)

Default constructor. Initializes empty list.

Owning Class

CaStringList

Arguments

None

Return Value

None

Example

```
CaStringList d;
```

More information:

- HUtilities Programming Fundamentals

CaStringList(const CaStringList &src)

Copy Constructor - Initializes new string list from source string list.

Owning Class

CaStringList

Arguments

const CaStringList &src - source list

Return Value

None

Example

```
CaStringList d; CaStringList newStringList(d);
```

More information:

- HUtilities Programming Fundamentals

bool Add(const CaString szString)

Adds the string to the list.

Owning Class

CaStringList

Arguments

const CaString szString - string to add

Return Value

bool - true if success, false if fails

Notes

If string is not valid then it fails.

Duplicates are not added, but function still returns true.

Example

```
CaString s("My String"); CaStringList d; bool bResult = d.Add(s);
```

More information:

- HUtilities Programming Fundamentals

CaString & Find(const CaString szString)

Search for the specified string in the list.

Owning Class

CaStringList

Arguments

const CaString szString - string to search for

Return Value

CaString& - reference to found string

Notes

Returns NULL string reference if not found. Use CaString::IsValid() to test results of find.

Example

```
CaStringList d; CaString s("My String"); CaString& rString = d.Find(s);
```

More information:

- HUtilities Programming Fundamentals

bool Remove(const CaString szString)

Find matching string in list and remove it.

Owning Class

CaStringList

Arguments

const CaString szString - string to remove

Return Value

bool - true if success, false if fails

Notes

Fails if string is not valid or not found in list.

Example

```
CaStringList d; CaString s("My String"); CaString& rString = d.Remove(s);
```

More information:

- HUtilities Programming Fundamentals

bool Add(LPCTSTR szString)

Adds the string to the list.

Owning Class

CaStringList

Arguments

LPCTSTR szString - string to add

Return Value

bool - true if success, false if fails

Notes

If string is not valid then it fails.

Duplicates are not added but function still returns true.

Example

```
CaStringList d; bool bResult = d.Add("My String");
```

More information:

- HUtilities Programming Fundamentals

CaString & Find(LPCTSTR szString)

Search for the specified string in the list.

Owning Class

CaStringList

Arguments

LPCTSTR szString - string to search for

Return Value

CaString& - reference to found string.

Notes

Returns NULL string reference if not found. Use CaString::IsValid() to test results of find.

Example

```
CaStringList d; bool bResult = d.Find("My String");
```

More information:

- HUtilities Programming Fundamentals

bool Remove(LPCTSTR szString)

Find matching string in list and remove it.

Owning Class

CaStringList

Arguments

LPCTSTR szString - string to remove

Return Value

bool - true if success, false if fails

Notes

Fails if string is not valid or not found in list.

Example

```
CaStringList d; bool bResult = d.Remove("My String");
```

More information:

- HUtilities Programming Fundamentals

bool IsEmpty(void) const

Determines if list is empty.

Owning Class

CaStringList

Arguments

None

Return Value

bool - true if NULL or empty, false if length > 0

Example

```
CaStringList d; bool bResult = d.IsEmpty();
```

More information:

- HUtilities Programming Fundamentals

void Clean(void)

Removes all elements from the list.

Owning Class

CaStringList

Arguments

None

Return Value

None

Example

```
CaStringList d; d.Clean();
```

More information:

- HUtilities Programming Fundamentals

int GetSize(void) const

Returns the number of elements in this list.

Owning Class

CaStringList

Arguments

None

Return Value

int - number of strings in list

Example

```
CaStringList d; int length = d.GetSize();
```

More information:

- HUtilities Programming Fundamentals

CaString GetAllNames(void) const

Concatenates all strings in the list into a new string separated by “\n\t”.

Owning Class

CaStringList

Arguments

None

Return Value

CaString - reference to this string

Example

```
CaStringList d; CaString szString = d.GetAllNames();
```

More information:

- HUtilities Programming Fundamentals

Int GetIndex(LPCTSTR szString) const

Finds and returns the index value for the string that matches szString.

Owning Class

CaStringList

Arguments

LPCTSTR szString - string to find

Return Value

int - index value for matching string

Example

```
CaStringList d; int iIndex = d.GetIndex("Matching String");
```

More information:

- HUtilities Programming Fundamentals

CaStringList& operator=(const CaStringList & CRight)

Copies one object to another.

Owning Class

CaStringList

Arguments

const CaStringList & CRight - the list to copy

Return Value

CaStringList & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals

bool operator==(const CaStringList & CRight) const

Compares the list with the specified list.

Owning Class

CaStringList

Arguments

const CaStringList & CRight

Return Value

bool - true if equal, false if not

More information:

- HUtilities Programming Fundamentals

bool operator!=(const CaStringList & CRight) const

Compares the list with the specified list.

Owning Class

CaStringList

Arguments

const CaStringList & CRight

Return Value

bool - false if equal, true if not

More information:

- HUtilities Programming Fundamentals

CaStringList& operator+=(const CaStringList & CRight)

Appends a string list to this one.

Owning Class

CaStringList

Arguments

const CaStringList & CRight - string list to append

Return Value

CaStringList& - reference to this string list

Notes

Duplicates are ignored.

Example

```
CaStringList d, e; d += e;
```

More information:

- HUtilities Programming Fundamentals

CaStringList& operator-=(const CaStringList & CRight)

Subtracts a string list from this one.

Owning Class

CaStringList

Arguments

const CaStringList & CRight - string list to subtract

Return Value

CaStringList& - reference to this string list

Notes

If right element is not found in left list it is ignored. Element order is irrelevant.

Example

```
CaStringList d; d -= e;
```

More information:

- HUtilities Programming Fundamentals

CaString & operator[] (int i)

Retrieves the string from the list at the specified index.

Owning Class

CaStringList

Arguments

int i - index into list

Return Value

CaString& - reference to the string found at the index

Notes

If index is out of range then a NULL string reference is returned. Use CaString::IsValid() to check results.

Example

```
CaStringList d; CaString& rS = d[0];
```

More information:

- HUtilities Programming Fundamentals

operator bool() const

Determines whether this is valid or not. If the “this” pointer is NULL list is invalid.

Owning Class

CaStringList

Arguments

None

Return Value

bool - true if valid, false if not

Example

```
CaStringList d; if(d) d.Add("MyString");
```

More information:

- HUtilities Programming Fundamentals

friend ostream& operator<<(ostream &o, const CaStringList &rList)

Writes strings from list onto an ostream.

Owning Class

CaStringList

Arguments

ostream &o - ostream on which to write

CaStringList &rList - reference to a String List

Return Value

ostream & - reference to the ostream

CaTimeStamp

CaTimeStamp objects contain date and time information.

Public Methods

- CaTimeStamp
- GetYear
- GetMonth
- GetDay
- GetHour
- GetMinute
- GetSecond
- GetFraction
- SetYear
- SetMonth
- SetDay
- SetHour
- SetMinute
- SetSecond
- SetFraction
- Operator=

- Operator==
- Operator!=
- operator<<

void CaTimeStamp(void)

Default constructor. Creates an empty TimeStamp.

Owning Class

CaTimeStamp

Arguments

None

Return Value

None

Example

```
CaTimeStamp t1;
```

More information:

- HUtilities Programming Fundamentals

void CaTimeStamp(const CaTimeStamp& rTimeStamp)

Copy constructor sets one TimeStamp to another.

Owning Class

CaTimeStamp

Arguments

const CaTimeStamp& rTimeStamp - the TimeStamp to copy

Return Value

None

Example

```
CaTimeStamp t1(t2);
```

More information:

- HUtilities Programming Fundamentals

void CaTimeStamp(HINT32 nYear, HINT32 nMonth, HINT32 nDay, HINT32 nHour, HINT32 nMinute, HINT32 nSecond, HINT32 nFraction)

Constructor. Sets initial values.

Owning Class

CaTimeStamp

Arguments

- HINT32 nYear
- HINT32 nMonth
- HINT32 nDay
- HINT32 nHour
- HINT32 nMinute
- HINT32 nSecond
- HINT32 nFraction

Return Value

None

Notes

nFraction defaults to zero.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4);
```

More information:

- HUtilities Programming Fundamentals

CaTimeStamp& SetYear(HINT32 iYear)

Sets Year value.

Owning Class

CaTimeStamp

Arguments

HINT32 iYear - Year value

Return Value

CaTimeStamp& - reference to this object so functions can be chained together

Example

```
CaTimeStamp t1; t1.SetYear(2001);
```

More information:

- HUtilities Programming Fundamentals
- GetYear

CaTimeStamp& SetMonth(HINT32 iMonth)

Sets Month value.

Owning Class

CaTimeStamp

Arguments

HINT32 iMonth - Month value

Return Value

CaTimeStamp& - reference to this object so functions can be chained together

Example

```
CaTimeStamp t1; t1.SetMonth(2001);
```

More information:

- HUtilities Programming Fundamentals
- GetMonth

CaTimeStamp& SetDay(HINT32 iDay)

Sets Day value.

Owning Class

CaTimeStamp

Arguments

HINT32 iDay - Day value

Return Value

CaTimeStamp& - reference to this object so functions can be chained together

Example

```
CaTimeStamp t1; t1.SetDay(2001);
```

More information:

- HUtilities Programming Fundamentals
- GetDay

CaTimeStamp& SetHour(HINT32 iHour)

Sets Hour value.

Owning Class

CaTimeStamp

Arguments

HINT32 iHour - Hour value

Return Value

CaTimeStamp& - reference to this object so functions can be chained together

Example

```
CaTimeStamp t1; t1.SetHour(2001);
```

More information:

- HUtilities Programming Fundamentals
- GetHour

CaTimeStamp& SetMinute(HINT32 iMinute)

Sets Minute value.

Owning Class

CaTimeStamp

Arguments

HINT32 iMinute - Minute value

Return Value

CaTimeStamp& - reference to this object so functions can be chained together

Example

```
CaTimeStamp t1; t1.SetMinute(2001);
```

More information:

- HUtilities Programming Fundamentals
- GetMinute

CaTimeStamp& SetSecond(HINT32 iSecond)

Sets Second value.

Owning Class

CaTimeStamp

Arguments

HINT32 iSecond - Second value

Return Value

CaTimeStamp& - reference to this object so functions can be chained together

Example

```
CaTimeStamp t1; t1.SetSecond(2001);
```

More information:

- HUtilities Programming Fundamentals
- GetSecond

CaTimeStamp& SetFraction(HINT32 iFraction)

Sets Fraction value.

Owning Class

CaTimeStamp

Arguments

HINT32 iFraction - Fraction value

Return Value

CaTimeStamp& - reference to this object so functions can be chained together

Example

```
CaTimeStamp t1; t1.SetFraction(2001);
```

More information:

- HUtilities Programming Fundamentals
- GetFraction

HINT32 GetYear(void) const

Gets Year value.

Owning Class

CaTimeStamp

Arguments

None

Return Value

HINT32 - Year value of TimeStamp

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iYear = t1.GetYear();
```

More information:

- HUtilities Programming Fundamentals
- SetYear

HINT32 GetMonth(void) const

Gets Month value.

Owning Class

CaTimeStamp

Arguments

None

Return Value

HINT32 - Month value of TimeStamp

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iMonth = t1.GetMonth();
```

More information:

- HUtilities Programming Fundamentals
- SetMonth

HINT32 GetDay(void) const

Gets Day value.

Owning Class

CaTimeStamp

Arguments

None

Return Value

HINT32 - Day value of TimeStamp

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iDay = t1.GetDay();
```

More information:

- HUtilities Programming Fundamentals
- SetDay

HINT32 GetHour(void) const

Gets Hour value.

Owning Class

CaTimeStamp

Arguments

None

Return Value

HINT32 - Hour value of TimeStamp

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iHour = t1.GetHour();
```

More information:

- HUtilities Programming Fundamentals
- SetHour

HINT32 GetMinute(void) const

Gets Minute value.

Owning Class

CaTimeStamp

Arguments

None

Return Value

HINT32 - Minute value of TimeStamp

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iMinute = t1.GetMinute();
```

More information:

- HUtilities Programming Fundamentals
- SetMinute

HINT32 GetSecond(void) const

Gets Second value.

Owning Class

CaTimeStamp

Arguments

None

Return Value

HINT32 - Second value of TimeStamp

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iSecond = t1.GetSecond();
```

More information:

- HUtilities Programming Fundamentals
- SetSecond

HINT32 GetFraction(void) const

Gets Fraction value.

Owning Class

CaTimeStamp

Arguments

None

Return Value

HINT32 - Fraction value of TimeStamp

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iFraction = t1.GetFraction();
```

More information:

- HUtilities Programming Fundamentals
- SetFraction

CPtHTimeStamp& ChangeTime(HINT32 iTimezone)

Converts time to specified time zone.

Owning Class

CaTimeStamp

Arguments

HINT32 iTimezone - Difference in hours between new time and GMT time

Return Value

CPtTimeStamp& - reference to this TimeStamp

Notes

Adjusts month and day as well.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); t1.ChangeTime(3);
```

More information:

- HUtilities Programming Fundamentals

CPtTimeStamp& ConvertGMTTime(void)

Converts time to specified time zone.

Owning Class

CaTimeStamp

Arguments

None

Return Value

CPtTimeStamp& - reference to this TimeStamp

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); t1.ConvertGMTTime();
```

More information:

- HUtilities Programming Fundamentals

CaTimeStamp& operator=(const CaTimeStamp &rTimeStamp)

Assignment operator. Copies one Time Stamp object to another.

Owning Class

CaTimeStamp

Arguments

const CaTimeStamp & rTimeStamp - the object to copy

Return Value

CaTimeStamp & - reference to left-hand object

More information:

- HUtilities Programming Fundamentals

bool operator==(const CaTimeStamp &rTimeStamp) const

Compares the TimeStamp with the specified object.

Owning Class

CaTimeStamp

Arguments

const CaTimeStamp &rTimeStamp

Return Value

bool - true if equal, false if not

More information:

- HUtilities Programming Fundamentals

bool operator!=(const CaTimeStamp &rTimeStamp) const

Compares the TimeStamp with the specified object.

Owning Class

CaTimeStamp

Arguments

const CaTimeStamp &rTimeStamp

Return Value

bool - false if equal, true if not

More information:

- HUtilities Programming Fundamentals

friend ostream& operator<<(ostream &o, const CaTimeStamp &rTimeStamp)

Writes a description of the contents onto an ostream.

Owning Class

CaTimeStamp

Arguments

ostream &o - ostream on which to write

CaTimeStamp &rTimeStamp - reference to a TimeStamp

Return Value

ostream & - reference to the ostream

Notes

This is normally used for debug purposes.

Example

```
CaTimeStamp b(5, ptr); cout << b;;
```

More information:

- HUtilities Programming Fundamentals

Chapter 4: Using the COM Interface of the Software Development Kit (CHSDK)

This section contains the following topics:

[Overview](#) (see page 541)

[CHSDK Programming Fundamentals](#) (see page 541)

Overview

The CHSDK is a set of C++ wrapper classes that encapsulate the persistence and manipulation of HSDK objects. It presents an object model that represents the data and their relationships that exist in the CA Harvest SCM Data Store. The CHSDK implements a set of client-side interfaces to access CA Harvest SCM data.

This document assumes a thorough knowledge of the CA Harvest SCM product and its usage.

CHSDK Messages

All CHSDK messages come from the HSDK layer.

Note: For more information on messages, see the *Messages Guide*.

CHSDK Programming Fundamentals

Usage

The CHSDK is available only for Windows platforms.

1. Install CA Harvest SCM prior to HSDK installation.
2. Install HSDK according to the installation instructions for the Windows platform.

Note: For HSDK installation instructions, see the *Implementation Guide*.

3. Register the CHSDK.dll (that is, regsvr32 chsdk.dll).

The CHSDK is now ready.

CHSDK Context

CHSDK operations are performed within a specified context. Once a user logs in to the principal CaWHarvest object, a CaWContext object is available. As the user changes the current context, CHSDK objects are reset or made available based on the new context. The CaWContext object owns the other CHSDK objects.

Object Lists and Security

The CHSDK provides a CaWContainer object. The CaWContainer is a text-keyed hash map of CaWdataArray objects.

CHSDK object security is maintained through the underlying CA Harvest SCM modules.

Chapter 5: CHSDK Class Descriptions

This section contains the following topics:

[CaWApprove](#) (see page 543)
[CaWCheckin](#) (see page 546)
[CaWCheckout](#) (see page 570)
[CaWConcurrentMerge](#) (see page 591)
[CaWContext](#) (see page 595)
[CaWCreatePackage](#) (see page 665)
[CaWCrossProjectMerge](#) (see page 671)
[CaWDeleteVersion](#) (see page 681)
[CaWDemote](#) (see page 683)
[CaWForm](#) (see page 687)
[CaWFormType](#) (see page 704)
[CaWFormAttachment](#) (see page 709)
[CaWHarvest](#) (see page 715)
[CaWInteractiveMerge](#) (see page 727)
[CaWPackage](#) (see page 749)
[CaWPackageChooser](#) (see page 751)
[CaWPromote](#) (see page 765)
[CaWRemoveItem](#) (see page 771)
[CaWSwitchPackage](#) (see page 778)
[CaWSQL](#) (see page 779)
[CaWTakeSnapshot](#) (see page 782)
[CaWUDP](#) (see page 786)
[CaWVersionChooser](#) (see page 796)

CaWApprove

CaWApprove encapsulates the HSDK Approve process. It cannot be instantiated directly. The Approve process can be set in the CaWContext object either by name or object ID. The CaWApprove object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the Project, State, and Approve Context.
- Select a set of Change Packages to Approve or Rejected using the CaWPackageChooser object.
- Set the Approve attributes.
- Invoke the Execute function.

Properties

- Approve
- Description

Methods

- Execute
- IsValid

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWPackageChooser

Approve Property

Gets or sets the Approve/Reject flag true or false.

Syntax

object.**Approve** = [value]

Owning Class

CaWApprove

Arguments

value - A boolean. If true Approve the Change Package list, if false Reject it.

Notes

The Approve/Reject flag defaults to true.

Example

```
Dim objApprove
Set objApprove = objContext.GetApprove()
objApprove.Approve = True
blnApprove = objApprove.Approve
```


Description Property

Gets or sets the Approve/Reject Description used to annotate why the package list was approved or rejected.

Syntax

object.**Description** = [value]

Owning Class

CaWApprove

Arguments

value - A string that sets the Approve/Reject description.

Notes

The Description defaults to a null string.

Example

```
Dim objApprove
Set objApprove = objContext.GetApprove()
objApprove.Description = "I approve this package"
strDescription = objApprove.Description
```

Execute Method

Executes the Approve process on the Package list currently selected using the Package Chooser.

Syntax

object.**Execute()**

Owning Class

CaWApprove

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Example

```
Dim objApprove
Set objApprove = objContext.GetApprove()
lngRet = objApprove.Execute()
```

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWApprove

Arguments

None

Return Value

A boolean - True if the *object* is a valid object, otherwise False.

Example

```
Dim objApprove
Set objApprove = objContext.GetApprove()
If objApprove.IsValid() <> 0 Then
    MsgBox "This is a valid object", vbOKonly, "Test Approve"
End If
```

CaWCheckin

CaWCheckin encapsulates the HSDK Checkin process. It cannot be instantiated directly. The Checkin process can be set in the CaWContext object either by name or object ID. The CaWCheckin object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the Project, State, Package, and Checkin Context.
- Select either a set of Versions to check in using the CaVersionChooser object or add a set of file names to a file name list, CaWStringList.
- Invoke the Execute function.

Properties

- RemoteMachineName
- AgentPortNumber
- RemoteUserName
- RemoteUserName

Methods

- SetCheckinMode
- SetPathOption
- SetItemOption
- SetClientDir
- SetViewPath
- SetDeleteAfterCI
- SetDescription
- SetFileList
- GetItemOption
- GetDescription
- GetCheckinMode
- GetNewItemOnBranch
- GetNewItemOnTrunk
- SetBranchOut
- GetBranchOut
- GetPathOption
- GetClientDir
- GetViewPath
- GetDeleteAfterCI
- GetFileList

- Execute
- IsValid

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWVersionChooser
- CaWStringList

RemoteMachineName Property

Gets or sets the host name of the remote machine where the files are located.

Syntax

object.**RemoteMachineName** = [*value*]

Owning Class

CaWCheckin

Arguments

value - A string that represents the machine name.

Notes

If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

A remote CA Harvest SCM File Agent must be running on the specified machine.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin
objCheckin.RemoteMachineName = "MyBuildMachine"
strRemoteMachineName = objCheckin.RemoteMachineName
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- SetViewPath
- SetPathOption
- AgentPortNumber
- RemoteUserName
- RemotePassword

AgentPortNumber Property

Gets or sets the port number of the remote machine where the files are located.

Syntax

object.AgentPortNumber = [*value*]

Owning Class

CaWCheckin

Arguments

value - A string that represents the machine name.

Notes

If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

A remote CA Harvest SCM File Agent must be running on the specified machine.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin
objCheckin.AgentPortNumber = "7001"
strAgentPortNumber = objCheckin.AgentPortNumber
```

More information:

[CHSDK Programming Fundamentals](#) (see page 541)

[CaWContext](#) (see page 595)

[SetClientDir Method](#) (see page 556)

[SetViewPath Method](#) (see page 557)

[SetPathOption MethodCaCheckin](#) (see page 554)

[RemoteMachineName Property](#) (see page 548)

[RemoteUserName Property](#) (see page 551)

[RemotePassword Property](#) (see page 552)

RemoteUserName Property

Gets or sets the user name for the login to the remote machine where the files are located.

Syntax

object.RemoteUserName = [value]

Owning Class

CaWCheckin

Arguments

value - A string that represents the user name.

Notes

If a remote computer is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote computer.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin
objCheckin.RemoteUserName = "George"
strRemoteUserName = objCheckin.RemoteUserName
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- RemoteMachineName
- AgentPortNumber
- SetViewPath
- SetPathOption
- RemotePassword

RemotePassword Property

Gets or sets the password for the username used to login to the remote machine where the files are located.

Syntax

object.RemotePassword = [value]

Owning Class

CaWCheckin

Arguments

value - A string that represents the password.

Notes

If a remote computer is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote computer.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin
objCheckin.RemoteUserName = "George"
strRemoteUserName = objCheckin.RemoteUserName
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- RemoteMachineName
- AgentPortNumber
- SetViewPath
- SetPathOption
- RemoteUserName

SetCheckinMode Method

Sets the Checkin Mode for this process.

Syntax

object.**SetCheckinMode**(*lngCheckinMode*)

Owning Class

CaWCheckin

Arguments

lngCheckinMode - A long that represents the Checkin Mode.

Return Value

Returns True if success, False if fail.

Notes

The following are the allowable values for the Checkin Mode:

0

Check in the file and release the lock.

1

Check in the file and reserve it again for update.

2

Release the lock but do not check in the file.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
BlnRet = objCheckin.SetCheckinMode(HAR_UPDATE_AND_RELEASE)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetCheckinMode

SetPathOption MethodCaCheckin

Sets the Path Option for this process. Files are checked into the repository in locations related to the view path, client directory, and the path option.

Syntax

object.**SetPathOption**(*IngOption*)

Owning Class

CaWCheckin

Arguments

IngOption - A long that represents the Path Option.

Return Value

Returns True if success, False if fail.

Notes

The following are the allowable values for the Path Option.

0

Checkin each new version to its existing view path location that mirrors the location in the client path.

1

Checkin each new version to its view path location that mirrors the location in the client path and create the view path if it does not yet exist.

2

Checkin each new version to the single view path location specified.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
blnRet = objCheckin.SetPathOption(CI_OPTION_PRESERVE_AND_CREATE_PATH)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetPathOption
- SetClientDir
- SetViewPath

SetItemOption Method

Sets the Item Option for this process.

Syntax

object.**SetItemOption**(*IngOption*)

Owning Class

CaWCheckin

Arguments

IngOption - A long that represents the Item Option.

Return Value

Returns True if success, False if fail.

Notes

The following are the allowable values for the Item Option:

0

Checkin the file whether or not it exists in the repository.

1

Checkin the file only if it does not yet exist in the repository.

2

Checkin the file only if it currently exists in the repository.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
blnRet = objCheckin.SetItemOption(CI_FILTER_NEW_OR_EXISTING_ITEMS)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetItemOption

SetClientDir Method

Sets the Client Directory anchor for this process.

Syntax

object.**SetClientDir**(*strCDir*)

Owning Class

CaWCheckin

Arguments

strCDir - A string that represents a valid directory.

Return Value

Returns True if success, False if fail.

Notes

This method is valid for file list checkin only; version list checkin does not use this method. If the checkin is from a remote machine then this directory is from the remote machine. (CA Harvest SCM stores the host location at checkout and uses these stored values for the version checkin. Version checkin may span many directories and remote machines and therefore cannot be driven by the SetClientDir location.)

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
BlnRet = objCheckin.SetClientDir("c:\temp\work")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetClientDir
- RemoteMachineName
- SetViewPath
- SetPathOption

SetViewPath Method

Sets the repository View Path anchor for this process.

Syntax

object.**SetViewPath**(strVPath)

Owning Class

CaWCheckin

Arguments

strVPath - A string that represents a valid repository view path.

Return Value

Returns True if success, False if fail.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
blnRet = objCheckin.SetViewPath("\rep\work")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- RemoteMachineName
- GetViewPath
- SetPathOption

SetDeleteAfterCI Method

Sets the Delete After Checkin attribute of this process.

Syntax

object.**SetDeleteAfterCI**(blnFlag)

Owning Class

CaWCheckin

Arguments

blnFlag - The flag value; True if delete the file, False if not.

Return Value

Returns True if success, False if fail.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
blnRet = objCheckin.SetDeleteAfterCI(False)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetDeleteAfterCI

SetDescription Method

Sets the Description to be associated with versions created by this Checkin.

Syntax

object.**SetDescription**(*strDesc*)

Owning Class

CaWCheckin

Arguments

strDesc - A string that represents the description.

Return Value

Returns True if success, False if fail.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
blnRet = objCheckin.SetDescription("Save work prior to vacation.")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetDescription

SetFileList Method

Sets the file list to be checked in by this process. If a remote machine name is set then these files must exist on the remote machine.

Syntax

object.**SetFileList**(objFileList)

Owning Class

CaWCheckin

Arguments

objFileList - A CaWStringList object that represents a list of file names to check in.

Return Value

None

Notes

If a remote computer is set, then a remote username and password must also be supplied. The Client Directory must exist on the remote computer.

Example

```
Dim objCheckin
Dim objFileList
Set objCheckin = objContext.GetCheckin()
Set objFileList = CreateObject("Chsdk.CaWStringList")
blnRet = objFileList.InitHObject()
blnRet = objFileList.Add("File1.vbs")
blnRet = objFileList.Add("File2.vbs")
objCheckin.SetFileList(objFileList)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetFileList
- SetClientDir
- RemoteMachineName
- SetViewPath
- SetPathOption
- RemoteUserName
- RemotePassword
- CaWStringList

GetItemOption Method

Gets the Item Option attribute for this process.

Syntax

```
object.GetItemOption()
```

Owning Class

CaWCheckin

Arguments

None

Return Value

A long that represents the value.

Example

```
Dim objCheckin  
Set objCheckin = objContext.GetCheckin()  
lngRet = objCheckin.GetItemOption()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetItemOption

GetDescription Method

Gets the Description attribute for this process.

Syntax

```
object.GetDescription()
```

Owning Class

CaWCheckin

Arguments

None

Return Value

A string that represents the description.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
strDescription = objCheckin.GetDescription()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetDescription

GetCheckinMode Method

Gets the Checkin Mode attribute for this process.

Syntax

object.**GetCheckinMode()**

Owning Class

CaWCheckin

Arguments

None

Return Value

A long that represents the value.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
lngCheckinMode = objCheckin.GetCheckinMode()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetCheckinMode

GetNewItemOnBranch Method

Retrieve the current setting for create versions for new items on a branch.

Syntax

```
object.GetNewItemOnBranch()
```

Owning Class

CaWCheckin

Arguments

None

Return Value

A Boolean that represents the current value of “CreateOnBranch” associated with the Checkin process.

Example

```
Dim objCheckin  
Set objCheckin = objContext.GetCheckin()  
blnNewItemOnBranch = objCheckin.GetNewItemOnBranch()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetNewItemOnTrunk

GetNewItemOnTrunk Method

Retrieve the current setting for create versions for new items on the trunk.

Syntax

```
object.GetNewItemOnTrunk()
```

Owning Class

CaWCheckin

Arguments

None

Return Value

A Boolean that represents the current value of “CreateOnTrunk” associated with the Checkin process.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
blnNewItemOnTrunk = objCheckin.GetNewItemOnTrunk()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetNewItemOnBranch

SetBranchOut Method

On execute, allow the creation of versions for new items on the trunk or a branch version.

Syntax

```
object.SetBranchOut(blnFlag)
```

Owning Class

CaWCheckin

Arguments

blnFlag - The flag value; True if check in new items on a branch, False if on the trunk.

Return Value

Returns True if success; False if fail.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
blnRet = objCheckin.SetBranchOut(False)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetBranchOut

GetBranchOut Method

Retrieve the current setting for whether the execute will create versions for new items on the trunk or on a branch.

Syntax

```
object.GetBranchOut()
```

Owning Class

CaWCheckin

Arguments

None

Return Value

A Boolean that represents the current value of “BranchOut” flag associated with the Checkin process.

Example

```
Dim objCheckin  
Set objCheckin = objContext.GetCheckin()  
blnBranchOut = objCheckin.GetBranchOut()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetBranchOut

GetPathOption Method

Gets the Path Option attribute for this process.

Syntax

```
object.GetPathOption()
```

Owning Class

CaWCheckin

Arguments

None

Return Value

A long that represents the value.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
lngPathOption = objCheckin.GetPathOption()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetPathOption

GetClientDir Method

Gets the Client Directory anchor for this process.

Syntax

object.**GetClientDir()**

Owning Class

CaWCheckin

Arguments

None

Return Value

A string that represents the client directory.

Notes

If a remote machine name is set then this is a directory on the remote machine.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
strClientDir = objCheckin.GetClientDir()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- RemoteMachineName

GetViewPath Method

Gets the repository view path anchor for this process.

Syntax

object.**GetViewPath()**

Owning Class

CaWCheckin

Arguments

None

Return Value

A string that represents the view path.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
strViewPath = objCheckin.GetViewPath()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetViewPath

GetDeleteAfterCI Method

Gets the Delete After Checkin attribute for this process.

Syntax

object.**GetDeleteAfterCI()**

Owning Class

CaWCheckin

Arguments

None

Return Value

A boolean that represents the value.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
blnViewPath = objCheckin.GetDeleteAfterCI()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetDeleteAfterCI

GetFileList Method

Gets a copy of the file list to be checked in by this process. If a remote machine name is set then these files must exist on the remote machine.

Syntax

object.**GetFileList()**

Owning Class

CaWCheckin

Arguments

None

Return Value

A CaWStringList object that represents a copy of the file list for checkin.

Notes

This list is a copy. Any modifications will only take effect if another SetFileList function is called. If a remote machine is set then a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Example

```
Dim objCheckin
Dim objFileList
Set objCheckin = objContext.GetCheckin()
Set objFileList = objCheckin.GetFileList()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetFileList
- SetClientDir
- RemoteMachineName
- SetViewPath
- SetPathOption
- RemoteUserName
- RemotePassword
- CaStringList

Execute Method

Executes the Checkin process on the Version list currently selected using the Version Chooser.

Syntax

object.**Execute**(*bInFileList*)

Owning Class

CaWCheckin

Arguments

bInFileList - If True checks in a file list, if False gets the list from the version chooser.

Return Value

A long - Zero if success, Nonzero if failure.

Notes

None

Example

```
Dim objCheckin
Dim objFileList
' Set check in parameters
Set objCheckin = objContext.GetCheckin()
objCheckin.SetFileList(objFileList)
objCheckin.SetCheckinMode(0)
objCheckin.SetPathOption(0)
objCheckin.SetItemOption(0)
objCheckin.SetDeleteAfterCI(False)
objCheckin.SetViewPath("\CHSDKRep")
objCheckin.SetClientDir("C:\temp")
' Select files to be checked in
Set objStringList = CreateObject("Chsdk.CaWStringList")
objStringList.InitHObject()
objStringList.Add("File1.vbs")
objStringList.Add("File2.vbs")
' Check in
lngRet = ObjCheckin.Execute(True)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWVersionChooser
- CaWStringList

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWCheckin

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
If objCheckin.IsValid() <> 0 Then
    msgbox "This is a valid object", vbOKonly, "Test Checkin"
End If
```

CaWCheckout

CaWCheckout encapsulates the HSDK Checkout process. It cannot be instantiated directly. The Checkout process can be set in the CaWContext object either by name or object ID. The CaWCheckout object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the Project, State, Package, and Checkout Context.
- Select a set of Versions to Checkout using the CaWVersionChooser.
- Invoke the Execute function.

Properties

- RemoteMachineName
- AgentPortNumber
- RemoteUserName
- RemotePassword

Methods

- SetCheckoutMode
- SetPathOption
- SetReplaceFile
- SetClientDir
- SetViewPath
- SetShareWorkDir

- SetUseCITimeStamp
- SetMVSFileTypes
- GetReplaceFile
- GetShareWorkDir
- GetCheckoutMode
- GetPathOption
- GetClientDir
- GetViewPath
- GetUseCITimeStamp
- GetMVSFileTypes
- Execute
- IsValid

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWVersionChooser
- CaStringList

RemoteMachineName Property

Gets or sets the host name of the remote machine where the files are located.

Syntax

object.RemoteMachineName = [value]

Owning Class

CaWCheckout

Arguments

value - A string that represents the machine name.

Notes

If a remote machine is set then a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

A remote CA Harvest SCM File Agent must be running on the specified machine.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
objCheckout.RemoteMachineName = "MyBuildMachine"
strRemoteMachineName = objCheckout.RemoteMachineName
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- SetViewPath
- SetPathOption
- RemoteUserName
- RemotePassword

AgentPortNumber Property

Gets or sets the port number of the remote machine where the files are located.

Syntax

object.AgentPortNumber = [*value*]

Owning Class

CaWCheckin

Arguments

value - A string that represents the machine name.

Notes

If a remote machine is set, then an agent port number, and a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

A remote CA Harvest SCM File Agent must be running on the specified machine.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin
objCheckin.AgentPortNumber = "7001"
strAgentPortNumber = objCheckin.AgentPortNumber
```

More information:

[CHSDK Programming Fundamentals](#) (see page 541)

[CaWContext](#) (see page 595)

[SetClientDir Method](#) (see page 556)

[SetViewPath Method](#) (see page 557)

[SetPathOption MethodCaCheckin](#) (see page 554)

[RemoteMachineName Property](#) (see page 548)

[RemoteUserName Property](#) (see page 551)

[RemotePassword Property](#) (see page 552)

RemoteUserName Property

Gets or sets the user name for the login to the remote machine where the files are located.

Syntax

object.RemoteUserName = [value]

Owning Class

CaWCheckout

Arguments

value - A string that represents the user name.

Notes

If a remote machine is set then a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
objCheckout.RemoteUserName = "George"
strRemoteUserName = objCheckout.RemoteUserName
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- RemoteMachineName
- SetViewPath
- SetPathOption
- RemotePassword

RemotePassword Property

Gets or sets the password for the username used to login to the remote machine where the files are located.

Syntax

object.RemotePassword = [value]

Owning Class

CaWCheckout

Arguments

value - A string that represents the password.

Notes

If a remote machine is set then a remote username and password must also be supplied. The Client Directory must exist on the remote machine.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
objCheckout.RemoteUserName = "George"
strRemoteUserName = objCheckout.RemoteUserName
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- RemoteMachineName
- SetViewPath
- SetPathOption
- RemoteUserName

SetCheckoutMode Method

Sets the Checkout Mode for this process.

Syntax

object.**SetCheckoutMode**(*lngCheckoutMode*)

Owning Class

CaWCheckout

Arguments

lngCheckoutMode - A long that represents the Checkout Mode.

Return Value

A boolean. True if success, False if fail.

Notes

The following are the allowable values for the Checkout Mode.

1

Checkout the versions for read only. No lock is set.

2

Checkout the versions for reserve.

3

Reserve the versions but do not write them to the file system. Set a lock.

4

Checkout selected versions for read only but only if they are newer than the ones previously checked out to the file system. No lock is set.

5

Checkout the versions for concurrent update and still allow other reservations with other packages.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
BlnRet = objCheckout.SetCheckoutMode(CO_MODE_BROWSE)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetCheckoutMode

SetPathOption Method

Sets the Path Option for this process. Files are checked out to the directory structure in locations related to the view path, client directory, and the path option.

Syntax

object.SetPathOption(*lngPathOption*)

Owning Class

CaWCheckout

Arguments

lngPathOption - A long that represents the Path Option.

Return Value

A boolean. True if success, False if fail

Notes

The following are the allowable values for the Path Option.

1

Check out each new version to its existing client path location that mirrors the location in the view path.

2

Check out each new version to its client path location that mirrors the location in the view path and create the client path if it does not yet exist.

3

Check out each new version to the single client path location specified.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
BlnRet = objCheckout.SetPathOption(CO_OPTION_PRESERVE)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetPathOption
- SetClientDir
- SetViewPath

SetReplaceFile Method

Sets Replace File attribute for this process.

Syntax

object.**SetReplaceFile**(*blnFlag*)

Owning Class

CaWCheckout

Arguments

blnFlag - A boolean, if True then read only files are replaced, if False they are not.

Return Value

A boolean. True if success, False if fail

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
BlnRet = objCheckout.SetReplaceFile(True)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetReplaceFile

SetClientDir Method

Sets the Client Directory anchor for this process.

Syntax

object.**SetClientDir**(*strCDir*)

Owning Class

CaWCheckout

Arguments

strCDir - A string that represents a valid directory.

Return Value

A boolean. True if success, False if fail

Notes

If the Checkout is from a remote machine then this directory is from the remote machine.

Example

```
Dim objCheckout  
Set objCheckout = objContext.GetCheckout()  
BlnRet = objCheckout.SetClientDir("c:\temp\work")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetClientDir
- RemoteMachineName
- SetViewPath
- SetPathOption

SetViewPath Method

Sets the repository View Path anchor for this process.

Syntax

object.**SetViewPath**(*strViewPath*)

Owning Class

CaWCheckout

Arguments

strViewPath - A string that represents a valid repository view path.

Return Value

A boolean. True if success, False if fail.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
blnRet = objCheckout.SetViewPath("\rep\work")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- RemoteMachineName
- GetViewPath
- SetPathOption

SetShareWorkDir Method

Sets the Delete After Checkout attribute of this process.

Syntax

object.**SetShareWorkDir**(*blnFlag*)

Owning Class

CaWCheckout

Arguments

blnFlag - A boolean. True if the client directory is shared, False if not.

Return Value

A boolean. True if success, False if fail

Notes

This option is for UNIX file systems. The file access is set such that group has access.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
blnRet = objCheckout.SetShareWorkDir(False)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetShareWorkDir

SetUseCITimeStamp Method

Sets the Use Checkin Timestamp attribute to be used by this Checkout.

Syntax

object.**SetUseCITimeStamp**(*blnFlag*)

Owning Class

CaWCheckout

Arguments

blnFlag - A boolean, if True sets time on file to time version was checked in, if False sets it to current time.

Return Value

A boolean. True if success, False if fail

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
blnRet = objCheckout.SetUseCITimeStamp(False)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetUseCITimeStamp

SetMVSFileTypes Method

Sets the MVS File Types attribute to be used by this process. This is only used if the remote file system is on the MVS platform.

Syntax

object.**SetMVSFileTypes**(*IngOption*)

Owning Class

CaWCheckout

Arguments

IngOption - A long that represents the MVS File Types Option.

Return Value

A boolean. True if success, False if fail.

Notes

The following are the allowable values for the Checkout Mode. They are defined in an include file built into the project.

1

The target file location is a PDS.

2

The target file location is a Sequential file.

Note: The remote machine name must be set to an MVS platform for this option to be used. The Client Directory must exist on the remote machine.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
blnRet = objCheckout.SetMVSFileTypes(HAR_MVS_PDS)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetMVSFileTypes
- SetClientDir
- RemoteMachineName
- SetViewPath
- SetPathOption
- RemoteUserName
- RemotePassword

GetCheckoutMode Method

Gets the Checkout Mode attribute for this process.

Syntax

```
object.GetCheckoutMode()
```

Owning Class

CaWCheckout

Arguments

None

Return Value

A long - The value.

Example

```
Dim objCheckout  
Set objCheckout = objContext.GetCheckout()  
lngCheckoutMode = objCheckout.GetCheckoutMode()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetCheckoutMode

GetPathOption Method

Gets the Path Option attribute for this process.

Syntax

object.GetPathOption()

Owning Class

CaWCheckout

Arguments

None

Return Value

A long - The value.

Example

```
Dim objCheckout  
Set objCheckout = objContext.GetCheckout()  
lngPathOption = objCheckout.GetPathOption()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetPathOption

GetClientDir Method

Gets the Client Directory anchor for this process.

Syntax

object.**GetClientDir**()

Owning Class

CaWCheckout

Arguments

None

Return Value

A string - The value.

Notes

If a remote machine name is set then this is a directory on the remote machine.

Example

```
Dim objCheckout  
Set objCheckout = objContext.GetCheckout()  
strClientDir = objCheckout.GetClientDir()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetClientDir
- RemoteMachineName

GetViewPath Method

Gets the repository View Path anchor for this process.

Syntax

object.**GetViewPath**()

Owning Class

CaWCheckout

Arguments

None

Return Value

A string - The value.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
strViewPath = objCheckout.GetViewPath()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetViewPath

GetReplaceFile Method

Gets the Replace File attribute for this process.

Syntax

object.**GetReplaceFile()**

Owning Class

CaWCheckout

Arguments

None

Return Value

A boolean - The value.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
bInReplaceFile = objCheckout.GetReplaceFile()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetReplaceFile

GetShareWorkDir Method

Gets the Share Working Directory attribute for this process.

Syntax

object.GetShareWorkDir()

Owning Class

CaWCheckout

Arguments

None

Return Value

A boolean - The value.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
bLnShareWorkDir = objCheckout.GetShareWorkDir()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetShareWorkDir

GetUseCITimeStamp Method

Gets the Use Checkin Timestamp attribute for this process.

Syntax

object.GetUseCITimeStamp()

Owning Class

CaWCheckout

Arguments

None

Return Value

A boolean - The value.

Example

```
Dim objCheckout  
Set objCheckout = objContext.GetCheckout()  
blnUseCITimeStamp = objCheckout.GetUseCITimeStamp()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetUseCITimeStamp

GetMVSFileTypes Method

Gets the MVSFileTypes attribute used by this process.

Syntax

object.**GetMVSFileTypes()**

Owning Class

CaWCheckout

Arguments

None

Return Value

A long - The value.

Notes

This attribute is only used if the remote machine name is set to an MVS platform.

Example

```
Dim objCheckout  
Set objCheckout = objContext.GetCheckout()  
lngMVSFileTypes = objCheckout.GetMVSFileTypes()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetMVSFileTypes
- RemoteMachineName

Execute Method

Executes the Checkout process on the Version list currently selected using the Version Chooser.

Syntax

object.Execute()

Owning Class

CaWCheckout

Arguments

None

Return Value

A long - Zero if success, Nonzero if failure.

Example

```
Dim objCheckout
Dim objVersionChooser
' Set checkout parameters
Set objCheckout = objContext.GetCheckout()
objCheckout.SetCheckoutMode(5)
objCheckout.SetPathOption(1)
objCheckout.SetReplaceFile(True)
objCheckout.SetShareWorkDir(False)
objCheckout.SetUseCITimeStamp(False)
objCheckout.SetClientDir("C:\temp")
objCheckout.SetViewPath("\CHSDKRep")
' Select versions to check out
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.ItemName = "File1.vbs"
objVersionChooser.Recursive = True
objVersionChooser.BranchOption = 1
objVersionChooser.ParentPath = "\CHSDKRep"
objVersionChooser.VersionItemOption = 1
objVersionChooser.VersionOption = 256
objVersionChooser.VersionStatusOption = 1
objVersionChooser.Execute()
' Check out
lngRet = objCheckout.Execute()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaVersionChooser

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWCheckout

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
If objCheckout.IsValid() <> 0 Then
    msgbox "This is a valid object", vbOKOnly, "Test Checkout"
End If
```

CaWConcurrentMerge

CaWConcurrentMerge encapsulates the HSDK ConcurrentMerge process. It cannot be instantiated directly. The ConcurrentMerge process can be set in the CaWContext object either by name or object ID. The CaWConcurrentMerge object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the Project, State, Package, and ConcurrentMerge Context.
- Concurrent Merge operates only on the Package in the current context.
- Set the ConcurrentMerge attributes.
- Invoke the Execute function.

Properties

- MergeRule
- VersionDescription

Methods

- Execute
- IsValid

More information:

- CHSDK Programming Fundamentals
- CaWContext

MergeRule Property

Gets or sets Rule for how to merge the versions of files in the package.

Syntax

object.MergeRule = [value]

Owning Class

CaWConcurrentMerge

Arguments

value - A string that represents the rule value.

Notes

The following are the allowable values for the Merge Rule. They are defined in an include file built into the project.

T

Take the entire trunk version and ignore the version on the branch. This corresponds to the GUI TakeTrunk Version option.

B

Take the entire branch version and ignore the version on the trunk. This corresponds to the GUI Take Branch Version option.

C

Create a merge tag only if changes in branch and trunk conflict. This corresponds to the GUI Merge Aggressively option.

A

Create a merge tag if there is any difference in the trunk and branch version at all. This corresponds to the GUI Merge Conservatively option.

Example

```
Dim objConcurrentMerge
Set objConcurrentMerge = objContext.GetConcurrentMerge()
objConcurrentMerge.MergeRule = "A"
strMergeRule = objConcurrentMerge.MergeRule
```

More information:

- CHSDK Programming Fundamentals
- GetMergeRule

VersionDescription Property

Provides a description to be associated with the merged version.

Syntax

object.VersionDescription = [value]

Owning Class

CaWConcurrentMerge

Arguments

value - A string that represents the description.

Notes

The description defaults to a null string.

Example

```
Dim objConcurrentMerge
Set objConcurrentMerge = objContext.GetConcurrentMerge()
objConcurrentMerge.VersionDescription = "My Merged Version"
strVersionDescription = objConcurrentMerge.VersionDescription
```

More information:

- CHSDK Programming Fundamentals
- GetVersionDescription.

Execute Method

Executes the ConcurrentMerge process on the Package in the current context.

Syntax

object.Execute()

Owning Class

CaWConcurrentMerge

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Example

```
Dim objConcurrentMerge
Set objConcurrentMerge = objContext.GetConcurrentMerge()
lngRet = objConcurrentMerge.Execute()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWConcurrentMerge

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objConcurrentMerge
Set objConcurrentMerge = objContext.GetConcurrentMerge ()
If objConcurrentMerge.IsValid() <> 0 Then
    msgbox "This is a valid object", vbOKonly, "Test ConcurrentMerge"
End If
```

CaWContext

CaWContext is a smart container that manages the current CA Harvest SCM context. If a Project is reset then the State is reset. If the State is reset then the View, Package, and all Processes are reset. State, View, and Process lists are retrieved based on the current context. A Package chooser and a Version chooser allow filtered selection based on the context.

The CaWContext Object has no public constructor. One copy is instantiated and maintained by the CaWHarvest object.

Methods

- GetProjectName
- GetProjectId
- GetStateName
- GetStatId
- GetProcessId
- GetPackageName
- GetPackageId
- GetViewName
- GetSnapshotViewName
- GetApprove
- GetApproveName
- GetCheckin
- GetCheckout
- GetCheckoutName
- GetClientPath
- GetConcurrentMerge
- GetConcurrentMergeName
- GetCreatePackage
- GetCreatePackageName
- GetCrossProjectMerge
- GetCrossProjectMergeName
- GetDeleteVersion
- GetDeleteVersionName
- GetDemote

- GetDemoteName
- GetInteractiveMerge
- GetInteractiveMergeName
- GetPackageChooser
- GetPromote
- GetPromoteName
- GetRemoveItem
- GetRemoveItemName
- GetSnapshotViewId
- GetSQL
- GetStateList
- GetTakeSnapshot
- GetTakeSnapshotName
- GetUDP
- GetUDPName
- GetVersionChooser
- GetViewPath
- GetViewPathId
- SetApproveId
- SetApproveName
- SetCheckinId
- SetCheckinName
- SetCheckoutId
- SetCheckoutName
- SetClientPath
- SetConcurrentMergeId
- SetConcurrentMergeName
- SetCreatePackageId
- SetCreatePackageName
- SetCrossProjectMergeById
- SetCrossProjectMergeByName
- SetDeleteVersionId

- SetDeleteVersionName
- SetDemoteld
- SetDemoteName
- SetInteractiveMerge
- SetInteractiveMergeld
- SetPackageld
- SetPackageName
- SetProjectId
- SetProjectName
- SetPromoteld
- SetPromoteName
- SetRemoveltemld
- SetRemoveltemName
- SetSnapshotView
- SetSnapshotViewById
- SetStateld
- SetStateName
- SetTakeSnapshotById
- SetTakeSnapshotByName
- SetUDPById

- SetUDPByName
- SetViewPath
- SetViewPathById

More information:

- CHSDK Programming Fundamentals
- CaWHarvest
- CaWContainer
- CaWPromote
- CaWCheckout
- CaWPackageChooser
- CaWVersionChooser
- CaWDemote
- CaWApprove
- CaWCreatePackage
- CaWDeleteVersion
- CaWRemoveItem
- CaWConcurrentMerge

GetProjectName Method

Retrieve the name of the current CA Harvest SCM Project.

Syntax

object.GetProjectName()

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the project name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strProjectName = objContext.GetProjectName()
```

More information:

- CHSDK Programming Fundamentals
- GetProjectId
- SetProjectName
- SetProjectId

GetProjectId Method

Retrieve the Object ID of the current CA Harvest SCM Project.

Syntax

```
object.GetProjectId()
```

Owning Class

CaWContext

Arguments

None

Return Value

A long that represents the project object ID if set, -1 if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
lngProjectObjId = objContext.GetProjectId()
```

More information:

- CHSDK Programming Fundamentals
- GetProjectName
- SetProjectName
- SetProjectId

GetStateName Method

Retrieve the name of the current CA Harvest SCM State.

Syntax

object.**GetStateName()**

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the state name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strStateName = objContext.GetStateName()
```

More information:

- CHSDK Programming Fundamentals
- GetStatId
- SetStateName
- SetStatId

GetStateId Method

Retrieve the Object ID of the current CA Harvest SCM State.

Syntax

object.**GetStatId()**

Owning Class

CaWContext

Arguments

None

Return Value

A long that represents the State Object Id if set, -1 if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
lngStateObjId = objContext.GetStateId()
```

More information:

- CHSDK Programming Fundamentals
- GetStateName
- SetStateName
- SetStateld

GetProcessId Method

Get the process ID of a process by its process type.

Syntax

lng = *object*.**GetProcessId**(*lngProcessType*)

Arguments

lngProcessType-A long representing the process type.

Return Value

A long representing the process ID.

Valid Process Types

- HAR_APPROVE_PROCESS_TYPE = 200
- HAR_CHECKIN_PROCESS_TYPE = 201
- HAR_CHECKOUT_PROCESS_TYPE = 202
- HAR_DELETEVERSIONS_PROCESS_TYPE = 203
- HAR_DEMOTE_PROCESS_TYPE = 204
- HAR_LISTDIFFERENCE_PROCESS_TYPE = 205
- HAR_LISTVERSION_PROCESS_TYPE = 206
- HAR_CONCURRENT_MERGE_PROCESS_TYPE = 207

- HAR_CROSSPROJECT_MERGE_PROCESS_TYPE = 208
- HAR_INTERACTIVE_MERGE_PROCESS_TYPE = 209
- HAR_NOTIFY_PROCESS_TYPE = 210
- HAR_CREATE_PACKAGE_PROCESS_TYPE = 211
- HAR_MOVE_PACKAGE_PROCESS_TYPE = 212
- HAR_PROMOTE_PROCESS_TYPE = 213
- HAR_REMOVEITEM_PROCESS_TYPE = 214
- HAR_TAKESNAPSHOT_PROCESS_TYPE = 215
- HAR_UDP_PROCESS_TYPE = 216
- HAR_RENAMEITEM_PROCESS_TYPE = 217
- HAR_SWITCH_PACKAGE_PROCESS_TYPE = 218

Example

```
blnRet = objContext.SetCheckoutName("Checkout")
if blnRet = true
    objId = objContext.GetProcessId(202)
end if
```

More information:

- CHSDK Programming Fundamentals
- SetCheckoutName
- GetCheckoutName
- GetCheckoutId
- SetCheckoutId

GetPackageName Method

Retrieve the name of the current CA Harvest SCM Package.

Syntax

```
object.GetPackageName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the Package name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strPackageName = objContext.GetPackageName()
```

More information:

- CHSDK Programming Fundamentals
- GetPackageId
- SetPackageName
- SetPackageId

GetPackageId Method

Retrieve the Object Id of the current CA Harvest SCM Package.

Syntax

object.**GetPackageId()**

Owning Class

CaWContext

Arguments

None

Return Value

A long that represents the Package Object Id if set, -1 if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
lngPackageObjId = objContext.GetPackageId()
```

More information:

- CHSDK Programming Fundamentals
- GetPackageName
- SetPackageName
- SetPackageId

GetViewName Method

Retrieve the name of the current CA Harvest SCM View.

Syntax

object.**GetViewName()**

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the View name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strViewName = objContext.GetViewName()
```

More information:

- CHSDK Programming Fundamentals
- SetStateName
- SetStatId

GetSnapshotViewName Method

Retrieve the name of the current CA Harvest SCM SnapshotView.

Syntax

object.GetSnapshotViewName()

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the SnapshotView name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strSnapshotViewName = objContext.GetSnapshotViewName()
```

More information:

- CHSDK Programming Fundamentals
- SetSnapshotView

GetCheckoutName Method

Retrieve the name of the current CA Harvest SCM Checkout.

Syntax

object.GetCheckoutName()

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the Checkout name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strCheckoutName = objContext.GetCheckoutName()
```

More information:

- CHSDK Programming Fundamentals
- GetCheckout
- SetCheckoutName
- SetCheckoutId

GetCheckinName Method

Retrieve the name of the current CA Harvest SCM Checkin.

Syntax

```
object.GetCheckinName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the Checkin name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strCheckinName = objContext.GetCheckinName()
```

More information:

- CHSDK Programming Fundamentals
- GetCheckin
- SetCheckinName
- SetCheckinId

GetPromoteName Method

Retrieve the name of the current CA Harvest SCM Promote.

Syntax

```
object.GetPromoteName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the Promote name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strPromoteName = objContext.GetPromoteName()
```

More information:

- CHSDK Programming Fundamentals
- GetPromote
- SetPromoteName
- SetPromoteId

GetDemoteName Method

Retrieves the name of the current CA Harvest SCM Demote.

Syntax

```
object.GetDemoteName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the Demote name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strDemoteName = objContext.GetDemoteName()
```

More information:

- CHSDK Programming Fundamentals
- GetDemote
- SetDemoteName
- SetDemoteld

GetApproveName Method

Retrieves the name of the current CA Harvest SCM Approve.

Syntax

```
object.GetApproveName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the Approve name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strApproveName = objContext.GetApproveName()
```

More information:

- CHSDK Programming Fundamentals
- GetApprove
- SetApproveName
- SetApproveId

GetCreatePackageName Method

Retrieve the name of the current CA Harvest SCM CreatePackage.

Syntax

```
object.GetCreatePackageName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the CreatePackage name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strCreatePackageName = objContext.GetCreatePackageName()
```

More information:

- CHSDK Programming Fundamentals
- GetCreatePackage
- SetCreatePackageName
- SetCreatePackageId

GetDeleteVersionName Method

Retrieve the name of the current CA Harvest SCM DeleteVersion.

Syntax

```
object.GetDeleteVersionName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the DeleteVersion name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strDeleteVersionName = objContext.GetDeleteVersionName()
```

More information:

- CHSDK Programming Fundamentals
- GetDeleteVersion
- SetDeleteVersionName
- SetDeleteVersionId

GetRemoveItemName Method

Retrieves the name of the current CA Harvest SCM RemoveItem.

Syntax

```
object.GetRemoveItemName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the RemoveItem name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strRemoveItemName = objContext.GetRemoveItemName()
```

More information:

- CHSDK Programming Fundamentals
- GetRemoveItem
- SetRemoveItemName
- SetRemoveItemId

GetConcurrentMergeName Method

Retrieves the name of the current CA Harvest SCM ConcurrentMerge.

Syntax

```
object.GetConcurrentMergeName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the ConcurrentMerge name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strConcurrentMergeName = objContext.GetConcurrentMergeName()
```

More information:

- CHSDK Programming Fundamentals
- GetConcurrentMerge
- SetConcurrentMergeName
- SetConcurrentMergeld

GetClientPath Method

Retrieves the current CA Harvest SCM client path.

Syntax

object.GetClientPath()

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the Client path if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strClientPath = objContext.GetClientPath()
```

More information:

- CHSDK Programming Fundamentals
- SetClientPath

GetViewPath Method

Retrieves the current CA Harvest SCM View path.

Syntax

object.**GetViewPath**()

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the View path if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strViewPath = objContext.GetViewPath()
```

More information:

- CHSDK Programming Fundamentals
- SetViewPath

GetViewPathId Method

Retrieve the Object ID of the current context viewpath.

Syntax

object.**GetViewPathId**()

Owning Class

CaWContext

Arguments

None

Return Value

A long represents the Package Object ID if set, -1 if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
IngViewPathId = objContext.GetViewPathId()
```

More information:

- CHSDK Programming Fundamentals
- GetViewPath
- SetViewPath
- SetViewPathbyId

GetCheckout Method

Retrieve the current CA Harvest SCM Checkout object.

Syntax

```
object.GetCheckout()
```

Owning Class

CaWContext

Arguments

None

Return Value

A CaWCheckout object that represents the current Checkout object if set, null if not set or error.

Example

```
Dim objCheckout
Set objCheckout = objContext.GetCheckout()
If objCheckout Is Nothing Then
    MsgBox "Can't get checkout object.", vbOKOnly + vbExclamation, "GetCheckout"
End If
```

More information:

- CHSDK Programming Fundamentals
- GetCheckoutName
- SetCheckoutName
- SetCheckoutId

GetCheckin Method

Retrieve the current CA Harvest SCM Checkin object.

Syntax

object.**GetCheckin()**

Owning Class

CaWContext

Arguments

None

Return Value

A CaWCheckin object that represents the current Checkin object if set, null if not set or error.

Example

```
Dim objCheckin
Set objCheckin = objContext.GetCheckin()
If objCheckin Is Nothing Then
    MsgBox "Can't get checkin object.", vbOKOnly + vbExclamation, "GetCheckin"
End If
```

More information:

- CHSDK Programming Fundamentals
- GetCheckinName
- SetCheckinName
- SetCheckinId

GetPromote Method

Retrieve the current CA Harvest SCM Promote object.

Syntax

object.**GetPromote()**

Owning Class

CaWContext

Arguments

None

Return Value

A CaWPromote object that represents the current Promote object if set, null if not set or error.

Example

```
Dim objPromote
Set objPromote = objContext.GetPromote()
If objPromote Is Nothing Then
    MsgBox "Can't get promote object.", vbOKOnly + vbExclamation, "GetPromote"
End If
```

More information:

- CHSDK Programming Fundamentals
- GetPromoteName
- SetPromoteName
- SetPromoteId

GetDemote Method

Retrieve the current CA Harvest SCM Demote object.

Syntax

object.**GetDemote()**

Owning Class

CaWContext

Arguments

None

Return Value

A CaWDemote object that represents the current Demote object if set, null if not set or error.

Example

```
Dim objDemote
Set objDemote = objContext.GetDemote()
If objDemote Is Nothing Then
    MsgBox "Can't get demote object.", vbOKOnly + vbExclamation, "GetDemote"
End If
```

More information:

- CHSDK Programming Fundamentals
- GetDemoteName
- SetDemoteName
- SetDemoteld

GetApprove Method

Retrieve the current CA Harvest SCM Approve object.

Syntax

object.GetApprove()

Owning Class

CaWContext

Arguments

None

Return Value

A CaWApprove object that represents the current Approve object if set, null if not set or error.

Example

```
Dim objApprove
Set objApprove = objContext.GetApprove()
If objApprove Is Nothing Then
    MsgBox "Can't get approve object.", vbOKOnly + vbExclamation, "GetApprove"
End If
```

More information:

- CHSDK Programming Fundamentals
- GetApproveName
- SetApproveName
- SetApproveId

GetCreatePackage Method

Retrieve the current CA Harvest SCM CreatePackage object.

Syntax

object.**GetCreatePackage()**

Owning Class

CaWContext

Arguments

None

Return Value

A CaWCreatePackage object that represents the current CreatePackage object if set, null if not set or error.

Example

```
Dim objCreatePackage
Set objCreatePackage = objContext.CreatePackage()
If objCreatePackage Is Nothing Then
    MsgBox "Can't get create package object.", vbOKOnly + vbExclamation,
    "GetCreatePackage"
End If
```

■ More information:

- CHSDK Programming Fundamentals
- GetCreatePackageName
- SetCreatePackageName
- SetCreatePackageId

GetDeleteVersion Method

Retrieve the current CA Harvest SCM DeleteVersion object.

Syntax

```
object.GetDeleteVersion()
```

Owning Class

CaWContext

Arguments

None

Return Value

A CaWDeleteVersion object that represents the current DeleteVersion object if set, null if not set or error.

Example

```
Dim objDeleteVersion
Set objDeleteVersion = objContext.GetDeleteVersion()
If objDeleteVersion Is Nothing Then
    MsgBox "Can't get delete version object.", vbOKOnly + vbExclamation,
        "GetDeleteVersion"
End If
```

■ More information:

- CHSDK Programming Fundamentals
- GetDeleteVersionName
- SetDeleteVersionName
- SetDeleteVersionId

GetInteractiveMerge Method

Retrieve the current CA Harvest SCM InteractiveMerge object.

Syntax

```
object.GetInteractiveMerge()
```

Owning Class

CaWContext

Arguments

None

Return Value

A CaWInteractiveMerge object that represents the current InteractiveMerge object if set, null if not set or error.

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()

If objInteractiveMerge Is Nothing Then
    MsgBox "Can't get Interactive merge object.", vbOKOnly + vbExclamation,
    "GetInteractiveMerge"
End If
```

More information:

- CHSDK Programming Fundamentals
- GetInteractiveMergeName
- SetInteractiveMergeName
- SetInteractiveMergeId

GetInteractiveMergeName Method

Retrieves the name of the current CA Harvest SCM InteractiveMerge.

Syntax

```
object.GetInteractiveMergeName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the InteractiveMerge name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strInteractiveMerge = objContext.GetInteractiveMergeName()
```

More information:

- CHSDK Programming Fundamentals
- GetInteractiveMerge
- SetInteractiveMergeName
- SetInteractiveMergeld

GetRemoveItem Method

Retrieves the current CA Harvest SCM RemoveItem object.

Syntax

object.**GetRemoveItem()**

Owning Class

CaWContext

Arguments

None

Return Value

A CaWRemoveItem object that represents the current RemoveItem object if set, null if not set or error.

Example

```
Dim objRemoveItem
Set objRemoveItem = objContext.GetRemoveItem()
If objRemoveItem Is Nothing Then
    msgbox "Can't get remove item object.", vbOKonly + vbExclamation, "GetRemoveItem"
End If
```

More information:

- CHSDK Programming Fundamentals
- GetRemoveItemName
- SetRemoveItemName
- SetRemoveItemId

GetConcurrentMerge Method

Retrieves the current CA Harvest SCM ConcurrentMerge object.

Syntax

object.**GetConcurrentMerge()**

Owning Class

CaWContext

Arguments

None

Return Value

A CaWConcurrentMerge object that represents the current ConcurrentMerge object if set, null if not set or error.

Example

```
Dim objConcurrentMerge
Set objConcurrentMerge = objContext.GetConcurrentMerge()
If objConcurrentMerge Is Nothing Then
    MsgBox "Can't get concurrent merge object.", vbOKOnly + vbExclamation,
        "GetConcurrentMerge"
End If
```

More information:

- CHSDK Programming Fundamentals
- GetConcurrentMergeName
- SetConcurrentMergeName
- SetConcurrentMergeld

GetPackageChooser Method

Retrieves the CA Harvest SCM PackageChooser object.

Syntax

object.GetPackageChooser()

Owning Class

CaWContext

Arguments

None

Return Value

A CaWPackageChooser object that represents the PackageChooser object, null if error.

Example

```
Dim objPackageChooser
Set objPackageChooser = objContext.GetPackageChooser()
If objPackageChooser Is Nothing Then

msgbox "Can't get package chooser object.", vbOKonly + vbExclamation,
"GetPackageChooser"

End If
```

More information:

- CHSDK Programming Fundamentals

GetVersionChooser Method

Retrieves the CA Harvest SCM VersionChooser object.

Syntax

object.**GetVersionChooser()**

Owning Class

CaWContext

Arguments

None

Return Value

A CaWVersionChooser object that represents the VersionChooser object, null if error.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
If objVersionChooser Is Nothing Then

msgbox "Can't get version chooser object.", vbOKonly + vbExclamation,
"GetVersionChooser"

End If
```

More information:

- CHSDK Programming Fundamentals

GetStateList Method

Retrieve a CaWContainer object that contains the attributes for each CA Harvest SCM State in the list.

Syntax

object.**GetStateList()**

Owning Class

CaWContext

Arguments

None

Return Value

Returns a CaWContainer object, null if error.

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME

- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME
- CA_ATTRKEY_PROJECTID
- CA_ATTRKEY_SNAPSHOT
- CA_ATTRKEY_VIEW_NAME
- CA_ATTRKEY_VIEWID

Example

```
Dim objContainer
Set objContainer = objContext.GetStateList()
If objContainer Is Nothing Then
    MsgBox "Can't get state list.", vbOKOnly + vbExclamation, "GetStateList"
End If
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

GetViewList Method

Retrieves a CaContainer object that contains the attributes for each CA Harvest SCM View in the list.

Syntax

```
object.GetViewList()
```

Owning Class

CaWContext

Arguments

None

Return Value

Returns a CaWContainer object, null if error.

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_PROJECTID
- CA_ATTRKEY_VIEW_TYPE

Example

```
Dim objContainer
Set objContainer = objContext.GetViewList()
If objContainer Is Nothing Then

    msgbox "Can't get view list.", vbOKonly + vbExclamation, "GetViewList"

End If
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

GetProcessList Method

Retrieves a CaContainer object that contains the attributes for each CA Harvest SCM Process in the list.

Syntax

object.**GetProcessList()**

Owning Class

CaWContext

Arguments

None

Return Value

Returns a CaWContainer object, null if error.

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_PROJECTID
- CA_ATTRKEY_STATEID
- CA_ATTRKEY_PROCTYPE

Example

```
Dim objContainer
Set objContainer = objContext.GetProcessList()
If objContainer Is Nothing Then
    msgbox "Can't get process list.", vbOKonly + vbExclamation, "GetProcessList"
End If
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

SetProjectName Method

Sets the specified Project as the current context Project.

Syntax

```
object.SetProjectName(strProjectName)
```

Owning Class

CaWContext

Arguments

strProjectName - A string that represents the Project name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetProjectName("My Project")
```

More information:

- CHSDK Programming Fundamentals
- SetProjectId
- GetProjectName
- GetProject
- CaWHarvest

SetProjectId Method

Sets the specified Project as the current context Project.

Syntax

object.**SetProjectId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the Project object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetProjectId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetProjectName
- GetProjectName
- GetProject
- CaWHarvest

SetStateName Method

Sets the specified State as the current context State.

Syntax

object.**SetStateName**(*strStateName*)

Owning Class

CaWContext

Arguments

strStateName - A string that represents the State name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetStateName("My State")
```

More information:

- CHSDK Programming Fundamentals
- SetStateId
- GetStateName
- GetStateId
- GetStateList

SetStateId Method

Sets the specified State as the current context State.

Syntax

object.**SetStateId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the State object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetStateId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetStateName
- GetStateName
- GetStateId
- GetStateList

SetCheckoutName Method

Sets the specified Checkout as the current context Checkout.

Syntax

object.**SetCheckoutName**(*strCheckoutName*)

Owning Class

CaWContext

Arguments

strCheckoutName - A string that represents the Checkout name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetCheckoutName("Checkout")
```

More information:

- CHSDK Programming Fundamentals
- SetCheckoutId
- GetCheckoutName
- GetCheckout

SetCheckoutId Method

Sets the specified Checkout as the current context Checkout.

Syntax

object.**SetCheckoutId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the Checkout object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetCheckoutId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetCheckoutName
- GetCheckoutName
- GetCheckout

SetCheckinName Method

Sets the specified Checkin as the current context Checkin.

Syntax

object.**SetCheckinName**(*strCheckinName*)

Owning Class

CaWContext

Arguments

strCheckinName - A string that represents the Checkin name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetCheckinName("Checkin")
```

More information:

- CHSDK Programming Fundamentals
- SetCheckinId
- GetCheckinName
- GetCheckin

SetCheckinId Method

Sets the specified Checkin as the current context Checkin.

Syntax

object.**SetCheckinId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the Checkin object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetCheckinId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetCheckinName
- GetCheckinName
- GetCheckin

SetPromoteName Method

Sets the specified Promote as the current context Promote.

Syntax

object.**SetPromoteName**(*strPromoteName*)

Owning Class

CaWContext

Arguments

strPromoteName - A string that represents the Promote name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetPromoteName("Promote")
```

More information:

- CHSDK Programming Fundamentals
- SetPromoteId
- GetPromoteName
- GetPromote

SetPromoteId Method

Sets the specified Promote as the current context Promote.

Syntax

object.**SetPromoteId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the Promote object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetPromoteId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetPromoteName
- GetPromoteName
- GetPromote

SetDemoteName Method

Sets the specified Demote as the current context Demote.

Syntax

object.**SetDemoteName**(*strDemoteName*)

Owning Class

CaWContext

Arguments

strDemoteName - A string that represents the Demote name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetDemoteName("Demote")
```

More information:

- CHSDK Programming Fundamentals
- SetDemoteld
- GetDemoteName
- GetDemote

SetDemoteId Method

Sets the specified Demote as the current context Demote.

Syntax

object.**SetDemoteId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the Demote object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetDemoteId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetDemoteName
- GetDemoteName
- GetDemote

SetApproveName Method

Sets the specified Approve as the current context Approve.

Syntax

object.**SetApproveName**(*strApproveName*)

Owning Class

CaWContext

Arguments

strApproveName - A string that represents the Approve name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetApproveName("Approve")
```

More information:

- CHSDK Programming Fundamentals
- SetApproveId
- GetApproveName
- GetApprove

SetApproveId Method

Sets the specified Approve as the current context Approve.

Syntax

object.**SetApproveId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the Approve object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetApproveId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetApproveName
- GetApproveName
- GetApprove

SetCreatePackageName Method

Sets the specified CreatePackage as the current context CreatePackage.

Syntax

object.**SetCreatePackageName**(*strCreatePackageName*)

Owning Class

CaWContext

Arguments

strCreatePackageName - A string that represents the CreatePackage name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetCreatePackageName("Create Package")
```

More information:

- CHSDK Programming Fundamentals
- SetCreatePackageId
- GetCreatePackageName
- GetCreatePackage

SetCreatePackageId Method

Sets the specified CreatePackage as the current context CreatePackage.

Syntax

object.**SetCreatePackageId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the CreatePackage object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetCreatePackageId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetCreatePackageName
- GetCreatePackageName
- GetCreatePackage

SetDeleteVersionName Method

Sets the specified DeleteVersion as the current context DeleteVersion.

Syntax

object.**SetDeleteVersionName**(*strDeleteVersionName*)

Owning Class

CaWContext

Arguments

strDeleteVersionName - A string that represents the DeleteVersion name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetDeleteVersionName("Delete Version")
```

More information:

- CHSDK Programming Fundamentals
- SetDeleteVersionId
- GetDeleteVersionName
- GetDeleteVersion

SetDeleteVersionId Method

Sets the specified DeleteVersion as the current context DeleteVersion.

Syntax

object.**SetDeleteVersionId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the DeleteVersion object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetDeleteVersionId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetDeleteVersionName
- GetDeleteVersionName
- GetDeleteVersion

SetInteractiveMergeName Method

Sets the specified InteractiveMerge as the current context InteractiveMerge.

Syntax

object.**SetInteractiveMergeName**(*strInteractiveMergeName*)

Owning Class

CaWContext

Arguments

strInteractiveMergeName-A string that represents the InteractiveMerge name.

Return Value

A boolean-True if success, False if fails.

Example

```
blnRet = objContext.SetInteractiveMergeName("Interactive Merge")
```

More information:

- CHSDK Programming Fundamentals
- GetInteractiveMerge
- GetInteractiveMergeName
- SetInteractiveMergeId

SetInteractiveMergeId Method

Sets the specified InteractiveMerge as the current context InteractiveMerge.

Syntax

object.**SetInteractiveMergeName**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId-A long that represents the InteractiveMerge object ID.

Return Value

A boolean-True if success, False if fails.

Example

```
blnRet = objContext.SetInteractiveMergeId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- GetInteractiveMerge
- GetInteractiveMergeName
- SetInteractiveMergeName

SetRemoveItemName Method

Sets the specified RemoveItem as the current context RemoveItem.

Syntax

object.**SetRemoveItemName**(*strRemoveItemName*)

Owning Class

CaWContext

Arguments

strRemoveItemName - A string that represents the RemoveItem name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetRemoveItemName("Remove Item")
```

More information:

- CHSDK Programming Fundamentals
- SetRemoveItemId
- GetRemoveItemName
- GetRemoveItem

SetRemoveItemId Method

Sets the specified RemoveItem as the current context RemoveItem.

Syntax

object.**RemoveItemId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the RemoveItem object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetRemoveItemId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetRemoveItemName
- GetRemoveItemName
- GetRemoveItem

SetConcurrentMergeName Method

Sets the specified ConcurrentMerge as the current context ConcurrentMerge.

Syntax

object.**SetConcurrentMergeName**(*strConcurrentMergeName*)

Owning Class

CaWContext

Arguments

strConcurrentMergeName - A string that represents the ConcurrentMerge name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetConcurrentMergeName("Concurrent Merge")
```

More information:

- CHSDK Programming Fundamentals
- SetConcurrentMergeId
- GetConcurrentMergeName
- GetConcurrentMerge

SetConcurrentMergeId Method

Sets the specified ConcurrentMerge as the current context ConcurrentMerge.

Syntax

object.**SetConcurrentMergeId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the ConcurrentMerge object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetConcurrentMergeId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetConcurrentMergeName
- GetConcurrentMergeName
- GetConcurrentMerge

SetPackageName Method

Sets the specified Package as the current context Package.

Syntax

object.**SetPackageName**(*strPackageName*)

Owning Class

CaWContext

Arguments

strPackageName - A string that represents the Package name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetPackageName("My Package")
```

More information:

- CHSDK Programming Fundamentals
- SetPackageId
- GetPackageName
- GetPackage

SetPackageId Method

Sets the specified Package as the current context Package.

Syntax

object.**SetPackageId**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long that represents the Package object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetPackageId(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- SetPackageName
- GetPackageName
- GetPackage

SetSnapshotView Method

Sets the specified SnapshotView as the current context SnapshotView.

Syntax

object.**SetSnapshotView**(*strSnapshotViewName*)

Owning Class

CaWContext

Arguments

strSnapshotViewName - A string that represents the SnapshotView name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetSnapshotView("Snapshot View")
```

More information:

- CHSDK Programming Fundamentals
- GetSnapshotViewName

SetClientPath Method

Sets the specified Client Path as the current context Client Path.

Syntax

object.**SetClientPath**(*strClientPath*)

Owning Class

CaWContext

Arguments

strClientPath - A string that represents the Client Path.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetClientPath("C:\temp")
```

More information:

- CHSDK Programming Fundamentals
- GetClientPath

SetViewPath Method

Sets the specified View Path as the current context View Path.

Syntax

object.**SetViewPath**(*strViewPath*)

Owning Class

CaWContext

Arguments

strViewPath - A string that represents the View Path.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetViewPath("\\CHSDKRep")
```

More information:

- CHSDK Programming Fundamentals
- GetViewPath

SetViewPathById Method

Sets the specified viewpath as the current context viewpath.

Syntax

object.**SetViewPathById**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId-A long representing the viewpath object ID.

Return Value

A Boolean-True if success, False if fails.

Example

```
BlnRet = obj.SetViewPathById(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- GetViewPathId
- GetViewPath
- SetViewPath

GetCrossProjectMerge Method

Retrieves the current CA Harvest SCM CrossProjectMerge object.

Syntax

object.**GetCrossProjectMerge**()

Owning Class

CaWContext

Arguments

None

Return Value

A CaWCrossProjectMerge object if set, otherwise a null object.

Example

```
Dim objCrossProjectMerge
Set objCrossProjectMerge = objContext.GetCrossProjectMerge()
If objCrossProjectMerge Is Nothing Then

    MsgBox "Can't get CrossProjectMerge object.", vbOKOnly + vbExclamation,
        "GetCrossProjectMerge"

End If
```

More information:

- CHSDK Programming Fundamentals
- SetCrossProjectMerge
- GetCrossProjectMergeName

GetCrossProjectMergeName Method

Retrieves the name of the current CA Harvest SCM CrossProjectMerge.

Syntax

```
object.GetCrossProjectMergeName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string that represents the CrossProjectMerge name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strCrossProjectMergeName = objContext.GetCrossProjectMergeName()
```

More information:

- CHSDK Programming Fundamentals
- SetCrossProjectMerge
- GetCrossProjectMerge

GetSnapshotViewId Method

Retrieves the object ID of the current CA Harvest SCM SnapshotView.

Syntax

object.GetSnapshotViewId()

Owning Class

CaWContext

Arguments

None

Return Value

A long that represents the SnapshotView id if set, -1 if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
lngSnapshotViewObjId = objContext.GetSnapshotViewId()
```

More information:

- CHSDK Programming Fundamentals
- SetSnapshotView
- GetSnapshotViewName

GetSQL Method

Retrieves the CA Harvest SCM SQL object.

Syntax

object.GetSQL()

Owning Class

CaWContext

Arguments

None

Return Value

A CaWSQL object or null object if error.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
Set objSQL = objContext.GetSQL()
```

More information:

- CHSDK Programming Fundamentals

GetTakeSnapshot Method

Syntax

object.GetTakeSnapshot()

Owning Class

CaWContext

Arguments

None

Return Value

A CaWTakeSnapshot object if set, otherwise a null object.

Example

```
Dim objTakeSnapshot
Set objTakeSnapshot = objContext.GetTakeSnapshot ()
If objTakeSnapshot Is Nothing Then

    msgbox "Can't get TakeSnapshot object.", vbOKonly + vbExclamation,
    "GetTakeSnapshot"

End If
```

More information:

- SetTakeSnapshot
- GetTakeSnapshotName
- CHSDK Programming Fundamentals

GetTakeSnapshotName Method

Retrieves the name of the current CA Harvest SCM TakeSnapshot.

Syntax

```
object.GetTakeSnapshotName()
```

Owning Class

CaWContext

Arguments

None

Return Value

A string represents the TakeSnapshot name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strTakeSnapshotName = objContext.GetTakeSnapshotName()
```

More information:

- CHSDK Programming Fundamentals
- SetTakeSnapshot
- GetTakeSnapshot

GetUDP Method

Retrieves the current CA Harvest SCM UDP object.

Syntax

object.**GetUDP()**

Owning Class

CaWContext

Arguments

None

Return Value

A CaWUDP object if set or a null object if error.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
Set objUDP = objContext.GetUDP()
```

More information:

- SetUDP
- GetUDPName
- CHSDK Programming Fundamentals

GetUDPName Method

Retrieves the name of the current CA Harvest SCM UDP.

Syntax

object.GetUDPName()

Owning Class

CaWContext

Arguments

None

Return Value

A string represents the UDP name if set, empty string if empty.

Example

```
Dim objContext
Set objContext = sobjHarvest.GetContext()
strUDP = objContext.GetUDPName()
```

More information:

- CHSDK Programming Fundamentals
- SetUDP
- GetUDP

SetCrossProjectMergeById Method

Sets the specified CrossProjectMerge as the current context CrossProjectMerge by object ID.

Syntax

***object*.SetCrossProjectMergeById(*lngObjId*)**

Owning Class

CaWContext

Arguments

lngObjId - A long represents the CrossProjectMerge object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetCrossProjectMergeById(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- GetCrossProjectMergeName
- GetCrossProjectMerge

SetCrossProjectMergeByName Method

Sets the specified CrossProjectMerge as the current context CrossProjectMerge.

Syntax

```
object.SetCrossProjectMergeByName(strCrossProjectMergeName)
```

Owning Class

CaWContext

Arguments

strCrossProjectMergeName - A string represents the CrossProjectMerge name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetCrossProjectMergeByName(strName)
```

More information:

- CHSDK Programming Fundamentals
- GetCrossProjectMergeName
- GetCrossProjectMerge

SetSnapshotViewById Method

Sets the specified SnapshotView as the current context SnapshotView.

Syntax

object.**SetSnapshotViewById**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long represents the SnapshotView object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetSnapshotViewById(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- GetSnapshotViewName
- GetSnapshotViewId

SetTakeSnapshotById Method

Sets the specified TakeSnapshot as the current context TakeSnapshot.

Syntax

object.**SetTakeSnapshotById**(*lngObjId*)

Owning Class

CaWContext

Arguments

lngObjId - A long represents the TakeSnapshot object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetTakeSnapshotById(lngObjId)
```

More information:

- CHSDK Programming Fundamentals
- GetTakeSnapshotName
- GetTakeSnapshot

SetTakeSnapshotByName Method

Sets the specified TakeSnapshot as the current context TakeSnapshot.

Syntax

object.**SetTakeSnapshotByName**(*strTakeSnapshotName*)

Owning Class

CaWContext

Arguments

strTakeSnapshotName - A string represents the TakeSnapshot process name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetTakeSnapshotByName(strName)
```

More information:

- CHSDK Programming Fundamentals
- GetTakeSnapshotName
- GetTakeSnapshot

SetUDPByName Method

Sets the specified UDP as the current context UDP.

Syntax

object.**SetUDPByName**(*strUDPName*)

Owning Class

CaWContext

Arguments

strUDPName - A string represents the UDP process name.

Return Value

A boolean - True if success, False if fails.

Example

```
blnRet = objContext.SetUDPByName(strName)
```

More information:

- CHSDK Programming Fundamentals
- GetUDPName
- GetUDP

SetUDPById Method

Sets the specified UDP as the current context UDP.

Syntax

object.**SetTakeSnapshotById**(*lngObjId*)

Owning Class

CaWContext

Arguments

LngObjId - A long represents the UDP object ID.

Return Value

A boolean - True if success, False if fails.

Example

```
bLnRet = objContext.SetUDPById(LngObjId)
```

More information:

- CHSDK Programming Fundamentals
- GetUDPName
- GetUDP

CaWCreatePackage

CaWCreatePackage encapsulates the HSDK CreateChangePackage process. It cannot be instantiated directly. The CreatePackage process can be set in the CaWContext object either by name or object ID. The CaWCreatePackage object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

A Create Package process has a default name expression to generate the package name if none is supplied. The DefaultName property returns the expression. If no expression then the default is "Package- number" where number is a sequential integer value.

The PackageName property returns the user specified name prior to an Execute call and the resulting package name after an Execute call. The name must be reset if additional Execute calls are to be made.

Usage

- Set the Project, State, and CreatePackage Context.
- Set the CreatePackage attributes.
- Invoke the Execute function.

Properties

- `PackageName`
- `AssignedUserName`
- `Description`
- `Priority`

Methods

- `Execute`
- `IsValid`

More information:

- CHSDK Programming Fundamentals
- `CaWContext`

PackageName Property

Gets or sets a name for the new Package.

Syntax

object.**PackageName** = [value]

Owning Class

`CaWCreatePackage`

Arguments

value - A string that sets the Package name.

Notes

If the package name attribute is set, then it is used without regard to the default name expression.

Example

```
Dim objCreatePackage
Set objCreatePackage = objContext.GetCreatePackage()
objCreatePackage.PackageName = "Package - 12"
strPackageName = objCreatePackage.PackageName
```

More information:

- CHSDK Programming Fundamentals

AssignedUserName Property

Retrieves assigned user or assigns a user to the new Package.

Syntax

object.AssignedUserName = [*value*]

Owning Class

CaWCreatePackage

Arguments

value - A string that sets the username for the assigned user

Example

```
Dim objCreatePackage
Set objCreatePackage = objContext.GetCreatePackage()
objCreatePackage.AssignedUserName = "User-1"
strAssignedUserName = objCreatePackage.AssignedUserName
```

More information:

- CHSDK Programming Fundamentals

Description Property

Gets or sets a description for the new package.

Syntax

object.**Description** = [value]

Owning Class

CaWCreatePackage

Arguments

value - A string that sets the description.

Example

```
Dim objCreatePackage
Set objCreatePackage = objContext.GetCreatePackage()
objCreatePackage.Description = "A good package"
strDescription = objCreatePackage.Description
```

More information:

- CHSDK Programming Fundamentals

Priority Property

Gets or sets the Package priority attribute.

Syntax

object.**Priority** = [value]

Owning Class

CaWCreatePackage

Arguments

value - A long that sets the priority value.

Notes

Priorities are integers but the meaning of each value is user defined.

Example

```
Dim objCreatePackage
Set objCreatePackage = objContext.CreatePackage()
objCreatePackage.Priority = 1
lngPriority = objCreatePackage.Priority
```

More information:

- CHSDK Programming Fundamentals

DefaultName Property (Read Only)

Retrieves the default name expression for the new package.

Syntax

object.DefaultName

Owning Class

CaWCreatePackage

Notes

The CA Harvest SCM administrator sets the default name expression as an attribute of the CA Harvest SCM Create Package process. The default name expression generates the package name if none is supplied with the PackageName property. If there is no default expression, then the default is "Package- number" where number is a sequential integer value.

Example

```
Dim objCreatePackage
Set objCreatePackage = objContext.CreatePackage()
strDefaultName = objCreatePackage.DefaultName
```

More information:

- CHSDK Programming Fundamentals
- PackageName

Execute Method

Executes the CreatePackage process on the Package list currently selected using the Package Chooser.

Syntax

object.Execute()

Owning Class

CaWCreatePackage

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Example

```
Dim objCreatePackage
Set objCreatePackage = objContext.GetCreatePackage()
lngRet = objCreatePackage.Execute()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWCreatePackage

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objCreatePackage
Set objCreatePackage = objContext.CreatePackage()
If objCreatePackage.IsValid() <> 0 Then
    MsgBox "This is a valid object", vbOKOnly, "Test CreatePackage"
End If
```

CaWCrossProjectMerge

CaWCrossProjectMerge encapsulates the HSDK CrossProjectMerge process. It cannot be instantiated directly. The CrossProjectMerge process can be set in the CaWContext object either by name or object ID. The CaWCrossProjectMerge object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

Set the Project, State, and CrossProjectMerge Context.

- Set the CrossProjectMerge attributes.
- Invoke the Execute function.

Properties

- MergeRule
- Placement
- VersionDescription

Methods

- SetFromStateById
- SetFromStateByName
- GetFromProjectId
- GetFromProjectName
- GetFromStateId
- GetFromStateName

- IsValid
- Execute

More information:

- CHSDK Programming Fundamentals
- CaWContext

MergeRule Property

Gets or sets Rule for how to merge the versions of files in the packages.

Syntax

object.MergeRule = [value]

Owning Class

CaWCrossProjectMerge

Arguments

value - A string that represents the rule.

Notes

The following are the allowable values for the Merge Rule.

T

Take the entire trunk version and ignore the version on the branch. This corresponds to the GUI TakeTrunk Version option.

B

Take the entire branch version and ignore the version on the trunk. This corresponds to the GUI Take Branch Version option.

C

Create a merge tag only if changes in branch and trunk conflict. This corresponds to the GUI Merge Aggressively option.

A

Create a merge tag if there is any difference in the trunk and branch version at all. This corresponds to the GUI Merge Conservatively option.

Example

```
Dim objCrossProjectMerge
Set objCrossProjectMerge = objContext.GetCrossProjectMerge();
objCrossProjectMerge.MergeRule = "A"
strMergeRule = objCrossProjectMerge.MergeRule
```


Placement Property

Gets or sets Rule for how the merged version appears in the destination state.

Syntax

object.Placement = [value]

Owning Class

CaWCrossProjectMerge

Arguments

value - A string that represents the placement.

Notes

The following are the allowable values for the Placement.

B

Creates a merge version on the target branch. This allows changes to be copied from the source project to the target project even if one or more target items are reserved for updates in the main trunk. With this option, a branch will be created to store the changes.

T

Creates a merge version on the target trunk.

A

Creates a merge version on the target trunk or branch. This allows changes to be copied from the source project to the target project even if one or more target items are reserved for update in the main trunk. If items are reserved for update on the trunk, a branch will be created to store the changes. If items are not reserved for update on the trunk, the items are simply copied to the trunk.

Example

```
Dim objCrossProjectMerge
Set objCrossProjectMerge = objContext.GetCrossProjectMerge();
objCrossProjectMerge.Placement = "T"
strMergeRule = objCrossProjectMerge.Placement
```

VersionDescription Property

Gets or sets text description of merged version.

Syntax

object.**VersionDescription** = [value]

Owning Class

CaWCrossProjectMerge

Arguments

value - A string that represents the description.

Example

```
Dim objCrossProjectMerge
Set objCrossProjectMerge = objContext.GetCrossProjectMerge();
objCrossProjectMerge.VersionDescription = "This is a description"
strDescription = objCrossProjectMerge.VersionDescription
```

SetFromStateByName Method

Sets the Project and State that contain the source packages to merge with. These must differ from the Project and State set in the Context.

Syntax

object.**SetFromStateByName**(*strProjectName*, *strStateName*)

Owning Class

CaWCrossProjectMerge

Arguments

strProjectName - A string represents the name of the project to merge with.

strStateName - A string represents the name of a state in *strProjectName* that contains the desired packages.

Return Value

A boolean - True if set succeeded, False if failed.

Notes

A list of available projects may be retrieved from a CaWHarvest object using GetProjectList(). From that list, select a project by name and extract its ObjId. Then, using the CaWHarvest object's GetStateList(ObjId) function, retrieve a list of available states in that project. The name or objid can then be used in SetFromState.

Example

```
Dim objXProjectMerge
Set objXProjectMerge = objContext.GetCrossProjectMerge()
blnRet = ObjXProjectMerge.SetFromState(strProjectName, strStateName)
```

More information:

- CHSDK Programming Fundamentals
- CaWHarvest
- GetProjectList
- GetStateList

SetFromStateById Method

Sets the Project and State that contain the source packages to merge with. These must differ from the Project and State set in the Context.

Syntax

object.**SetFromStateById**(*lngProjectId*, *lngrStateId*)

Owning Class

CaWCrossProjectMerge

Arguments

lngProjectId - A long represents the object ID of the project to merge with.

lngrStateId - A long represents the object ID of a state in the project with lngProjectId that contains the desired packages.

Return Value

None

Notes

A list of available projects may be retrieved from a CaWHarvest object using GetProjectList(). From that list, select a project by name and extract its ObjId. Then, using the CaWHarvest object's GetStateList(ObjId) function, retrieve a list of available states in that project. The name or objid can then be used in SetFromState.

Example

```
Dim objXProjectMerge
Set objXProjectMerge = objContext.GetCrossProjectMerge()
bLnRet = ObjXProjectMerge.SetFromState(lngProjectId, lngStateId)
```

More information:

- CHSDK Programming Fundamentals
- CaWHarvest
- GetProjectList
- GetStateList

GetFromProjectId Method

Returns the ObjID of the Project set by SetFromState.

Syntax

object.GetFromProjectId()

Owning Class

CaWCrossProjectMerge

Arguments

None

Return Value

A long that represents the Project Id containing source state and packages.

Example

```
Dim objXProjectMerge
Set objXProjectMerge = objContext.GetCrossProjectMerge()
lngProjectId = ObjXProjectMerge.GetFromProjectId()
```

More information:

- CHSDK Programming Fundamentals
- CaWHarvest
- SetFromState

GetFromProjectName Method

Returns the Name of the Project set by SetFromState.

Syntax

```
object.GetFromProjectName()
```

Owning Class

CaWCrossProjectMerge

Return Value

A string represents the Project Name containing source state and packages

Example

```
Dim objXProjectMerge
Set objXProjectMerge = objContext.GetCrossProjectMerge()
strProjectName = ObjXProjectMerge.GetFromProjectName()
```

More information:

- CHSDK Programming Fundamentals
- CaWHarvest
- SetFromState

GetFromStateId Method

Returns the ObjID of the State set by SetFromState.

Syntax

object.GetFromStateId()

Owning Class

CaWCrossProjectMerge

Arguments

None

Return Value

A long that represents the State Id containing source state & packages

Example

```
Dim objXProjectMerge
Set objXProjectMerge = objContext.GetCrossProjectMerge()
lngStateId = ObjXProjectMerge.GetFromStateId()
```

More information:

- CHSDK Programming Fundamentals
- CaWHarvest
- SetFromState

GetFromStateName Method

Returns the Name of the State set by SetFromState.

Syntax

object.GetFromStateName()

Owning Class

CaWCrossProjectMerge

Arguments

None

Return Value

A string that represents the State Name containing source state and packages.

Example

```
Dim objXProjectMerge
Set objXProjectMerge = objContext.GetCrossProjectMerge()
strStateName = ObjXProjectMerge.GetFromStateName()
```

More information:

- CHSDK Programming Fundamentals
- CaWHarvest
- SetFromState

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWCrossProjectMerge

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objCrossProjectMerge
Set objCrossProjectMerge = objContext.GetCrossProjectMerge ()
If objCrossProjectMerge.IsValid() <> 0 Then
    msgbox "This is a valid object", vbOKonly, "Test CrossProjectMerge"
End If
```

Execute Method

Executes the CrossProjectMerge process on the Package in the current context with the Packages in the Package Chooser.

Syntax

object.Execute()

Owning Class

CaWCrossProjectMerge

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Notes

Packages in Package Chooser must be from the Project and State specified by SetFromState(). This is accomplished by setting the Project and State the CaWPackageChooser will search with CaWPackageChooser::ProjectObjId(iProject) and CaWPackageChooser::SetStateObjId(iState), and then CaWPackageChooser::Execute (false) to allow searching outside of context.

Example

```
Dim objCrossProjectMerge
Set objCrossProjectMerge = rContext.GetCrossProjectMerge();
lngResult = objCrossProjectMerge.Execute();
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWPackageChooser
- ProjectObjId
- SetStatId
- CaWHarvest
- GetProjectList
- GetStateList

CaWDeleteVersion

CaWDeleteVersion encapsulates the HSDK DeleteVersion process. It cannot be instantiated directly. The DeleteVersion process can be set in the CaWContext object either by name or object ID. The CaWDeleteVersion object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the Project, State, and DeleteVersion Context.
- Select a set Versions to delete using the CaWVersionChooser object.
- Invoke the Execute function.

Methods

- Execute
- IsValid

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWVersionChooser

Execute Method

Executes the DeleteVersion process on the version list currently selected using the Version Chooser.

Syntax

object.Execute()

Owning Class

CaWDeleteVersion

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Example

```
Dim objDeleteVersion
Set objDeleteVersion = objContext.GetDeleteVersion()
' Select versions using VersionChooser
'''
' Execute delete version
lngRet = objDeleteVersion.Execute()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWVersionChooser

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWDeleteVersion

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objDeleteVersion
Set objDeleteVersion = objContext.GetDeleteVersion()
If objDeleteVersion.IsValid() <> 0 Then
    msgbox "This is a valid object", vbOKonly, "Test DeleteVersion"
End If
```

CaWDemote

CaWDemote encapsulates the HSDK Demote process. It cannot be instantiated directly. The Demote process can be set in the CaWContext object either by name or object ID. The CaWDemote object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the Project, State, and Demote Context.
- Select a set of Change Packages to Demote using the CaPackageChooser object.
- Set the Demote attributes.
- Invoke the Execute function.

Methods

- SetCheckPackageDependencies
- SetCheckPackageGroupBind
- GetCheckPackageDependencies
- GetCheckPackageGroupBind
- Execute
- IsValid

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWPackageChooser

SetCheckPackageDependencies Method

Sets Check Package Dependencies flag.

Syntax

object.**SetCheckPackageDependencies**(*blnCheckPackageDependencies*)

Owning Class

CaWDemote

Arguments

blnCheckPackageDependencies - A boolean, True if verify dependencies prior to Demote, False if not.

Return Value

A boolean - True if set succeeded, False if failed.

Example

```
Dim objDemote
objDemote = objContext.GetDemote()
blnRet = objDemote.SetCheckPackageDependencies(True)
```

More information:

- CHSDK Programming Fundamentals
- GetCheckPackageDependencies

SetCheckPackageGroupBind Method

Specifies whether packages in a package group must all be Demoted together.

Syntax

object.**SetCheckPackageGroupBind**(*blnFlag*)

Owning Class

CaWDemote

Arguments

blnFlag - A boolean, True if yes, False if no.

Return Value

A boolean - True if set succeeded, False if failed.

Example

```
Dim objDemote
objDemote = objContext.GetDemote()
blnRet = objDemote.SetCheckPackageGroupBind(True)
```

More information:

- CHSDK Programming Fundamentals
- GetCheckPackageGroupBind

GetCheckPackageDependencies Method

Gets the current value of the Check Package Dependencies flag.

Syntax

object.**GetCheckPackageDependencies()**

Owning Class

CaWDemote

Arguments

None

Return Value

A boolean - Current value of flag.

Example

```
Dim objDemote
objDemote = objContext.GetDemote()
bInRet = objDemote.GetCheckPackageDependencies()
```

More information:

- CHSDK Programming Fundamentals
- SetCheckPackageDependencies

GetCheckPackageGroupBind Method

Gets the current value of the Check Package Group Bind flag.

Syntax

object.**GetCheckPackageGroupBind()**

Owning Class

CaWDemote

Arguments

None

Return Value

A boolean - Current value of flag.

Example

```
Dim objDemote
objDemote = objContext.GetDemote()
bLnRet = objDemote.GetCheckPackageGroupBind()
```

More information:

- CHSDK Programming Fundamentals
- SetCheckPackageGroupBind

Execute Method

Executes the Demote process on the Package list currently selected using the Package Chooser.

Syntax

object.Execute()

Owning Class

CaWDemote

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Example

```
Dim objDemote
objDemote = objContext.GetDemote()
lngRet = objDemote.Execute()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWPackageChooser

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWDemote

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objDemote
Set objDemote = objContext.GetDemote()
If objDemote.IsValid() <> 0 Then
    MsgBox "This is a valid object", vbOKonly, "Test Demote"
End If
```

CaWForm

The CaWForm object is used to retrieve and set values in a form in addition to creating a new form based on an existing form type. After a new form is created, it may be associated with a package. In addition, form attachments may be added or removed from an existing form.

Public Methods

- GetHistory
- GetColumnHasValue
- GetFormType
- GetColumnValue
- SetColumnValue
- GetFormTypeName
- GetFormTypeObjId
- AddPackage

- RemovePackage
- GetPackageList
- GetNotInPackageList
- GetFormAttachmentList
- AddFormAttachment
- RemoveFormAttachment
- GetFormAttachmentById
- Update
- SetFileAgent
- SetFileAgent2
- InitHObject
- Delete
- Get
- SetName

More information:

- CHSDK Programming Fundamentals
- CaWFormType
- CaWFormAttachment

GetHistory Method

Returns a CaWContainer containing form history.

Syntax

object.**GetHistory()**

Owning Class

CaWForm

Return Value

Returns a CaWContainer object

Notes

History attributes are stored with the following hash keys:

- CA_ATTRKEY_ACTION
- CA_ATTRKEY_USERNAME
- CA_ATTRKEY_EXECDTIME

Example

```
Container = Form.GetHistory()
```

More information:

- CHSDK Programming Fundamentals

GetColumnHasValue Method

Returns a boolean indicating if the given column has a value.

Syntax

object.**GetColumnHasValue**(*szColName*)

Owning Class

CaWForm

Arguments

szColName - Name of column.

Return Value

Returns true if column has value, false otherwise

Example

```
bool bRet = Form.GetColumnHasValue("Name")
```

More information:

- CHSDK Programming Fundamentals
- GetColumnValue
- SetColumnValue

GetFormType Method

Retrieves the form type from the form.

Syntax

```
object.GetFormType()
```

Owning Class

CaWForm

Arguments

None

Return Value

Returns a reference to the form type object

Example

```
FormType = Form.GetFormType()
```

More information:

- CHSDK Programming Fundamentals
- CaWFormType

GetColumnValue Method

Retrieves the data associated with the specified column.

Syntax

```
object.GetColumnValue(szColName)
```

Owning Class

CaWForm

Arguments

szColName - name of column to retrieve.

Return Value

Returns a reference to the column data

Example

```
Data = Form.GetColumnValue("Name")
```

More information:

- CHSDK Programming Fundamentals
- GetColumnHasValue
- SetColumnValue

SetColumnValue Method

Sets data into a column.

Syntax

object.**SetcolumnValue**(*szColName*, *rData*)

Owning Class

CaWForm

Arguments

szColName - name of column to set.

rData - data to set.

Return Value

Returns true if success, false if fail.

Example

```
bool bResult = Form.SetColumnValue("Name", rData)
```

More information:

- CHSDK Programming Fundamentals
- GetColumnHasValue
- GetColumnValue

GetFormTypeName Method

Retrieves the form type name.

Syntax

```
object.GetFormTypeName()
```

Owning Class

CaWForm

Arguments

None

Return Value

LPCTSTR - Name of form type

Example

```
LPCTSTR szName = Form.GetFormTypeName()
```

More information:

- CHSDK Programming Fundamentals
- CaWFormType

GetFormTypeObjId Method

Retrieves the Obj ID of the form type.

Syntax

```
object.GetFormTypeObjId()
```

Owning Class

CaWForm

Arguments

None

Return Value

HAROBJID - Obj Id of the form type.

Example

```
ObjId = Form.GetFormTypeObjId()
```

More information:

- CHSDK Programming Fundamentals
- CaWFormType

AddPackage Method

Adds a package to this form.

Syntax

```
object.AddPackage(pPackage)
```

Owning Class

CaWForm

Arguments

pPackage - package to add.

Return Value

Returns true if success, false if fail.

Example

```
bool bResult = Form.AddPackage(pPackage)
```

More information:

- CHSDK Programming Fundamentals
- RemovePackage
- GetPackageList
- GetNotInPackageList

RemovePackage Method

Remove a package from this form.

Syntax

object.**RemovePackage**(*rPackage*)

Owning Class

CaWForm

Arguments

rPackage - Package to remove from form.

Return Value

Returns true if success, false if fail

Example

```
bool bResult = Form.RemovePackage(rPackage)
```

More information:

- CHSDK Programming Fundamentals
- AddPackage
- GetPackageList
- GetNotInPackageList

GetPackageList Method

Retrieves package list associated with form.

Syntax

object.**GetPackageList**()

Owning Class

CaWForm

Return Value

Returns a reference to a package list

Example

```
List = Form.GetPackageList()
```

More information:

- CHSDK Programming Fundamentals
- AddPackage
- RemovePackage
- GetNotInPackageList

GetNotInPackageList Method

Retrieves package list not associated with form.

Syntax

```
object.GetNotInPackageList()
```

Owning Class

CaWForm

Return Value

Returns a reference to a package list

Example

```
List = Form.GetNotInPackageList()
```

More information:

- CHSDK Programming Fundamentals
- AddPackage
- RemovePackage
- GetPackageList

GetFormAttachmentList Method

Retrieves a list of form attachments.

Owning Class

CaWForm

Return Value

Returns a CaWContainer object

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME, CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME

Example

```
Container = Form.GetFormAttachmentList()
```

More information:

- CHSDK Programming Fundamentals
- CaWFormAttachment
- AddFormAttachment
- RemoveFormAttachment
- GetFormAttachmentById

AddFormAttachment Method

Adds an attachment to a form.

Syntax

```
object.AddFormAttachment(szFile, iType, bCompress)
```

Owning Class

CaWForm

Arguments

szFile - attachment name.

iType - attachment type.

bCompress - flag if attachment should be compressed.

Return Value

Returns true on success, false on fail.

Example

```
bResult = Form.AddFormAttachment  
(szFile, ATTACHMENT_FILE, false)
```

More information:

- CHSDK Programming Fundamentals
- GetFormAttachmentList
- RemoveFormAttachment
- GetFormAttachment
- GetFormAttachmentById
- SetFileAgent

RemoveFormAttachment Method

Removes an attachment associated with a form.

Syntax

```
object.RemoveFormAttachment(szFile)
```

Owning Class

CaWForm

Arguments

szFile - attachment name.

Return Value

Returns true on success, false on fail

Example

```
bool bResult = Form.RemoveFormAttachment(szFile)
```

More information:

- CHSDK Programming Fundamentals
- GetFormAttachmentList
- GetFormAttachment
- RemoveFormAttachment
- GetFormAttachmentById

RemoveFormAttachment Method

Removes an attachment associated with a form.

Syntax

object.**RemoveFormAttachment**(*iObjId*)

Owning Class

CaWForm

Arguments

iObjId - Obj id of attachment to remove.

Return Value

Returns true on success, false on fail.

Example

```
bResult = Form.RemoveFormAttachment(ObjId)
```

More information:

- CHSDK Programming Fundamentals
- GetFormAttachmentList
- GetFormAttachment
- RemoveFormAttachment
- GetFormAttachmentById

GetFormAttachmentById Method

Retrieve a form attachment by object ID.

Syntax

object.GetformAttachmentById(*iObjId*)

Owning Class

CaWForm

Arguments

iObjId - object ID of attachment to retrieve

Return Value

Returns form attachment pointer

Example

```
pFA = Form.GetformAttachmentById(ObjId)
```

More information:

- CHSDK Programming Fundamentals
- GetFormAttachmentList
- GetFormAttachment
- RemoveFormAttachment

Update Method

Updates the form.

Syntax

object.**Update()**

Owning Class

CaWForm

Return Value

Zero if success, non-zero if failure.

Example

```
iResult = Form.Update()
```

More information:

- CHSDK Programming Fundamentals

Get Method

Retrieves the form.

Syntax

object.**Get()**

Owning Class

CaWForm

Arguments

None

Return Value

Zero if success, non-zero if failure.

Example

```
iResult = Form.Get()
```

More information:

- CHSDK Programming Fundamentals

Delete Method

Deletes the form.

Syntax

object.Delete()

Owning Class

CaWForm

Arguments

None

Return Value

Zero if success, non-zero if failure.

Example

```
iResult = Form.Delete()
```

More information:

- CHSDK Programming Fundamentals

SetName Method

Sets the form name

Syntax

object.SetName(*sName*)

Owning Class

CaWForm

Arguments

sName - new name for form

Return Value

True on success, false on failure

Example

```
bResult = Form.SetName(sName)
```

More information:

- CHSDK Programming Fundamentals

InitHObject Method

Initialize encapsulated Form object.

Syntax

```
object.InitHObject()
```

Owning Class

CaWForm

Arguments

CaWFormType - the form type to base the form on

Name - new name of form

Return Value

A boolean - True if succeeded, False if failed

Example

```
Dim objForm
Set objForm = CreateObject("Chsdk.CaWForm")
blnRet = objFormt.InitHObject(objFormType, "new name")
```

More information:

- HUtilities Programming Fundamentals

SetFileAgent Method

Sets information used for accessing a remote system.

Syntax

```
object.SetFileAgent(szRemoteMachine, szRemoteUser, szRemotePassword,  
szClientPath)
```

Owning Class

CaWForm

Arguments

szRemoteMachine - remote machine name.

szRemoteUser - remote user name.

szRemotePassword - remote password.

szClientPath - path to client file.

Return Value

None

Example

```
Form.SetFileAgent(szBroker, szUser, szPassword, szPath)
```

More information:

- CHSDK Programming Fundamentals
- AddFormAttachment

SetFileAgent2 Method

Sets information used for accessing a remote system.

Syntax

```
object.SetFileAgent2(szRemoteMachine, szRemoteUser, szRemotePassword,  
szClientPath, szPortNumber)
```

Owning Class

CaWForm

Arguments

szRemoteMachine - remote machine name.

szRemoteUser - remote user name.

szRemotePassword - remote password.

szClientPath - path to client file.

szPortNumber – agent port number.

Return Value

None

Example

```
Form.SetFileAgent2(szBroker, szUser, szPassword, szPath, "7001")
```

More information:

[CHSDK Programming Fundamentals](#) (see page 541)

[CaWFormAttachment](#) (see page 709)

CaWFormType

The CaWFormType object is used to retrieve default values associated with a CA Harvest SCM form type. The CaWFormType can be retrieved from a specific CaWForm, or a list of form types can be retrieved from the CaWHarvest session.

The CaWFormType object must be retrieved from an existing form or CA Harvest SCM session.

Public Methods

- GetColumnDefaultValue
- GetTableName
- GetColumnIndex
- GetColumnMaxLength
- GetColumnName
- GetColumnType
- RemoveColumn
- GetColumnCount

More information:

- CHSDK Programming Fundamentals

GetColumnDefaultValue Method

Retrieves the default value at the specified index.

Syntax

object.GetColumnDefaultValue(*iIndex*)

Owning Class

CaWFormType

Arguments

iIndex - index to retrieve.

Return Value

Reference to the column data.

Example

```
rData = FormType.GetColumnDefaultValue(iIndex)
```

More information:

- CHSDK Programming Fundamentals
- GetColumnType
- GetColumnCount
- GetColumnName

GetTableName Method

Retrieves the table name.

Syntax

```
object.GetTableName()
```

Owning Class

CaWFormType

Arguments

None

Return Value

LPCTSTR - name of table.

Example

```
LPCTSTR szTable = FormType.GetTableName()
```

More information:

- CHSDK Programming Fundamentals

GetColumnIndex Method

Retrieves an index based on a column name.

Syntax

object.**GetColumnIndex**(*szColumnName*)

Owning Class

CaWFormType

Arguments

szColumnName - column name to use.

Return Value

HINT32 - index of specified column name.

Example

```
iResult = FormType.GetColumnIndex(szColumnName)
```

More information:

- CHSDK Programming Fundamentals
- GetColumnName
- GetColumnType
- GetColumnCount

GetColumnMaxLength Method

Retrieves the maximum length of the given column.

Syntax

object.**GetColumnMaxLength**(*iIndex*)

Owning Class

CaWFormType

Arguments

iIndex - index of column length to retrieve.

Return Value

HINT32 - length of column at specified index.

Example

```
iResult = FormType.GetColumnMaxLength(iIndex)
```

More information:

- CHSDK Programming Fundamentals
- GetColumnName
- GetColumnType
- GetColumnCount

GetColumnName Method

Retrieves the column name at the specified index.

Syntax

object.**GetColumnName**(*iIndex*)

Owning Class

CaWFormType

Arguments

iIndex - column index to retrieve name from.

Return Value

LPCTSTR - name at specified index.

Example

```
LPCTSTR szName = FormType.GetColumnName(iIndex)
```

More information:

- CHSDK Programming Fundamentals
- GetColumnType
- GetColumnCount
- GetColumnMaxLength

GetColumnType Method

Retrieve the type of a column.

Syntax

```
object.GetColumnType(iIndex)
```

Owning Class

CaWFormType

Arguments

HINT32 - index of column type to retrieve.

Return Value

CaData::Types - data type at index.

Example

```
eType = FormType.GetColumnType(iIndex)
```

More information:

- CHSDK Programming Fundamentals
- GetColumnName
- GetColumnCount

GetColumnCount Method

Retrieves the number of columns available.

Syntax

```
object.GetColumnCount()
```

Owning Class

CaWFormType

Arguments

None

Return Value

HUIN32 - number of columns available.

Example

```
uResult = FormType.GetColumnCount()
```

More information:

- CHSDK Programming Fundamentals
- GetColumnName
- GetColumnType

CaWFormAttachment

The CaWFormAttachment object is used to get and set attributes associated with a form attachment, as well as retrieve the data associated with a form attachment. The form attachment values can be retrieved from the form with CaWForm GetFormAttachmentList(). Specific attachments can then be retrieved with CaWForm GetFormAttachmentById().

The CaFormAttachment object can only be retrieved from an existing form.

Public Methods

- GetCompression
- SetCompression
- GetAttachmentType
- SetAttachmentType
- SetFormObjId
- GetFileSize
- CopyFileData
- SetFileAgent
- SetFileAgent2

More information:

- CHSDK Programming Fundamentals

GetCompression Method

Retrieves flag indicating if compression is used.

Syntax

object.**GetCompression**()

Owning Class

CaWFormAttachment

Arguments

None

Return Value

Returns true on success, false on fail.

Example

```
bool bResult = FormAttachment.GetCompression()
```

More information:

- CHSDK Programming Fundamentals
- SetCompression

SetCompression Method

Sets flag denoting if compression to be used.

Syntax

object.**SetCompression**(*bCompression*)

Owning Class

CaWFormAttachment

Arguments

True if compression used, false if not used.

Return Value

None

Example

```
FormAttachment.SetCompression(false)
```

More information:

- CHSDK Programming Fundamentals
- GetCompression

GetAttachmentType Method

Retrieves the attachment type.

Syntax

```
object.GetAttachmentType()
```

Owning Class

CaWFormAttachment

Arguments

None

Return Value

HINT32 - attachment type: ATTACHMENT_FILE, ATTACHMENT_URL

Example

```
HINT32 iResult = FormAttachment.GetAttachmentType()
```

More information:

- CHSDK Programming Fundamentals
- SetAttachmentType

SetAttachmentType Method

Sets the form attachment type.

Syntax

object.SetAttachmentType(*iType*)

Owning Class

CaWFormAttachment

Arguments

HINT32 - ATTACHMENT_FILE, ATTACHMENT_URL.

Return Value

None

Example

FormAttachment.SetAttachmentType(ATTACHMENT_FILE)

More information:

- CHSDK Programming Fundamentals
- GetAttachmentType

SetFormObjId Method

Sets the objid of the attachment.

Syntax

object.SetFormObjId(*ObjId*)

Owning Class

CaWFormAttachment

Arguments

ObjId - Object Id to set.

Return Value

None

Example

FormAttachment.SetFormObjId(ObjId)

More information:

- CHSDK Programming Fundamentals

GetFileSize Method

Retrieves the file size of the attachment.

Syntax

object.**GetFileSize()**

Owning Class

CaWFormAttachment

Arguments

None

Return Value

HINT32 - file size.

Example

```
HINT32 iFileSize = FormAttachment.GetFileSize()
```

More information:

- CHSDK Programming Fundamentals
- CopyFileData

CopyFileData Method

Copies file data.

Syntax

object.**CopyFileData()**

Owning Class

CaWFormAttachment

Arguments

None

Return Value

Returns true on success, false on fail.

Example

```
bool bResult = FormAttachment.CopyFileData()
```

More information:

- CHSDK Programming Fundamentals
- GetFileSize

SetFileAgent Method

Sets information used for accessing a remote system.

Syntax

```
object.SetFileAgent(szRemoteMachine, szRemoteUser, szRemotePassword,  
szClientPath)
```

Owning Class

CaWFormAttachment

Arguments

szRemoteMachine - remote machine name.

szRemoteUser - remote user name.

szRemotePassword - remote password.

szClientPath - client path.

Return Value

None

Example

```
FormAttachment.SetFileAgent(szBroker, szUser, szPassword, szPath)
```

More information:

- CHSDK Programming Fundamentals
- CopyFileData

SetFileAgent2 Method

Sets information used for accessing a remote system.

Syntax

object.**SetFileAgent2**(*szRemoteMachine*, *szRemoteUser*, *szRemotePassword*, *szClientPath*, *szPortNumber*)

Owning Class

CaWForm

Arguments

szRemoteMachine - remote machine name.

szRemoteUser - remote user name.

szRemotePassword - remote password.

szClientPath - path to client file.

szPortNumber – agent port number.

Return Value

None

Example

```
Form.SetFileAgent2(szBroker, szUser, szPassword, szPath, "7001")
```

More information:

[CHSDK Programming Fundamentals](#) (see page 541)

[CaWFormAttachment](#) (see page 709)

CaWHarvest

CaWHarvest is the top level CHSDK object that provides the CA Harvest SCM login and initializes the CHSDK and communications layers. It maintains a static log stream. It also provides access to the Project and User lists.

Methods

- Login
- Logout
- GetAuthenticationSetting
- GetCaseSensLogin

- GetProductInfo
- GetProjectList
- GetUserList
- SetLogName
- GetContext
- GetBrokerName
- GetStateList

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWContainer

SetBroker Method

Connects to the CA Harvest SCM Broker.

Syntax

```
object.SetBroker(strBrokerName)
```

Owning Class

CaWHarvest

Arguments

strBrokerName—A string that represents the name of the CA Harvest SCM Broker.

Return Value

A long, zero if success, nonzero if failure.

Note: The SetBroker method must be called once after the CaWHarvest object is created. It will establish a connection to the RTserver where the broker connects to.

Example

```
Dim sobjHarvest
Set sobjHarvest = CreateObject("Chsdk.CaWHarvest")
If sobjHarvest Is Nothing Then
msgbox "Can't create Harvest.", vbOKOnly + vbExclamation, "Error"
Exit Sub
End If
LngRet = sobjHarvest.SetBroker("CMBroker")
```

Login Method

Logs into the specified CA Harvest SCM Broker.

Syntax

object.Login(strBrokerName, strUserName, strPassword)

Owning Class

CaWHarvest

Arguments

strBrokerName - A string that represents the name of the CA Harvest SCM Broker.

strUserName - A string that represents the User Name.

strPassword - A string that represents the Password.

Return Value

A long, zero if success, Nonzero if failure. GetLastMessage will give the last message from the server for both success and failure.

Example

```
Dim sobjHarvest
Set sobjHarvest = CreateObject("Chsdk.CaWHarvest")
If sobjHarvest Is Nothing Then
    MsgBox "Can't create Harvest.", vbOKOnly + vbExclamation, "Error"
Exit Sub

End If
LngRet = sobjHarvest.Login("CMBroker", "CMUser", "MyPassword")
```

GetLastMessage Method

Returns the last message from the server for Login or ChangePassword requests.

Syntax

object.GetLastMessage()

Owning Class

CaWHarvest

Arguments

None

Return Value

A string that represents the last message received from the server concerning Login or ChangePassword methods.

Example

```
Dim subjHarvest
Set subjHarvest = CreateObject("Chsdk.CaWHarvest")
If subjHarvest Is Nothing Then
    MsgBox "Can't create Harvest.", vbOKOnly + vbExclamation, "Error"
Exit Sub

End If
LngRet = subjHarvest.Login("CMBroker", "CMUser", "MyPassword")
MsgBox "Login Result: ", subjHarvest.GetLastMessage()
```

ChangePassword Method

Changes the password of the current user.

Syntax

object.**ChangePassword**(szNewPassword)

Owning Class

CaWHarvest

Arguments

szNewPassword - A string that represents the user's new password.

Return Value

A long, zero if success, Nonzero if failure. GetLastMessage will give the last message from the server for both success and failure.

Notes

- If your site uses internal authentication (CA Harvest SCM authentication), changing the password is enabled.
- If your site uses external authentication, such as Microsoft Active Directory, this function is disabled; you must change the password using methods provided by the authentication server. For example, if using Microsoft Active Directory, you can change passwords by entering Ctrl+Alt+Del and using the Change Password option.
- If your site uses external authentication, changing passwords fails with an error message.

Exceptions

Throws CException on error.

Example

```
Dim sobjHarvest
Set sobjHarvest = CreateObject("Chsdk.CaWHarvest")
If sobjHarvest Is Nothing Then
    MsgBox "Can't create Harvest.", vbOKOnly + vbExclamation, "Error"
Exit Sub

End If
LngRet = sobjHarvest.Login("CMBroker", "CMUser", "MyPassword")
sobjHarvest.ChangePassword("MyNewPassword")
```

More information:

- GetCaseSensLogin

Logout Method

Log out from the CA Harvest SCM Broker. Clear the context and close the output log file.

Syntax

object.Logout()

Owning Class

CaWHarvest

Arguments

None

Return Value

None

Example

```
sobjHarvest.Logout();
```

More information:

- CHSDK Programming Fundamentals
- Login

GetBrokerName Method

Retrieves the name of the current CA Harvest SCM service broker.

Syntax

```
object.GetBrokerName()
```

Owning Class

CaWHarvest

Arguments

None

Return Value

A string that represents the Broker name, empty string if not set.

Example

```
strName = sobjHarvest.GetBrokerName();
```

More information:

- CHSDK Programming Fundamentals

GetContext Method

Retrieves the current context object.

Syntax

object.GetContext()

Owning Class

CaWHarvest

Arguments

None

Return Value

A CaWContext object that represents the context object, null object if error.

Example

```
Dim objContext  
objContext = sobjHarvest.GetContext()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetAuthenticationSetting Method

Retrieves the authentication server flag for the requested setting.

Syntax

object.GetAuthenticationSetting(*nSetting*)

Owning Class

CaWHarvest

Arguments

nSetting - An enumeration of the authentication server setting to retrieve. Possible values include:

- AUTHSERV_EDIT_PASSWORD
- AUTHSERV_EDIT_USERNAME
- AUTHSERV_EDIT_REALNAME
- AUTHSERV_EDIT_PHONE

- AUTHSERV_EDIT_EXT
- AUTHSERV_EDIT_FAX
- AUTHSERV_EDIT_EMAIL
- AUTHSERV_EDIT_NOTE
- AUTHSERV_GET_PASSPOLICY
- AUTHSERV_EDIT_PASSPOLICY
- AUTHSERV_AUTOCREATE_USERNAME

Return Value

A boolean denoting if the setting is True or False.

Example

```
blnRet = sobjHarvest.GetAuthenticationSetting(AUTHSERV_EDIT_PASSWORD)
```

More information:

- CHSDK Programming Fundamentals
- GetCaseSensLogin

GetCaseSensLogin Method

Retrieves the authentication server flag, denoting if case sensitivity is used for operations pertaining to the login name.

Syntax

```
object.GetCaseSensLogin()
```

Owning Class

CaWHarvest

Arguments

None

Return Value

A boolean denoting authentication server setting.

Example

```
blnRet = sobjHarvest.GetCaseSensLogin()
```

More information:

- CHSDK Programming Fundamentals
- GetAuthenticationSetting

GetProductInfo Method

Retrieves a CaWContainer object that contains the CA Harvest SCM product information attributes in the list.

Syntax

object.GetProductInfo()

Owning Class

CaWHarvest

Arguments

None

Return Value

A CaWContainer object, null object if error.

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_CLIENT_VERSION
- CA_ATTRKEY_CLIENT_BUILD
- CA_ATTRKEY_CLIENT_PATCH

Example

```
Dim objProductInfo  
objProductInfo = sobjHarvest.GetProductInfo()
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

GetProjectList Method

Retrieve a CaWContainer object that contains the attributes for each CA Harvest SCM Project in the list.

Syntax

object.GetProjectList()

Owning Class

CaWHarvest

Arguments

None

Return Value

A CaWContainer object, null object if error.

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME

Example

```
Dim objProjectList
objProjectList = sobjHarvest.GetProjectList()
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

GetUserList Method

Retrieves a CaWContainer object that contains the attributes for each CA Harvest SCM User in the list.

Syntax

object.**GetUserList()**

Owning Class

CaWHarvest

Arguments

None

Return Value

A CaWContainer object, null object if error.

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME
- CA_ATTRKEY_REALNAME
- CA_ATTRKEY_EMAIL

Example

```
Dim objUserList
objUserList = sobjHarvest.GetUserList()
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

SetLogName Method

The SetLogName method sets the file name for the output log and open the file. If log file name is not set, all output messages will be discarded.

Syntax

```
object.SetLogName()
```

Owning Class

CaWHarvest

Arguments

A string that represents the output log file name.

Return Value

A boolean, True if success, False if fail.

Example

```
Dim subjHarvest
Set subjHarvest = CreateObject("Chsdk.CaWHarvest")
If subjHarvest Is Nothing Then

    MsgBox "Can't create Harvest.", vbOKOnly + vbExclamation, "Error"
Exit Sub

End If
bLnRet = subjHarvest.SetLogName("C:\temp\vblog.txt")
lngRet = subjHarvest.Login("CMBroker", "CMUser", "MyPassword")
```

More information:

- CHSDK Programming Fundamentals

GetStateList Method

Retrieves a CaWContainer object that contains the attributes for each CA Harvest SCM State in the CA Harvest SCM Project specified by the lngProjectId argument.

Syntax

object.**GetStateList**(lngProjectId)

Owning Class

CaWHarvest

Arguments

lngProjectId - A long represents the project object ID of the desired project that contains this statelist.

Return Value

A CaWContainer object, null object if error.

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID

- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME

Example

```
Dim objStateList  
objStateList = sobjHarvest.GetStateList()
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

CaWInteractiveMerge

CaWInteractiveMerge encapsulates the HSDK Interactive Merge process. It cannot be instantiated directly.

The InteractiveMerge process can be set in the CaWContext object either by name or object ID. The CaWInteractiveMerge object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the project, state, merged-tagged version item, and interactive merge context.
- Interactive Merge operates only on the merged-tagged version item in the current context. Interactive Merge involves three steps: check out, resolve the conflict with the extensive tool, and check in the resolved file.

Methods

- GetCompress
- GetDescription
- GetExecuteFlag
- GetParentPackageName
- GetParentStateName
- GetParentMappedVer

- GetMergedEnvName
- GetMergedStateName
- GetMergedPackageName
- GetMergedMappedVer
- GetExtMergeToolFlag
- GetMergedFile
- GetResolveFlag
- GetScriptFile
- SetDescription
- SetExecuteFlag
- SetExtMergeToolFlag
- SetMergedFile
- SetResolveFlag
- SetScriptFile
- Execute
- Resolve

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetCompress Method

Gets the Compress status of the merge-tagged file.

Syntax

object.**GetCompress()**

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

Y-if compressed, N-if not

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetCompress()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetDescription Method

Gets the current value of the Interactive Merge Description.

Syntax

object.**GetDescription()**

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the description

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetDescription()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetDescription

GetExecuteFlag Method

Gets the interactive merge options, 3-way or 2-way.

Syntax

object.**GetExecuteFlag**()

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

0

Specifies a 3-way merge

1

Specifies a 2-way merge

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
lngRet = objInteractiveMerge.GetExecuteFlag()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetExecuteFlag

GetParentPackageName Method

Gets the parent package name of the merge-tagged item.

Syntax

object.**GetParentPackageName()**

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the parent package name.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetParentPackageName()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetParentStateName Method

Gets the parent state name of the merge-tagged item.

Syntax

object.GetParentStateName()

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the parent state name.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetParentStateName()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetParentMappedVer Method

Gets the parent mapped version of the merge-tagged item.

Syntax

```
object.GetParentMappedVer()
```

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the parent mapped version name.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetParentMappedVer()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetMergedEnvName Method

Gets the current merged project name.

Syntax

```
object.GetMergedEnvName()
```

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the current merged project name.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetMergedEnvName()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetMergedStateName Method

Gets the merged state name.

Syntax

object.**GetMergedStateName()**

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the current merged state name.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetMergedStateName()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetMergedPackageName Method

Gets the current merged package name.

Syntax

object.**GetMergedPackageName()**

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the current merged package name.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetMergedPackageName()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetMergedMappedVer Method

Gets the current merged mapped version.

Syntax

object.**GetMergedMappedVer()**

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the current merged mapped version.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetMergedMappedVer()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetExtMergeToolFlag Method

Gets the current merged tool flag.

Syntax

```
object.GetMergedMappedVer()
```

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

True-if set, false-if not set.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
bRet = objInteractiveMerge.GetExtMergeToolFlag()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetExtmergeToolFlag

GetMergedFile Method

Gets the merged file.

Syntax

object.**GetMergedFile()**

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the merged file name

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetMergedFile()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetMergedFile

GetResolveFlag Method

Gets the resolve flag.

Syntax

object.**GetResolveFlag()**

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value**T**

Trunk

B

Branch

C

Cancel

M

Merge

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetResolveFlag()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetResolveFlag

GetScriptFile Method

Gets the script file.

Syntax

object.**GetScriptFile()**

Owning Class

CaWInteractiveMerge

Arguments

None

Return Value

A string that represents the script file name

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
strRet = objInteractiveMerge.GetScriptFile()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- SetScriptFile

SetDescription Method

Sets the description.

Syntax

object.**SetDescription**(*strDesc*)

Owning Class

CaWInteractiveMerge

Arguments

A string that represents the description

Return Value

True-if successful, false-if failed

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
bRet = objInteractiveMerge.SetDescription("My Description")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetDescription

SetExecuteFlag Method

Sets the execute flag.

Syntax

object.**SetExecuteFlag**(*lngFlag*)

Owning Class

CaWInteractiveMerge

Arguments**0**

Specifies a 3-way merge

1

Specifies a 2-way merge

Return Value

None

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
bRet = objInteractiveMerge.SetExecuteFlag(0)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetExecuteFlag

SetExtMergeToolFlag Method

Sets the extensive merge tool.

Syntax

object.**SetExtMergeToolFlag**(*bFlag*)

Owning Class

CaWInteractiveMerge

Arguments

true

Sets extensive merge tool.

false

Unsets extensive merge tool.

Return Value

None

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
objInteractiveMerge.SetExtMergeToolFlag(true)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetExtMergeToolFlag

SetMergedFile Method

Sets merged file.

Syntax

object.**SetMergedFile**(*strFile*)

Owning Class

CaWInteractiveMerge

Arguments

A string that represents Merged File

Return Value

None

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
objInteractiveMerge.SetMergedFile("mergeFile.cpp")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetMergedFile

SetResolveFlag Method

Sets resolved flag.

Syntax

object.**SetMergedFile**(*strFlag*)

Owning Class

CaWInteractiveMerge

Arguments**T**

Trunk

B

Branch

C

Cancel

M

Merge

Return Value

None

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
objInteractiveMerge.SetResolveFlag("M")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetResolveFlag

SetScriptFile Method

Sets script file.

Syntax

object.**SetScriptFile**(*strFile*)

Owning Class

CaWInteractiveMerge

Arguments

A string that represents the script file name.

Return Value

None

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
objInteractiveMerge.SetScriptFile("Script")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- GetScriptFile

Execute Method

Executes an interactive merge.

Syntax

object.**Execute**(*lngVerId*)

Owning Class

CaWInteractiveMerge

Arguments

A number that represents the version id of the merge-tagged item

Return Value

0-if successful, non-zero-if failed.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
objInteractiveMerge.SetScriptFile("script");
objInteractiveMerge.SetExecuteFlag(0);
objInteractiveMerge.Execute(81);
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

Resolve Method

Resolves the merge-tagged file.

Syntax

object.**Resolve**(*lngVerId*)

Owning Class

CaWInteractiveMerge

Arguments

IngVerID-version id of the merge-tagged item.

Return Value

0-if successful, non-zero-if failed.

Notes

None

Example

```
Dim objInteractiveMerge
Set objInteractiveMerge = objContext.GetInteractiveMerge()
objInteractiveMerge.SetScriptFile("script");
objInteractiveMerge.SetExecuteFlag(0);
objInteractiveMerge.Execute(81);
objInteractiveMerge.SetMergedFile( "Resolve" );
objInteractiveMerge.SetResolveFlag( "M" );
objInteractiveMerge.Resolve(81);
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

CaWPackage

CaWPackage encapsulates the HSDK Package object. It cannot be instantiated directly. The CaWPackage object can be set in the CaWContext object either by name or object ID. It can only be retrieved from the CaWContext after it has been set.

Usage

- Delete and Update the package.

Methods

- Delete
- Update

More information:

- CHSDK Programming Fundamentals
- CaWContext

Delete Method

Deletes the current package.

Syntax

object.Delete()

Owning Class

CaWPackage

Arguments

None

Return Values

A boolean - True if succeeded, False if failed.

Example

```
Dim objPackage
Set objPackage = objContext.GetPackage()
blnRet = objPackage.Delete()
```

More information:

- CHSDK Programming Fundamentals

Update Method

Updates the current package.

Syntax

object.Update()

Owning Class

CaWPackage

Arguments

None

Return Values

A boolean - True if succeeded, False if failed.

Example

```
Dim objPackage
Set objPackage = objContext.GetPackage()
blnRet = objPackage.Update()
```

More information:

- CHSDK Programming Fundamentals

CaWPackageChooser

CaWPackageChooser cannot be instantiated directly. The CaWPackageChooser object can only be retrieved from the CaWContext.

The Package Chooser provides a mechanism to select a set of packages based upon filtered criteria. Each Package Chooser attribute adds an additional restriction on the selected package set. The resulting package list is available to CaWApprove, CaWPromote, and CaWDemote processes in the current context. A list of attributes from the selected versions can be retrieved in the form of a CaWContainer.

Usage

- Optionally set the Project and State Context.
- Set the desired Package Chooser attributes.
- Invoke the Execute function with the bUseCurrentContext set to either use the current context or override it.

Properties

- ProjectObjId
- StateObjId
- PackageGroupObjId
- AssigneeObjId
- FromDate
- ToDate
- CreatorObjId
- ModifierObjId

- ApproveUserObjId
- Priority
- SubString
- PkgStatus
- PkgEntryFromDate
- PkgEntryToDate

Methods

- Clear
- GetPackageList
- Execute
- IsValid
- SetPackageName

More information:

- CHSDK Programming Fundamentals
- CaContext
- CaContainer
- CaCheckin
- CaCheckout
- CaDeleteVersion

ProjectObjId Property (Write Only)

Sets Project Object ID if not searching in current context.

Syntax

object.**ProjectObjId** = [value]

Owning Class

CaWPackageChooser

Arguments

value - A long that represents the Project Object Id.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.ProjectObjId = 10024
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- Execute

StateObjId Property (Write Only)

Sets State Object ID if not searching within current context.

Syntax

object.**StateObjId** = [*value*]

Owning Class

CaWPackageChooser

Arguments

value - A long that represents the State Object Id.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.StateObjId = 10024
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- Execute

PkgGroupObjId Property (Write Only)

Set Package Group Object ID.

Syntax

object.PkgGroupObjId = [value]

Owning Class

CaWPackageChooser

Arguments

value - A long that represents the Package Group Id.

Example

```
Dim objPackageChooser  
objPackageChooser = rContext.GetPackageChooser()  
objPackageChooser.PkgGroupObjId = 10024
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

AssigneeObjId Property (Write Only)

Set Package Assignee Object Id.

Syntax

object.AssigneeObjId = [value]

Owning Class

CaWPackageChooser

Arguments

value - A long that represents the Assignee Object Id.

Example

```
Dim objPackageChooser  
objPackageChooser = rContext.GetPackageChooser()  
objPackageChooser.AssigneeObjId = 10024
```

FromDate Property (Write Only)

Sets the From Date value of the date range.

Syntax

object.**FromDate** = [value]

Owning Class

CaWPackageChooser

Arguments

value - A CaWTimeStamp object that represents the time stamp value.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.FromDate = objTimeStamp
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWTimeStamp
- ToDate

ToDate Property (Write Only)

Sets the To Date value of the date range.

Syntax

object.**ToDate** = [value]

Owning Class

CaWPackageChooser

Arguments

value - A CaWTimeStamp object that represents the time stamp value.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.FromDate = objTimeStamp
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWTimeStamp
- FromDate

CreatorObjId Property (Write Only)

Set Package Creator Object Id.

Syntax

object.**CreatorObjId** = [*value*]

Owning Class

CaWPackageChooser

Arguments

value - A long that represents the Creator Object ID.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.CreatorObjId = 10024
```

ModifierObjId Property (Write Only)

Sets Package Modifier Object ID.

Syntax

object.**ModifierObjId** = [*value*]

Owning Class

CaWPackageChooser

Arguments

value - A long that represents the most recent Modifier Object Id.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.ModifierObjId = 10024
```

ApproveUserObjId Property (Write Only)

Sets Package Approve User Object Id.

Syntax

object.**ApproveUserObjId** = [*value*]

Owning Class

CaWPackageChooser

Arguments

value - A long that represents the Id of user who has approved packages.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.ApproveUserObjId = 10024
```

Priority Property (Write Only)

Sets Package Priority.

Syntax

object.**ApproveUserObjId** = [*value*]

Owning Class

CaWPackageChooser

Arguments

value - A long that represents the User defined priority value to search for.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.Priority = 1
```

SubString Property (Write Only)

Sets substring expression for searching the Note attribute of packages.

Syntax

object.**SubString** = [value]

Owning Class

CaWPackageChooser

Arguments

value - A substring expression.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.SubString = "**required for Version 1.5**"
```

PkgStatus Property (Write Only)

Sets the Package Status Option to search by Package approval status.

Syntax

object.**PkgStatus** = [value]

Owning Class

CaWPackageChooser

Arguments

value - A long that represents the Package Approval Status attribute.

Notes

The following are the allowable values for the Package Status Option:

0

Get only approved packages.

1

Get only rejected packages.

2

Get packages that have not been approved or rejected.

3

Get packages with any approval status.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.PkgStatus = APPROVE_STATUS_APPROVED
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

PkgEntryFromDate Property (Write Only)

Sets the State Entry From Date value of the date range.

Syntax

object.**PkgEntryFromDate** = [*value*]

Owning Class

CaWPackageChooser

Arguments

value - A CaWTimeStamp object that represents the time stamp value.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.PkgEntryFromDate = objTimeStamp
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWTimeStamp
- PkgEntryToDate

PkgEntryToDate Property (Write Only)

Sets the State Entry To Date value of the date range.

Syntax

object.**PkgEntryToDate** = [*value*]

Owning Class

CaWPackageChooser

Arguments

value - A CaWTimeStamp object that represents the time stamp value.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.PkgEntryToDate = objTimeStamp
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWTimeStamp
- PkgEntryFromDate

Clear Method

Removes all selection criteria.

Syntax

object.Clear()

Owning Class

CaWPackageChooser

Arguments

None

Return Value

None

Example

```
Dim objPackageChooser
objPackageChooser = objContext.GetPackageChooser()
objPackageChooser.Clear()
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetPackageList Method

Retrieves a CaWContainer object that contains the attributes for each CA Harvest SCM Package in the list.

Syntax

object.GetPackageList()

Owning Class

CaWPackageChooser

Arguments

None

Return Value

Returns a CaWContainer object, null object if error.

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID
- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME
- CA_ATTRKEY_PROJECTID
- CA_ATTRKEY_STATEID
- CA_ATTRKEY_ASSIGNEEID
- CA_ATTRKEY_ASSIGNEE_NAME
- CA_ATTRKEY_PRIORITY

Example

```
Dim objPackageChooser
Dim objPackageList
objPackageChooser = objContext.GetPackageChooser()
Set objPackageList = objPackageChooser.GetPackageList()
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

Execute Method

Executes the PackageChooser based on the set of package selection criteria set in the package chooser attributes.

Syntax

object.Execute(*blnUseCurrentContext*)

Owning Class

CaWPackageChooser

Arguments

blnUseCurrentContext - A boolean, if True restrict package search to current context, if False use Project Id and State Id if set.

Return Value

A long - Positive if success, Zero if failure.

Example

```
Dim objPackageChooser
Dim objPackageList
objPackageChooser = objContext.GetPackageChooser()
lngRet = objPackageChooser.Execute(True)
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- ProjectObjId
- StateObjId

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWPackageChooser

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objPackageChooser
Set objPackageChooser = objContext.GetPackageChooser()
If objPackageChooser.IsValid() <> 0 Then
    MsgBox "This is a valid object", vbOKOnly, "Test PackageChooser"
End If
```

SetPackageName Method

Sets a name expression for search by name pattern.

Syntax

object.**SetPackageName** [*value*], *index*

Owning Class

CaWPackageChooser

Arguments

value - A string, name expression to search for package names.

index - A long value.

Example

```
Dim objPackageChooser
objPackageChooser = rContext.GetPackageChooser()
objPackageChooser.SetPackageName "QA: *", 0
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

CaWPromote

CaWPromote encapsulates the HSDK Promote process. It cannot be instantiated directly. The Promote process can be set in the CaWContext object either by name or object ID. The CaWPromote object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the Project, State, and Promote Context.
- Select a set of Change Packages to Promote using the CaPackageChooser object.
- Set the Promote attributes.
- Invoke the Execute function.

Methods

- SetMergedPkgsOnly
- SetCheckPackageDependencies
- SetCheckPackageGroupBind
- GetMergedPkgsOnly
- GetCheckPackageDependencies
- GetCheckPackageGroupBind
- Execute
- IsValid

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWPackageChooser

SetMergedPkgsOnly Method

Sets the Merged Packages Only flag.

Syntax

object.**SetMergedPkgsOnly**(*blnFlag*)

Owning Class

CaWPromote

Arguments

blnFlag - A boolean, if only merged packages can be promoted. If False packages with branched versions can be promoted.

Return Value

A boolean - True if set succeeded, False if failed.

Example

```
Dim objPromote
Set objPromote = objContext.GetPromote()
blnRet = objPromote.SetMergedPkgsOnly(True);
```

More information:

- CHSDK Programming Fundamentals
- GetMergedPkgsOnly

SetCheckPackageDependencies Method

Sets Check Package Dependencies flag.

Syntax

object.**SetCheckPackageDependencies**(*blnFlag*)

Owning Class

CaWPromote

Arguments

blnFlag - A boolean, True if verify dependencies prior to promote, False if not.

Return Value

A boolean - True if set succeeded, False if failed.

Example

```
Dim objPromote
Set objPromote = objContext.GetPromote()
blnRet = objPromote.SetCheckPackageDependencies(True);
```

More information:

- CHSDK Programming Fundamentals
- GetCheckPackageDependencies

SetCheckPackageGroupBind Method

Specifies whether or not packages in a package group must all be promoted together.

Syntax

object.**SetCheckPackageGroupBind**(*blnFlag*)

Owning Class

CaWPromote

Arguments

blnFlag - A boolean. True if yes, False if no.

Return Value

A boolean - True if set succeeded, False if failed.

Example

```
Dim objPromote
Set objPromote = objContext.GetPromote()
blnRet = objPromote.SetCheckPackageGroupBind(True);
```

More information:

- CHSDK Programming Fundamentals
- GetCheckPackageGroupBind

GetMergedPkgsOnly Method

Gets the current value of the Merged Packages Only flag.

Syntax

object.**GetMergedPkgsOnly**()

Owning Class

CaWPromote

Arguments

None

Return Value

A boolean - Current value of flag.

Example

```
Dim objPromote
Set objPromote = objContext.GetPromote()
blnRet = objPromote.GetMergedPkgsOnly();
```

More information:

- CHSDK Programming Fundamentals
- SetMergedPkgsOnly

GetCheckPackageDependencies Method

Gets the current value of the Check Package Dependencies flag.

Syntax

object.**GetCheckPackageDependencies()**

Owning Class

CaWPromote

Arguments

None

Return Value

A boolean - Current value of flag.

Example

```
Dim objPromote
Set objPromote = objContext.GetPromote()
blnRet = objPromote.GetCheckPackageDependencies();
```

More information:

- CHSDK Programming Fundamentals
- SetCheckPackageDependencies

GetCheckPackageGroupBind Method

Gets the current value of the Check Package Group Bind flag.

Syntax

object.GetCheckPackageGroupBind()

Owning Class

CaWPromote

Arguments

None

Return Value

A boolean - Current value of flag.

Example

```
Dim objPromote
Set objPromote = objContext.GetPromote()
bLnRet = objPromote.GetCheckPackageGroupBind();
```

More information:

- CHSDK Programming Fundamentals
- SetCheckPackageGroupBind

Execute Method

Executes the Promote process on the Package list currently selected using the Package Chooser.

Syntax

object.Execute()

Owning Class

CaWPromote

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Example

```
Dim objPromote
Set objPromote = objContext.GetPromote()
lngRet = objPromote.Execute();
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWPackageChooser

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWPromote

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objPromote
Set objPromote = objContext.GetPromote()
If objPromote.IsValid() <> 0 Then
    msgbox "This is a valid object", vbOKonly, "Test Promote"
End If
```

CaWRemoveItem

CaWRemoveItem encapsulates the HSDK RemoveItem process. It cannot be instantiated directly. The RemoveItem process can be set in the CaWContext object either by name or object ID. The CaWRemoveItem object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values. Set the item either by object ID or by name and path. If name is used then the recursive flag is relevant. This is a logical remove that creates a version with a “Delete” tag.

Usage

- Set the Project, State, and RemoveItem Context.
- Set the RemoveItem attributes, including the item to be removed.
- Invoke the Execute function.

Properties

- ItemObjId
- ItemName
- ItemPath
- Recursive

Methods

- SetDescription
- GetDescription
- Execute
- IsValid

More information:

- CHSDK Programming Fundamentals
- CaWContext

ItemObjId Property

Identifies the item to be removed by its object ID.

Syntax

object.ItemObjId = [value]

Owning Class

CaWRemoveItem

Arguments

value - A long that represents the Item Object Id.

Notes

Takes precedence over name. If present then name, path, and recursive are ignored.

Example

```
Dim objRemoveItem
Set objRemoveItem = objContext.GetRemoveItem()
objRemoveItem.ItemObjId = 12
lngItemObjId = objRemoveItem.ItemObjId
```

More information:

- CHSDK Programming Fundamentals
- ItemName

ItemName Property

Identifies the item to be removed by its name.

Syntax

object.**ItemName** = [*value*]

Owning Class

CaWRemoveItem

Arguments

value - A string that represents the Item Name.

Notes

The name is ignored if an item object ID is set. The name is used in conjunction with the path and recursive attributes.

Example

```
Dim objRemoveItem
Set objRemoveItem = objContext.GetRemoveItem()
objRemoveItem.ItemName = "Item1.txt"
strName = objRemoveItem.ItemName
```

More information:

- CHSDK Programming Fundamentals
- ItemObjId
- ItemPath
- Recursive

ItemPath Property

Sets the view path of the item to be removed.

Syntax

object.**ItemPath** = [*value*]

Owning Class

CaWRemoveItem

Arguments

value - A string that represents a valid view path.

Example

```
Dim objRemoveItem
Set objRemoveItem = objContext.GetRemoveItem()
objRemoveItem.ItemPath = "\rep\code"
strItemPath = objRemoveItem.ItemPath
```

More information:

- CHSDK Programming Fundamentals
- ItemName
- ItemObjId
- Recursive

Recursive Property

Specifies whether or not to look recursively below the current item path for the item.

Syntax

object.**Recursive** = [*value*]

Owning Class

CaWRemoveItem

Arguments

value - True if yes, False if no.

Notes

Used in conjunction with ItemName and ItemPath properties.

Example

```
Dim objRemoveItem
Set objRemoveItem = objContext.GetRemoveItem()
objRemoveItem.Recursive = False
blnRecursive = objRemoveItem.Recursive
```

More information:

- CHSDK Programming Fundamentals
- ItemName
- ItemObjId
- ItemPath

SetDescription Method

Sets a description to annotate why the item is being removed.

Syntax

object.**SetDescription**(*strDescription*)

Owning Class

CaWRemoveItem

Arguments

strDescription - A string that represents the explanation.

Return Value

A boolean - True if set succeeded, False if failed.

Example

```
Dim objRemoveItem
Set objRemoveItem = objContext.GetRemoveItem()
blnRet = objRemoveItem.SetDescription("Obsolete");
```

More information:

- CHSDK Programming Fundamentals
- GetDescription

GetDescription Method

Retrieves the current value of the description attribute.

Syntax

object.**GetDescription()**

Owning Class

CaWRemoveItem

Arguments

None

Return Value

A string - Current description value.

Example

```
Dim objRemoveItem
Set objRemoveItem = objContext.GetRemoveItem()
strDescription = objRemoveItem.GetDescription();
```

More information:

- CHSDK Programming Fundamentals
- setDescription

Execute Method

Executes the RemoveItem process on the Package list currently selected using the Package Chooser.

Syntax

object.**Execute()**

Owning Class

CaWRemoveItem

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Example

```
Dim objRemoveItem
Set objRemoveItem = objContext.GetRemoveItem()
lngRet = objRemoveItem.Execute();
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWPackageChooser

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWRemoveItem

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objRemoveItem
Set objRemoveItem = objContext.GetRemoveItem()
If objRemoveItem.IsValid() <> 0 Then
    MsgBox "This is a valid object", vbOKOnly, "Test RemoveItem"
End If
```

CaWSwitchPackage

CaWSwitchPackage encapsulates the CA Harvest SCM SwitchPackage process. It cannot be instantiated directly. The SwitchPackage process can be set in the CaWContext object either by name or object ID. The CaWSwitchPackage object can only be retrieved from the CaWContext after it has been set.

Usage

- Set the Project, State, and SwitchPackage Context.
- Set the SwitchPackage attributes, including the package to be switched and the target package name.
- Invoke the Execute function.

Note: If you need to select a subset of versions from the source package, these versions must be selected in to the version chooser. If the version list is empty, then *all* versions in the source package will be switched to the target package.

Public Methods

- Execute

More information:

- HSDK Programming Fundamentals
- CaWContext

Execute Method

Executes the SwitchPackage process. To select a subset of the versions from the source package, use the version chooser.

Syntax

object.Execute(*strSourcePackage*, *strTargetPackage*)

Owning Class

CaWSwitchPackage

Arguments

strSourcePackage-The source package for the switch package process.

strTargetPackage-The target package for the switch package process.

Return Value

A long - Zero if success. Nonzero if failure.

Notes

None

Example

```
CaWSwitchPackage& rSwitchPackage = rContext.GetSwitchPackage();CaString szName =  
rSwitchPackage. GetPackageName(true);
```

More information:

- HSDK Programming Fundamentals
- CaWContext

CaWSQL

CaWSQL encapsulates the HSDK SQL process. It cannot be instantiated directly. The CaWSQL object can only be retrieved from the CaWContext, but there is not other setup necessary. The CaWSQL object is instantiated when the CaContext is created.

Usage

- Set the SQL string value.
- Invoke the Execute function.

Methods

- SetSQLStatement
- GetSQLResult
- Execute

More information:

- CHSDK Programming Fundamentals
- CaWContext

SetSQLStatement Method

Sets the SQL text to execute in the CA Harvest SCM database.

Syntax

object.**SetSQLStatement**(*strSQL*)

Owning Class

CaWSQL

Arguments

strSQL - The SQL statement.

Return Value

None

Example

```
Dim objSQL
Set objSQL = objContext.GetSQL()
objSQL.SetSQLStatement("Select * from HARUSER")
```

GetSQLResult Method

Retrieves the results from the execution of the SQL statement.

Syntax

object.**GetSQLResult**(*strSQL*)

Owning Class

CaWSQL

Arguments

strSQL - The SQL statement.

Return Value

A CaWContainer object - Results

Example

```
Dim objSQL
Set objSQL = objContext.GetSQL()
objSQL.SetSQLStatement("Select * from HARUSER")
objSQL.Execute(); CaContainer sqldata = rSQL.GetSQLResult();
Set objContainer = objSQL.GetSQLResult();
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

Execute Method

Executes the SQL statement supplied by SetSQLStatement against the CA Harvest SCM database.

Syntax

object.Execute()

Owning Class

CaWSQL

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Example

```
Dim objSQL
Set objSQL = objContext.GetSQL()
objSQL.SetSQLStatement("Select * from HARUSER")
objSQL.Execute(); CaContainer sqldata = rSQL.GetSQLResult();
Set objContainer = objSQL.GetSQLResult();
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

CaWTakeSnapshot

CaWTakeSnapshot encapsulates the HSDK TakeSnapshot process. It cannot be instantiated directly. The TakeSnapshot process can be set in the CaWContext object either by name or object ID. The CaWTakeSnapshot object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the Project, State, and TakeSnapshot Context.

Three types of snapshots are possible. The first creates a snapshot view of the current contents of the view in selected state in the context. The second uses a passed in CaWTimeStamp to take a snapshot of the contents matching that time. The third combines selected packages from Package Chooser and the name of a reference snapshotview to create a snapshot where the contents of the packages are overlayed on the view of the reference snapshot.
- Invoke the Execute function with the appropriate arguments for the desired behavior.

Methods

- SetCanViewExternallyFlag
- GetCanViewExternallyFlag
- IsValid
- Execute

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWTimeStamp
- CaWPackageChooser

SetCanViewExternallyFlag Method

Sets the CanViewExternally flag true or false.

Syntax

object.SetCanViewExternallyFlag(*blnFlag*)

Owning Class

CaWTakeSnapshot

Arguments

blnFlag - A boolean, If true snapshot is visible to other projects, if false only current project has visibility.

Return Value

bool - Value that was set.

Notes

The CanViewExternally flag defaults to true.

Example

```
Dim objTakeSnapshot
Set objTakeSnapshot = objContext.GetTakeSnapshot()
objTakeSnapshot.SetCanViewExternallyFlag(True)
```

More information:

- CHSDK Programming Fundamentals
- GetCanViewExternallyFlag

GetCanViewExternallyFlag Method

Gets the CanViewExternally flag true or false.

Syntax

object.**GetCanViewExternallyFlag()**

Owning Class

CaWTakeSnapshot

Arguments

None

Return Value

A boolean - If true snapshot is visible to other projects, if false only current project has visibility.

Notes

The CanViewExternally flag defaults to true.

Example

```
Dim objTakeSnapshot
Set objTakeSnapshot = objContext.GetTakeSnapshot()
BlnFlag = objTakeSnapshot.GetCanViewExternallyFlag()
```

More information:

- CHSDK Programming Fundamentals
- SetCanViewExternallyFlag

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWTakeSnapshot

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objTakeSnapshot
Set objTakeSnapshot = objContext.GetTakeSnapshot()
If objTakeSnapshot.IsValid() <> 0 Then
    msgbox "This is a valid object", vbOKonly, "Test TakeSnapshot"
End If
```


Execute Method

Executes the TakeSnapshot process on one of the following:

- The State selected in the CaWContext.
- The State selected in the CaWContext at a given time.
- The packages selected in the package chooser overlaid on a previous Snapshot view

Syntax

object.**Execute**(*strViewName*, *objTimeStamp*, *strRefSnapshot*)

Owning Class

CaWTakeSnapshot

Arguments

strViewName - A string represents the view name

objTimeStamp - A CaWTimeStamp object or a null object

strRefSnapshot - A string represents the referenced snapshot or a null string

Return Value

A long - Zero if success. Nonzero if failure.

Example

```
Dim objTakeSnapshot
Set objTakeSnapshot = objContext.GetTakeSnapshot()
lngRet = objTakeSnapshot.Execute(strViewName, Nothing, "")
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWTimeStamp

CaWUDP

CaWUDP encapsulates the HSDK UDP process. It cannot be instantiated directly. The UDP process can be set in the CaWContext object either by name or object ID. The CaWUDP object can only be retrieved from the CaWContext after it has been set. The CA Harvest SCM administrator sets default attribute values.

Usage

- Set the Project, State, and UDP Context.
- Set the UDP attributes.
- Invoke the Execute function.

Properties

- UsePackageList
- UseVersionList
- UDPTYPE
- ProgramName
- Description
- UserCanModifyInput

Methods

- SetInput
- SetAddCmdLineArg
- GetAddCmdLineArg
- GetInput
- Execute

More information:

- CHSDK Programming Fundamentals
- CaWContext

UsePackageList Property

Gets or sets a boolean value for whether to pass the UDP process a Package List which would be expanded by the [PACKAGE] system variable.

Syntax

object.UsePackageList = [value]

Owning Class

CaWUDP

Arguments

value - A boolean that indicates using packages in package chooser or not.

Notes

- To pass a list of packages to a UDP process, this boolean is only part of the process.
- First, the actual UDP command must make use as an argument the system variable [PACKAGE]. The hserver will replace this variable name with the names of the packages sent by the client.
- Second, this boolean must be set before Execute() is called.
- Finally, the package chooser must be used to select the packages and executed. The contents of the package chooser must match the contents of the desired package list.

Example

```
Dim objUDP
Dim objPackageChooser
Set objUDP = objContext.GetUDP()
Set objPackageChooser = objContext.GetPackageChooser()
objPackageChooser.Clear()
objPackageChooser.SetPackageName(objContext.GetPackageName())
objPackageChooser.Execute()
objUDP.UsePackageList = True
objUDP.Execute()
```

More information:

- CHSDK Programming Fundamentals
- CaWPackageChooser

UseVersionList Property

Gets or sets a boolean value for whether to pass the UDP process a Version List which would be expanded by the [VERSION] system variable.

Syntax

object.UseVersionList = [value]

Owning Class

CaWUDP

Arguments

value - A boolean that indicates using versions in version chooser or not.

Notes

- To pass a list of versions to a UDP process, this boolean is only part of the process.
- First, the actual UDP command must make use as an argument the system variable [VERSION]. The hserver will replace this variable name with the names and version number of the versions sent by the client.
- Second, this boolean must be set before Execute() is called.
- Finally, the version chooser must be used to select the versions of items and executed. The contents of the version chooser must match the contents of the desired version list.

Example

```
Dim objUDP
Dim objVersionChooser
Set objUDP = objContext.GetUDP()
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.Clear()
objVersionChooser.SetItemName("*.cpp")
objVersionChooser.Execute()
objUDP.UseVersionList = True
objUDP.Execute()
```

More information:

- CHSDK Programming Fundamentals
- CaWVersionChooser

UDPType Property (Read Only)

Gets the string describing the type of UDP that this process defines.

Syntax

object.UDPType

Owning Class

CaWUDP

Return Value

A string description of UDP type.

Notes

There are two possible types of UDPs:

Client

Client side UDP execution (local machine)

Server

Server side UDP execution (CA Harvest SCM server)

Example

```
Dim objUDP
Set objUDP = objContext.GetUDP()
strUDPTYPE = objUDP.UDPTYPE
```

More information:

- CHSDK Programming Fundamentals

ProgramName Property (Read Only)

Gets the complete program name and argument for the UDP process specified by the CA Harvest SCM administrator.

Syntax

object.ProgramName

Owning Class

CaWUDP

Return Value

A string containing command and arguments for UDP process

Example

```
Dim objUDP
Set objUDP = objContext.GetUDP()
strProgramName = objUDP.ProgramName
```

More information:

- CHSDK Programming Fundamentals

Description Property (Read Only)

Returns the description for the UDP process set by the CA Harvest SCM administrator.

Syntax

object.Description

Owning Class

CaWUDP

Return Value

A string description for this process.

Example

```
Dim objUDP
Set objUDP = objContext.GetUDP()
strDescription = objUDP.Description
```

More information:

- CHSDK Programming Fundamentals

UserCanModifyInput Property (Read Only)

Returns the boolean value which specifies whether “Secure Input” was set for the UDP process by the CA Harvest SCM administrator.

Syntax

object.UserCanModifyInput

Owning Class

CaWUDP

Return Value

A boolean - Indicates whether input is user modifiable.

Notes

- true - Using SetInput(), the user can replace the default values for the string that is passed to stdin of the executed UDP.
- false - User cannot modify input, using SetInput() will fail.

Example

```
Dim objUDP
Set objUDP = objContext.GetUDP()
If objUDP.UserCanModifyInput Then
    objUDP.SetInput("Input String")
End If
```

More information:

- CHSDK Programming Fundamentals
- SetInput
- GetInput

SetInput Method

Sets the string that will be piped to stdin for the UDP process.

Syntax

object.SetInput(*strInput*)

Owning Class

CaWUDP

Arguments

strInput - The string to be passed to stdin.

Return Value

None

Notes

- Permission must be given by CA Harvest SCM Administer to allow user-modifiable input. See `GetUserCanModifyInput()` to determine if the administrator has secured input.
- `SetInput` replaces any input that was specified by the administrator for that UDP.

Example

```
Dim objUDP
Set objUDP = objContext.GetUDP()
If objUDP.UserCanModifyInput Then
    objUDP.SetInput("Input String")
End If
```

More information:

- CHSDK Programming Fundamentals
- `UserCanModifyInput`
- `GetInput`

GetInput Method

Returns the current string that will be passed to the stdin of the process on execution.

Syntax

object.**GetInput()**

Owning Class

CaWUDP

Arguments

None

Return Value

A string that will be passed to input of executing process.

Notes

- Permission must be given by CA Harvest SCM Administer to allow user-modifiable input. See `GetUserCanModifyInput()` to determine if the administrator has secured input.
- `GetInput` returns the default string specified by the CA Harvest SCM administrator if not yet modified by `SetInput()`; otherwise returns the current value for the input.
- Output from the process will be sent to the CaHarvest log.

Example

```
Dim objUDP
Set objUDP = objContext.GetUDP()
strInput = objUDP.GetInput()
```

More information:

- CHSDK Programming Fundamentals
- `SetInput`

SetAddCmdLineArg Method

Sets additional command line arguments to be passed to the UDP executing process.

Syntax

object.**SetAddCmdLineArg**(*strCmdLineArg*)

Owning Class

CaWUDP

Arguments

strCmdLineArg - A string containing the arguments value.

Return Value

None

Notes

- SetAddCmdLineArg appends the arguments specified to the entire UDP command. Any arguments specified by the administrator during the setup of the UDP are retained.
- Arguments that are specified are prepended with a white space character.

Example

```
Dim objUDP
Set objUDP = objContext.GetUDP()
objUDP.SetAddCmdLineArg("-v -h")
objUDP.Execute()
```

More information:

- CHSDK Programming Fundamentals
- GetAddCmdLineArg

GetAddCmdLineArg Method

Retrieves the additional command line arguments that will be passed to the executing UDP.

Syntax

object.GetAddCmdLineArg()

Owning Class

CaWUDP

Arguments

None

Return Value

A string of arguments.

Notes

- SetAddCmdLineArg appends the arguments specified to the entire UDP command. Any arguments specified by the administrator during the setup of the UDP are retained.
- Arguments that are specified are prepended with a white space character.

Example

```
Dim objUDP
Set objUDP = objContext.GetUDP()
strCmdArg = objUDP.GetAddCmdLineArg()
```

More information:

- CHSDK Programming Fundamentals
- SetAddCmdLineArg

Execute Method

Executes the UDP process.

Syntax

object.**Execute()**

Owning Class

CaWUDP

Arguments

None

Return Value

A long - Zero if success. Nonzero if failure.

Notes

- If using Version Lists or Package Lists, their appropriate chooser must be run `Execute()` before running `CaWUDP.Execute()`.
- Output from the process will be sent to the output log.

Example

```
Dim objUDP
Set objUDP = objContext.GetUDP()
objUDP.Execute()
```

More information:

- CHSDK Programming Fundamentals
- CaWHarvest
- SetLogName
- CaWPackageChooser
- CaWVersionChooser

CaWVersionChooser

CaWVersionChooser cannot be instantiated directly. The CaWVersionChooser object can only be retrieved from the CaWContext after the project and state have been correctly added to the context.

The Version Chooser provides a mechanism to select a set of versions based upon filtered criteria. Each Version Chooser attribute adds an additional restriction on the selected version set. The resulting version list is available to CaWDeleteVersion, CaWCheckin and CaWCheckout processes in the current context. A list of attributes from the selected versions can be retrieved in the form of a CaWContainer.

Usage

- Set the Project and State Context.
Note: If the State is a Snapshot State then set a Snapshot View in the current context. If the State is not a Snapshot State and there is no Working View then the Version Chooser is not active.
- Set the desired Version Chooser attributes.
- Invoke the Execute function.

Properties

- ParentPath
- ItemName
- Recursive
- BranchOption
- VersionOption
- VersionItemOption
- VersionStatusOption
- VersionDateOption
- FromDate
- ToDate

Methods

- Clear
- SetItemObjId
- SetPackageObjId
- GetVersionList
- Execute
- IsValid

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWContainer
- CaWCheckin
- CaWCheckout
- CaWDeleteVersion

ParentPath Property (Write Only)

Removes all selection criteria.

Syntax

object.ParentPath = [value]

Owning Class

CaWVersionChooser

Arguments

value - A string that represents the parent repository view path to begin search.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.ParentPath = "\repl\mydir"
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- Recursive

ItemName Property (Write Only)

Sets a name expression to search for versions.

Syntax

object.ItemName = [value]

Owning Class

CaWVersionChooser

Arguments

value - A name expression.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.ItemName = "ca*.h"
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

Recursive Property (Write Only)

Sets recursive flag to search below the parent path.

Syntax

object.Recursive = [*value*]

Owning Class

CaWVersionChooser

Arguments

value - True if recursive, False if no.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.Recursive = True
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- ItemName

BranchOption Property (Write Only)

Sets the Branch option to specify whether trunk versions, branch versions, or both are to be retrieved.

Syntax

object.**BranchOption** = [*value*]

Owning Class

CaWVersionChooser

Arguments

value - A long that represents the Branch Option.

Notes

The following are the allowable values for the Branch Option:

1

Only get trunk versions.

16

Only get branch versions.

256

Only get unmerged branches.

4096

Only get merged branches.

273

Get trunk and branched versions.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.BranchOption = BRANCH_FILTER_TRUNK_AND_BRANCH
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

VersionOption Property (Write Only)

Sets the Version Option to specify whether to get only the latest or multiple versions of the same item.

Syntax

object.**VersionOption** = [*value*]

Owning Class

CaWVersionChooser

Arguments

value - A long that represents the Version Option.

Notes

The following are the allowable values for the Version Option:

16

Get all versions of an item in the view.

256

Get only the latest version of an item in the view.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.VersionOption = VERSION_FILTER_LATEST_IN_VIEW
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

VersionItemOption Property (Write Only)

Sets the Version Item Option to specify whether to base versions of an item, modified versions of an item, or both.

Syntax

object.**VersionItemOption** = [*value*]

Owning Class

CaWVersionChooser

Arguments

value - A long that represents the Version Item Option.

Notes

The following are the allowable values for the Version Item Option. They are defined in an include file built into the project.

1

Get only modified versions of an item in the view.

16

Get only base versions of an item in the view.

17

Get both base and modified versions of an item in the view.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.VersionItemOption = VERSION_FILTER_ITEM_BOTH
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

VersionStatusOption Property (Write Only)

Sets the Version Status Option to specify what types of versions to retrieve.

Syntax

object.**VersionStatusOption** = [*value*]

Owning Class

CaWVersionChooser

Arguments

value - A long that represents the Version Status Option.

Notes

The following are the allowable values for the Version Status Option:

1

Get only versions with a Normal tag.

16

Get only versions with a Reserve tag.

256

Get only versions with a Merge tag.

4096

Get only versions with a Delete tag.

65536

Get only versions with a Frozen tag.

69905

Get versions with any tag.

69904

Get versions with any tag except a Normal tag.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.VersionStatusOption = VERSION_FILTER_ALL_TAG
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

VersionDateOption Property (Write Only)

Sets the Version Date Option to specify date ranges.

Syntax

object.**VersionDateOption** = [*value*]

Owning Class

CaWVersionChooser

Arguments

value - A long that represents the Version Date Option.

Notes

The following are the allowable values for the Version Date Option. They are defined in an include file built into the project.

0

Don't filter on dates.

1

Get only versions created between the from and to dates.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.VersionDateOption = VERSION_OPTION_DATE_BETWEEN
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- FromDate
- ToDate

FromDate Property (Write Only)

Sets the From Date value of the date range.

Syntax

object.**FromDate** = [*value*]

Owning Class

CaWVersionChooser

Arguments

value - A CaWTimeStamp object that represents the time stamp value.

Notes

Only used if the Version Date Option is set.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.FromDate = objTimeStamp
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWTimeStamp
- VersionDateOption
- ToDate

ToDate Property (Write Only)

Sets the To Date value of the date range.

Syntax

object.**ToDate** = [*value*]

Owning Class

CaWVersionChooser

Arguments

value - A CaWTimeStamp object that represents the time stamp value.

Notes

Only used if the Version Date Option is set.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.ToDate = objTimeStamp
```

More information:

- CHSDK Programming Fundamentals
- CaWContext
- CaWTimeStamp
- VersionDateOption
- FromDate

Clear Method

Removes all selection criteria.

Syntax

object.Clear()

Owning Class

CaWVersionChooser

Arguments

None

Return Value

None

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.Clear();
```

More information:

- CHSDK Programming Fundamentals
- CaContext

SetItemObjId Method

Sets one in a set of Item Object Ids whose versions will be searched for.

Syntax

object.**SetItemObjId**(*IngItemObjId*, *IngIndex*)

Owning Class

CaWVersionChooser

Arguments

IngItemObjId - A long that represents the Item Object Id.

IngIndex - A long, the set of ids is stored in an indexed array. Defaults to zero.

Return Value

None

Notes

The set of IDs is stored in an ordered array. The index value sets a specific object ID value location. Index locations must be set in numerical order.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.SetItemObjId(10024, 1);
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

SetPackageObjId Method

Sets one in a set of Package Object Ids whose versions will be searched for.

Syntax

object.**SetPackageObjId**(*IngPackageObjId*, *IngIndex*)

Owning Class

CaWVersionChooser

Arguments

IngPackageObjId - A long that represents the Package Id whose versions to search for.

IngIndex - A long, the set of ids is stored in an indexed array. Defaults to zero.

Return Value

None

Notes

The set of ids is stored in an ordered array. The index value sets a specific object ID value location. Index locations must be set in numerical order.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.SetPackageObjId(10024, 1);
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

GetVersionList Method

Retrieve a CaContainer object that contains the attributes for each CA Harvest SCM Version in the list.

Syntax

object.**GetVersionList()**

Owning Class

CaWVersionChooser

Arguments

None

Return Value

Returns a CaWContainer object, null object if error.

Notes

List attributes are stored with the following hash keys:

- CA_ATTRKEY_NAME
- CA_ATTRKEY_OBJID

- CA_ATTRKEY_CREATORID
- CA_ATTRKEY_CREATOR_NAME
- CA_ATTRKEY_CREATION_TIME
- CA_ATTRKEY_MODIFIERID
- CA_ATTRKEY_MODIFIER_NAME
- CA_ATTRKEY_MODIFIED_TIME
- CA_ATTRKEY_PACKAGEID
- CA_ATTRKEY_ITEMID
- CA_ATTRKEY_ITEM_TYPE
- CA_ATTRKEY_PARENT_VERSIONID
- CA_ATTRKEY_MERGED_VERSIONID
- CA_ATTRKEY_MAPPED_VERSION_NAME
- CA_ATTRKEY_VERSION_STATUS
- CA_ATTRKEY_PACKAGE_NAME
- CA_ATTRKEY_FULL_PATH_NAME
- CA_ATTRKEY_CLIENT_MACHINE
- CA_ATTRKEY_CLIENT_PATH
- CA_ATTRKEY_VERSIONDATASIZE

Example

```
Dim objVersionChooser
Dim objVersionList
Set objVersionChooser = objContext.GetVersionChooser()
Set objVersionList = objVersionChooser.GetVersionList ();
```

More information:

- CHSDK Programming Fundamentals
- CaWContainer

Execute Method

Executes the VersionChooser based on the set of version selection criteria set in the version chooser attributes.

Syntax

object.Execute()

Owning Class

CaWVersionChooser

Arguments

None

Return Value

A long - Positive if success, Zero if failure.

Example

```
Dim objVersionChooser
Set objVersionChooser = objContext.GetVersionChooser()
objVersionChooser.Execute();
```

More information:

- CHSDK Programming Fundamentals
- CaWContext

IsValid Method

Checks if the *object* is a valid object.

Syntax

object.IsValid()

Owning Class

CaWVersionChooser

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
Dim objVersionChooser:  
  Set objVersionChooser = objContext.GetRemoveItem()  
If objVersionChooser.IsValid() <> 0 Then  
  msgbox "This is a valid object", vbOKonly, "Test VersionChooser"  
End If
```


Chapter 6: COM Wrapper for HUtilities Class Descriptions

This section contains the following topics:

[HUtilities Programming Fundamentals](#) (see page 813)

[CaWContainer](#) (see page 813)

[CaWData](#) (see page 820)

[CaWDataArray](#) (see page 827)

[CaWStringList](#) (see page 830)

[CaWTimeStamp](#) (see page 837)

HUtilities Programming Fundamentals

HUtilities is a lower-level CA Harvest SCM module. It defines a set of cross-platform classes that encapsulate data and data containers used throughout CA Harvest SCM. A subset of that class set is exported as a set of encapsulation classes. The encapsulation removes the need for CHSDK users to include the Standard Template Library (STL) in CHSDK applications.

CaWContainer

CaWContainer is an STL hash-map of CaWDataArray objects. Strings are used as the hash-map key values. CaWData objects are stored as elements in the CaWDataArray. CaWData objects provide a type-safe container for LPCTSTR, CaWTimeStamp, and long data types.

Methods

- Clear
- Props
- GetKeyCount
- GetKeyList
- GetKeyElementCount
- IsKeyExist

- IsEmpty
- FindData
- FindData2

More information:

- CaWdataArray
- CaWData

Clear Method

Empties the container.

Syntax

object.Clear()

Owning Class

CaWContainer

Arguments

None

Return Value

None

Example

```
ObjContainer.Clear()
```

More information:

- HUtilities Programming Fundamentals

Props Method

Retrieves a referent to an object within the container. The contents of the object can be read or altered.

Syntax

object.Props(*strKey*, *lngIndex*)

Owning Class

CaWContainer

Arguments

strKey - A string that represents the key value of the CaWDataArray to locate.

lngIndex - A long that indicates the location within the CaWDataArray where the CaWData object resides.

Return Value

A CaWData object - A referent to the located object.

Example

```
Dim objData
Set objData = objContainer.Props("MyKey", 2);
```

More information:

- HUtilities Programming Fundamentals
- IsKeyExist
- GetKeyElementCount
- CaWData

GetKeyCount Method

Retrieves the quantity of hash-map keys currently in the container.

Syntax

object.**GetKeyCount()**

Owning Class

CaWContainer

Arguments

None

Return Value

A long - Number of keys.

Example

```
lngNumKeys = objContainer.GetKeyCount();
```

More information:

- HUtilities Programming Fundamentals
- IsKeyExist
- GetKeyElementCount

GetKeyList Method

Retrieves a list of the hash-map keys currently in the container.

Syntax

object.**GetKeyList()**

Owning Class

CaWContainer

Arguments

None

Return Value

A CaWStringList object - The key list.

Example

```
Dim objStringList
Set objStringList = objContainer.GetKeyList();
```

More information:

- HUtilities Programming Fundamentals
- CaWStringList

GetKeyElementCount Method

Retrieves the quantity of CaWData objects in the CaWDataArray indexed by the specified hash-map key value.

Syntax

object.**GetKeyElementCount(strKey)**

Owning Class

CaWContainer

Arguments

strKey - The key value to check.

Return Value

A long - Number of CaWData objects.

Example

```
lngNumElements = objContainer.GetKeyElementCount("MyKey");
```

More information:

- HUtilities Programming Fundamentals
- IsKeyExist
- CaWdataArray
- CaWData

IsKeyExist Method

Checks for the existence of the specified key value.

Syntax

object.**IsKeyExist**(*strKey*)

Owning Class

CaWContainer

Arguments

strKey - The key value to check.

Return Value

A boolean - True if it exists, False if not.

Example

```
blnHaveKey = objContainer.IsKeyExist("MyKey");
```

More information:

- HUtilities Programming Fundamentals
- Props

IsEmpty Method

Checks if the container is empty.

Syntax

object.IsEmpty()

Owning Class

CaWContainer

Arguments

None

Return Value

A boolean - True if it empty, False if not.

Example

```
blnHaveKey = objContainer.IsEmpty();
```

More information:

- HUtilities Programming Fundamentals

FindData Method

Searches all elements indexed by the specified key value for any whose value matches the supplied string. Returns a set of indexes to CaWData objects that match the values.

Syntax

object.FindData(*strKey*, *strValue*)

Owning Class

CaWContainer

Arguments

strKey - A string that indicates the Key value.

strValue - String to search for.

Return Value

A CaWdataArray object - array of matching indexes.

Example

```
Dim objdataArray  
Set objdataArray = objContainer.FindData(CA_ATTRKEY_NAME, strProjectName)
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaWdataArray
- CaWData

FindData2 Method

Searches all elements indexed by the specified key value for any whose value matches the supplied string. Returns a set of indexes to CaWData objects that match the values.

Syntax

object.**FindData2**(*strKey*, *lngValue*)

Owning Class

CaWContainer

Arguments

strKey - A string that indicates the Key value

lngValue - Long value to search for.

Return Value

A CaWDataArray object - Array of matching indexes.

Example

```
Dim objDataArray
objDataArray = objContainer.FindData2(CA_ATTRKEY_PRIORITY, 1)
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaWDataArray
- CaWData

CaWData

CaWData objects provide a type-safe container for LPCTSTR, CaTimeStamp, and long data types. They have no type until set. Once they are set the value can only be retrieved as that same type. Once set they can be overwritten by the same data type but no other.

Methods

- GetType
- IsNull
- SetNull
- InitHObject
- GetLongValue
- GetStringValue
- GetTimeStampValue
- SetLongValue
- SetStringValue
- SetTimeStampValue

More information:

- CaWTimeStamp

SetNull Method

Sets the Data object's value to NULL but does not change its type.

Syntax

object.SetNull()

Owning Class

CaWData

Arguments

None

Return Value

None

Example

```
objData.SetNull();
```

More information:

- HUtilities Programming Fundamentals
- CaTimeStamp
- IsNull

IsNull Method

Checks to see whether the value of the CaWData object is NULL.

Syntax

object.IsNull()

Owning Class

CaWData

Arguments

None

Return Value

A boolean - True if NULL, False if not.

Example

```
blnResult = objData.IsNull();
```

More information:

- HUtilities Programming Fundamentals
- CaTimeStamp
- SetNull

GetType Method

Retrieves the data type of the object.

Syntax

```
object.GetType()
```

Owning Class

CaWData

Arguments

None

Return Value

A long - Data type of object

Notes

Data type can have the following values:

0

Value has not been set.

1

Long value.

2

String value.

4

TimeStamp object accessed as CaWTimeStamp.

Example

```
lngType = objData.GetType();
```

More information:

- HUtilities Programming Fundamentals
- CaTimeStamp

InitHObject Method

Initialize encapsulated CA Harvest SCM object.

Syntax

```
object.InitHObject()
```

Owning Class

CaWData

Arguments

None

Return Value

A boolean - True if succeeded, False if failed.

Example

```
Dim objData  
Set objData = CreateObject("Chsdk.CaWData")  
blnRet = objData.InitHObject()
```

More information:

- HUtilities Programming Fundamentals

SetLongValue Method

Sets the data value to the specified long value.

Syntax

```
object.SetLongValue(lngValue)
```

Owning Class

CaWData

Arguments

lngValue - A long.

Return Value

None

More information:

- HUtilities Programming Fundamentals

SetStringValue Method

Sets the data value to the specified string value.

Syntax

```
object.SetStringValue(strValue)
```

Owning Class

CaWData

Arguments

strValue - A string.

Return Value

None

More information:

- HUtilities Programming Fundamentals

SetTimeStampValue Method

Sets the data value to the specified time stamp value.

Syntax

object.**SetTimeStampValue**(*objTimeStamp*)

Owning Class

CaWData

Arguments

objTimeStamp - A CaWTimeStamp object.

Return Value

None

More information:

- HUtilities Programming Fundamentals
- CaWTimeStamp

GetLongValue Method

Retrieves the value of the CaWData object as a long object.

Syntax

object.**GetLongValue**()

Owning Class

CaWData

Arguments

None

Return Value

A long.

Notes

Errors if CaWData value is not of type long.

More information:

- HUtilities Programming Fundamentals

GetStringValue Method

Retrieves the value of the CaWData object as a string object.

Syntax

object.GetStringValue()

Owning Class

CaWData

Arguments

None

Return Value

A string.

Notes

Errors if CaWData value is not of type STRING.

More information:

- HUtilities Programming Fundamentals

GetTimeStampValue Method

Retrieves the value of the CaWData object as timestamp object.

Syntax

object.GetTimeStampValue()

Owning Class

CaWData

Arguments

None

Return Value

A CaWTimeStamp object

Notes

Errors if CaWData value is not of type TIMESTAMP.

More information:

- HUtilities Programming Fundamentals
- CaWTimeStamp

CaWdataArray

CaWdataArray is an STL vector of CaWData objects. CaWData objects are stored as elements in the CaWdataArray. CaWData objects provide a type-safe dataArray for LPCTSTR, CaTimeStamp, and long data types.

Methods

- Clear
- GetSize
- Find
- GetAt

More information:

- CaData

Clear Method

Empties the dataArray.

Syntax

object.Clear()

Owning Class

CaWdataArray

Arguments

None

Return Value

None

Example

```
objdataArray.clear();
```

More information:

- HUtilities Programming Fundamentals

GetSize Method

Retrieves the quantity of CaWData objects in the array.

Syntax

object.GetSize()

Owning Class

CaWdataArray

Arguments

None

Return Value

A long - Number of objects in array.

Example

```
lngSize = objDataArray.GetSize();
```

More information:

- HUtilities Programming Fundamentals

Find Method

Searches all elements for one whose value matches the supplied object. Returns the array index of the found object in an argument.

Syntax

object.Find(*objData*, *lngIndex*)

Owning Class

CaWdataArray

Arguments

objData - A CaWData object to search for.

lngIndex - A long that indicates the location to place array index if found.

Return Value

A boolean - True if found, False if not.

Notes

If not found then lngIndex is not changed.

Example

```
Dim lngIndex  
ObjDataArray.Find(objData, lngIndex)
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaWData

GetAt Method

Retrieves the indexed CaWData object.

Syntax

object.**GetAt**(*lngIndex*)

Owning Class

CaWdataArray

Arguments

lngIndex - index value.

Return Value

A CaWData object - Indexed object.

Example

```
Dim objData  
Set objData = objDataArray.GetAt(0).
```

More information:

- HUtilities Programming Fundamentals
- Props
- CaWdataArray
- CaWData

CaWStringList

The CaWStringList class is a container class to hold strings.

Methods

- Add
- Find
- Remove
- Clean
- GetSize
- IsEmpty
- GetAllNames
- operator bool()
- GetAt
- InitHObject
- IsValid

More information:

- HUtilities Programming Fundamentals

Add Method

Adds the string to the list.

Syntax

object.**Add**(*strString*)

Owning Class

CaWStringList

Arguments

strString - string to add.

Return Value

A boolean - True if success, False if fails.

Notes

If string is not valid then it fails.

Duplicates are not added but function still returns true.

Example

```
blnResult = objStringList.Add("My String");
```

More information:

- HUtilities Programming Fundamentals

Find Method

Search for the specified string in the list.

Syntax

object.**Find**(*strString*)

Owning Class

CaWStringList

Arguments

strString - String to search for.

Return Value

A string - Reference to found string.

Notes

Returns NULL string if not found.

Example

```
strString = objStringList.Find("My String");
```

More information:

- HUtilities Programming Fundamentals

Remove Method

Find matching string in list and remove it.

Syntax

object.**Remove**(*strString*)

Owning Class

CaWStringList

Arguments

strString - String to remove.

Return Value

A boolean - True if success, False if fails.

Notes

Fails if string is not valid or not found in list.

Example

```
strString = objStringList.Remove("My String");
```

More information:

- HUtilities Programming Fundamentals

Add Method

Adds the string to the list.

Syntax

object.**Add**(*strString*)

Owning Class

CaWStringList

Arguments

strString - String to add.

Return Value

A boolean - True if success, False if fails.

Notes

If string is not valid then it fails.

Duplicates are not added but function still returns true.

Example

```
blnResult = objStringList.Add("My String");
```

More information:

- HUtilities Programming Fundamentals

IsEmpty Method

Determines if list is empty.

Syntax

```
object.IsEmpty()
```

Owning Class

CaWStringList

Arguments

None

Return Value

A boolean - True if NULL or empty, False if length > 0.

Example

```
blnResult = objStringList.IsEmpty();
```

More information:

- HUtilities Programming Fundamentals

Clean Method

Removes all elements from the list.

Syntax

object.Clean()

Owning Class

CaWStringList

Arguments

None

Return Value

None

Example

```
objStringList.Clean();
```

More information:

- HUtilities Programming Fundamentals

GetSize Method

Returns the number of elements in this list.

Syntax

object.GetSize()

Owning Class

CaWStringList

Arguments

None

Return Value

A long - Number of strings in list.

Example

```
lngLength = objStringList.GetSize();
```

More information:

- HUtilities Programming Fundamentals

GetAllNames Method

Concatenates all strings in the list into a new string separated by “\n\t”.

Syntax

object.**GetAllName**()

Owning Class

CaWStringList

Arguments

None

Return Value

A string - Rference to this string.

Example

```
strString = objStringList.GetAllNames();
```

More information:

- HUtilities Programming Fundamentals

GetAt Method

Retrieves the string from the list at the specified index.

Syntax

object.**GetAt**(*lngIndex*)

Owning Class

CaWStringList

Arguments

lngIndex - A long that represents the index into list.

Return Value

A string - Rference to the string found at the index

Notes

If index is out of range then a NULL string reference is returned.

Example

```
strString = objStringList.GetAt(0);
```

More information:

- HUtilities Programming Fundamentals

IsValid Method

Checks if the *object* is a valid object.

Syntax

```
object.IsValid()
```

Owning Class

CaWStringList

Arguments

None

Return Value

A boolean - True if the *object* is a valid object otherwise False.

Example

```
If objStringList.IsValid() <> 0 Then  
    msgbox "This is a valid object", vbOKonly, "Test StringList"  
End If
```

InitHObject Method

Initialize encapsulated CA Harvest SCM object.

Syntax

```
object.InitHObject()
```

Owning Class

CaWStringList

Arguments

None

Return Value

A boolean - True if succeeded, False if failed.

Example

```
Dim objStringList
Set objStringList = CreateObject("Chsdk.CaWStringList")
blnRet = objStringList.InitHObject()
```

More information:

- HUtilities Programming Fundamentals

CaWTimeStamp

CaWTimeStamp objects contain date and time information.

Methods

- GetYear
- GetMonth
- GetDay
- GetHour
- GetMinute
- GetSecond
- GetFraction
- SetYear
- SetMonth
- SetDay
- SetHour
- SetMinute
- SetSecond
- SetFraction
- InitHObject
- ChangeTime
- ConvertGMTTime

SetYear Method

Sets Year value.

Syntax

object.**SetYear**(*IngYear*)

Owning Class

CaWTimeStamp

Arguments

IngYear - Along that indicates the Year value.

Return Value

A CaWTimeStamp object - Encapsulates a reference to CaTimeStamp object so functions can be chained together.

Example

```
objTimeStamp.SetYear(2001);
```

More information:

- HUtilities Programming Fundamentals
- GetYear

SetMonth Method

Sets Month value.

Syntax

object.**SetMonth**(*IngMonth*)

Owning Class

CaWTimeStamp

Arguments

IngMonth - Month value.

Return Value

A CaWTimeStamp object - Encapsulates a reference to CaTimeStamp object so functions can be chained together.

Example

```
objTimeStamp.SetMonth(9);
```

More information:

- HUtilities Programming Fundamentals
- GetMonth

SetDay Method

Sets Day value.

Syntax

object.**SetDay**(*IngDay*)

Owning Class

CaWTimeStamp

Arguments

IngDay - Day value.

Return Value

A CaWTimeStamp object - Encapsulates a reference to CaTimeStamp object so functions can be chained together.

Example

```
objTimeStamp.SetDay(11);
```

More information:

- HUtilities Programming Fundamentals
- GetDay

SetHour Method

Sets Hour value.

Syntax

object.**SetHour**(*IngHour*)

Owning Class

CaWTimeStamp

Arguments

IngHour - Hour value.

Return Value

A CaWTimeStamp object - encapsulates a reference to CaTimeStamp object so functions can be chained together.

Example

```
objTimeStamp.SetHour(8);
```

More information:

- HUtilities Programming Fundamentals
- GetHour

SetMinute Method

Sets Minute value.

Syntax

object.**SetMinute**(*IngMinute*)

Owning Class

CaWTimeStamp

Arguments

IngMinute - Minute value.

Return Value

A CaWTimeStamp object - Encapsulates a reference to CaTimeStamp object so functions can be chained together.

Example

```
objTimeStamp.SetMinute(45);
```

More information:

- HUtilities Programming Fundamentals
- GetMinute

SetSecond Method

Sets Second value.

Syntax

object.**SetSecond**(*IngSecond*)

Owning Class

CaWTimeStamp

Arguments

IngSecond - Second value.

Return Value

A CaWTimeStamp object - Encapsulates a reference to CaTimeStamp object so functions can be chained together.

Example

```
objTimeStamp.SetSecond(0);
```

More information:

- HUtilities Programming Fundamentals
- GetSecond

SetFraction Method

Sets Fraction value.

Syntax

object.**SetFraction**(*IngFraction*)

Owning Class

CaWTimeStamp

Arguments

IngFraction - Fraction value.

Return Value

A CaWTimeStamp object - Encapsulates a reference to CaTimeStamp object so functions can be chained together.

Example

```
objTimeStamp.SetFraction(0);
```

More information:

- HUtilities Programming Fundamentals
- GetFraction

GetYear Method

Gets Year value.

Syntax

object.**GetYear**()

Owning Class

CaWTimeStamp

Arguments

None

Return Value

A long - Year value of TimeStamp.

Example

```
lngYear = objTimeStamp.GetYear();
```

More information:

- HUtilities Programming Fundamentals
- SetYear

GetMonth Method

Gets Month value.

Syntax

```
object.GetMonth()
```

Owning Class

CaWTimeStamp

Arguments

None

Return Value

A long - Month value of TimeStamp.

Example

```
lngMonth = objTimeStamp.GetMonth();
```

More information:

- HUtilities Programming Fundamentals
- SetMonth

GetDay Method

Gets Day value.

Syntax

object.**GetDay()**

Owning Class

CaWTimeStamp

Arguments

None

Return Value

A long - Day value of TimeStamp.

Example

```
lngDay = objTimeStamp.GetDay();
```

More information:

- HUtilities Programming Fundamentals
- SetDay

GetHour Method

Gets Hour value.

Syntax

object.**GetHour()**

Owning Class

CaWTimeStamp

Arguments

None

Return Value

A long - Hour value of TimeStamp.

Example

```
lngHour = objTimeStamp.GetHour();
```

More information:

- HUtilities Programming Fundamentals
- SetHour

GetMinute Method

Gets Minute value.

Syntax

```
object.GetMinute()
```

Owning Class

CaWTimeStamp

Arguments

None

Return Value

A long - Minute value of TimeStamp.

Example

```
lngMinute = objTimeStamp.GetMinute();
```

More information:

- HUtilities Programming Fundamentals
- SetMinute

GetSecond Method

Gets Second value.

Syntax

object.**GetSecond()**

Owning Class

CaWTimeStamp

Arguments

None

Return Value

A long - Second value of TimeStamp.

Example

```
lngSecond = objTimeStamp.GetSecond();
```

More information:

- HUtilities Programming Fundamentals
- SetSecond

GetFraction Method

Gets Fraction value.

Syntax

object.**GetFraction()**

Owning Class

CaWTimeStamp

Arguments

None

Return Value

A long - Fraction value of TimeStamp.

Example

```
lngFraction = objTimeStamp.GetFraction();
```

More information:

- HUtilities Programming Fundamentals
- SetFraction

ChangeTime Method

Converts time to specified timezone.

Syntax

```
object.ChangeTime(lngTimezone)
```

Owning Class

CaWTimeStamp

Arguments

HINT32 iTimezone - Difference in hours between new time and GMT time.

Return Value

A CaWTimeStamp object - Reference to this TimeStamp.

Notes

Adjusts month and day as well.

Example

```
objTimeStamp.ChangeTime(3);
```

More information:

- HUtilities Programming Fundamentals

ConvertGMTTime Method

Converts time to specified timezone.

Syntax

object.ConvertGMTTime()

Owning Class

CaWTimeStamp

Arguments

None

Return Value

A CaWTimeStamp object - Reference to this TimeStamp.

Example

```
objTimeStamp.ConvertGMTTime();
```

More information:

- HUtilities Programming Fundamentals

InitHObject Method

Initializes encapsulated CA Harvest SCM object.

Syntax

object.InitHObject()

Owning Class

CaWTimeStamp

Arguments

None

Return Value

A boolean - True if succeeded, False if failed.

Example

```
Dim objTimeStamp:  
Set objTimeStamp = CreateObject("Chsdk.CaWTimeStamp")  
blnRet = objTimeStamp.InitHObject()
```

More information:

- HUtilities Programming Fundamentals