

# AllFusion<sup>™</sup> Harvest Change Manager

## SDK Reference Guide

5.11



Computer Associates<sup>™</sup>

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been 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 the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc. (CA)

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Contents

## Chapter 1: Using the Harvest Software Development Kit

Overview .....	1-1
Operating System Requirements .....	1-1
Obsolete Versions .....	1-2
Compatibility With Earlier Versions .....	1-2
HSDK Programming Fundamentals .....	1-2
Usage .....	1-2
Installation .....	1-3
Variable Portability Type Definitions .....	1-5
HSDK Context .....	1-5
Object Lists .....	1-5
Object Security .....	1-5
Constructors .....	1-6
HSDK Class Descriptions .....	1-6
HSDK Class Hierarchy .....	1-6
Additional Class Dependencies .....	1-6
HSDK Sample Program .....	1-7

## Chapter 2: HSDK Class Descriptions

CaApprove .....	2-1
Void SetApprove(bool bFlag) .....	2-2
Void SetDescription(LPCTSTR szDescription) .....	2-2
bool GetApprove(void) .....	2-3
LPCTSTR GetDescription(void) .....	2-3
HINT32 Execute(void) .....	2-3
CaCheckin .....	2-4
bool SetCheckinMode(HINT32 iMode) .....	2-6
bool SetPathOption(HINT32 iOption) .....	2-7
bool SetItemOption(HINT32 iOption) .....	2-8
bool SetClientDir(LPCTSTR szCDir) .....	2-8

---

bool SetViewPath(LPCTSTR szVPath) .....	2-9
bool SetDeleteAfterCI(bool bFlag) .....	2-9
bool SetDescription(LPCTSTR szDesc) .....	2-9
void SetRemoteMachineName(LPCTSTR szRemoteMachineName) .....	2-10
void SetRemoteUserName(LPCTSTR szRemoteUserName) .....	2-10
void SetRemotePassword(LPCTSTR szRemotePassword) .....	2-11
void SetFileList(CaStringList &FileList) .....	2-11
HINT32 GetItemOption(void) .....	2-12
LPCTSTR GetDescription(void) .....	2-12
HINT32 GetCheckinMode(void) .....	2-13
HINT32 GetPathOption(void) .....	2-13
LPCTSTR GetClientDir(void) .....	2-14
LPCTSTR GetViewPath(void) .....	2-14
bool GetDeleteAfterCI(void) .....	2-15
LPCTSTR GetRemoteMachineName(void) .....	2-15
LPCTSTR GetRemoteUserName(void) .....	2-16
LPCTSTR GetRemotePassword(void) .....	2-16
CaStringList GetFileList(void) .....	2-17
HINT32 CreateItemPath( LPCTSTR szPathName, LPCTSTR szParentPath) .....	2-17
void SetCallBack( PFN_CACICB pfCallBack ) .....	2-18
void SetCBUserData( void* pUserData ) .....	2-19
PFN_CACICB GetCallBack(void); .....	2-19
void* GetCBUserData(void); .....	2-20
HINT32 Execute(bool bFileList) .....	2-20
CaCheckout .....	2-21
bool SetCheckoutMode(HINT32 iMode) .....	2-22
bool SetPathOption(HINT32 iOption) .....	2-23
bool SetReplaceFile(bool bFlag) .....	2-24
bool SetClientDir(LPCTSTR szCDir) .....	2-24
bool SetViewPath(LPCTSTR szVPath) .....	2-25
bool SetShareWorkDir(bool bFlag) .....	2-25
bool SetUseCITimeStamp(bool bFlag) .....	2-26
void SetRemoteMachineName(LPCTSTR szRemoteMachineName) .....	2-26
void SetRemoteUserName(LPCTSTR szRemoteUserName) .....	2-26
void SetRemotePassword(LPCTSTR szRemotePassword) .....	2-27
bool SetMVSFileTypes(HINT32 iOption) .....	2-27
HINT32 GetCheckoutMode(void) .....	2-28
HINT32 GetPathOption(void) .....	2-28
LPCTSTR GetClientDir(void) .....	2-29
LPCTSTR GetViewPath(void) .....	2-29
bool GetReplaceFile(void) .....	2-30
bool GetShareWorkDir(void) .....	2-30

---

bool GetUseCITimeStamp(void) .....	2-30
LPCTSTR GetRemoteMachineName(void) .....	2-31
LPCTSTR GetRemoteUserName(void) .....	2-31
LPCTSTR GetRemotePassword(void) .....	2-32
HINT32 GetMVSFileTypes(void) .....	2-32
void SetCallBack( PFN_CACOCB pfCallBack ) .....	2-33
void SetCBUserData( void* pUserData ) .....	2-34
PFN_CACOCB GetCallBack(void); .....	2-34
void* GetCBUserData(void); .....	2-35
HINT32 Execute(void) .....	2-35
CaConcurrentMerge .....	2-36
void SetMergeRule( LPCTSTR szRule ) .....	2-37
void SetVersionDescription( LPCTSTR szDescription ) .....	2-38
LPCTSTR GetMergeRule(void) .....	2-38
LPCTSTR GetVersionDescription(void) .....	2-39
HINT32 Execute(void) .....	2-39
CaContext .....	2-40
LPCTSTR GetProjectName(void) const .....	2-43
HAROBJID GetProjectId(void) const .....	2-43
LPCTSTR GetStateName(void) const .....	2-43
HAROBJID GetStateId(void) const .....	2-44
LPCTSTR GetPackageName(void) const .....	2-44
HAROBJID GetPackageId(void) const .....	2-44
HAROBJID GetViewName(void) const .....	2-45
LPCTSTR GetViewName(void) const .....	2-45
LPCTSTR GetSnapshotViewName(void) const .....	2-45
HAROBJID GetSnapshotViewId(void) const .....	2-46
LPCTSTR GetCheckoutName(void) const .....	2-46
LPCTSTR GetCheckinName(void) const .....	2-46
LPCTSTR GetPromoteName(void) const .....	2-47
LPCTSTR GetDemoteName(void) const .....	2-47
LPCTSTR GetApproveName(void) const .....	2-47
LPCTSTR GetCreatePackageName(void) const .....	2-48
LPCTSTR GetDeleteVersionName(void) const .....	2-48
LPCTSTR GetRemoveItemName(void) const .....	2-48
LPCTSTR GetConcurrentMergeName(void) const .....	2-49
LPCTSTR GetCrossProjectMergeName(void) const .....	2-49
LPCTSTR GetTakeSnapshotName(void) const .....	2-49
LPCTSTR GetUDPName(void) const .....	2-50
LPCTSTR GetUserName(void) const .....	2-50
LPCTSTR GetWebserverName(void) .....	2-50
HAROBJID GetUserId(void) const .....	2-51

---

---

LPCTSTR GetClientPath(void) const .....	2-51
LPCTSTR GetViewPath(void) const .....	2-51
CaCheckout& GetCheckout(void) const .....	2-52
CaCheckin& GetCheckin(void) const .....	2-52
CaPromote& GetPromote(void) const .....	2-53
CaDemote& GetDemote(void) const .....	2-53
CaApprove& GetApprove(void) const .....	2-53
CaCreatePackage& GetCreatePackage(void) const .....	2-54
CaDeleteVersion& GetDeleteVersion(void) const .....	2-54
CaRemoveItem& GetRemoveItem(void) const .....	2-55
CaConcurrentMerge& GetConcurrentMerge(void) const .....	2-55
CaCrossProjectMerge& GetCrossProjectMerge(void) const .....	2-55
CaTakeSnapshot& GetTakeSnapshot(void) const .....	2-56
CaUDP& GetUDP(void) const .....	2-56
CaSQL& GetSQL(void) const .....	2-57
CaPackageChooser& GetPackageChooser(void) const .....	2-57
CaVersionChooser& GetVersionChooser(void) const .....	2-57
CaHarvest& GetHarvestSession (void) const .....	2-58
CaContainer GetStateList(void) const .....	2-58
CaContainer GetViewList(void) const .....	2-59
CaContainer GetProcessList(void) const .....	2-59
bool SetProject(LPCTSTR szProjectName) .....	2-60
bool SetProject(HAROBJID iObjId) .....	2-60
bool SetState(LPCTSTR szStateName) .....	2-60
bool SetState(HAROBJID iObjId) .....	2-61
bool SetCheckout(LPCTSTR szCheckoutName) .....	2-61
bool SetCheckout(HAROBJID iObjId) .....	2-61
bool SetCheckin(LPCTSTR szCheckinName) .....	2-62
bool SetCheckin(HAROBJID iObjId) .....	2-62
bool SetPromote(LPCTSTR szPromoteName) .....	2-62
bool SetPromote(HAROBJID iObjId) .....	2-63
bool SetDemote(LPCTSTR szDemoteName) .....	2-63
bool SetDemote(HAROBJID iObjId) .....	2-64
bool SetCrossProjectMerge(LPCTSTR szCrossProjectMergeName) .....	2-64
bool SetCrossProjectMerge(HAROBJID iObjId) .....	2-64
bool SetTakeSnapshot(LPCTSTR szTakeSnapshotName) .....	2-65
bool SetTakeSnapshot(HAROBJID iObjId) .....	2-65
bool SetUDP(LPCTSTR szUDPName) .....	2-65
bool SetUDP(HAROBJID iObjId) .....	2-66
bool SetApprove(LPCTSTR szApproveName) .....	2-66
bool SetApprove(HAROBJID iObjId) .....	2-66
void SetWebserverName( LPCTSTR szWebserverName ) .....	2-67

---

bool SetCreatePackage(LPCTSTR szCreatePackageName) .....	2-67
bool SetCreatePackage(HAROBJID iObjId) .....	2-67
bool SetDeleteVersion(LPCTSTR szDeleteVersionName) .....	2-68
bool SetDeleteVersion(HAROBJID iObjId) .....	2-68
bool SetRemoveItem(LPCTSTR szRemoveItemName) .....	2-68
bool SetRemoveItem(HAROBJID iObjId) .....	2-69
bool SetConcurrentMerge(LPCTSTR szConcurrentMergeName) .....	2-69
bool SetConcurrentMerge(HAROBJID iObjId) .....	2-70
bool SetPackage(LPCTSTR szPackageName) .....	2-70
bool SetPackage(HAROBJID iObjId) .....	2-70
bool SetSnapshotView(LPCTSTR szSnapshotViewName) .....	2-71
bool SetSnapshotView( HAROBJID iObjid ) .....	2-71
void SetClientPath(LPCTSTR szClientPath) .....	2-71
void SetViewPath(LPCTSTR szViewPath) .....	2-72
bool IsPackageSet() .....	2-72
CaCreatePackage .....	2-73
void SetPackageName(LPCTSTR szPackageName) .....	2-74
void SetAssignedUserName( LPCTSTR szAssignedUserName) .....	2-74
void SetDescription(LPCTSTR szDescription) .....	2-75
void SetPriority(HINT32 iPriority) .....	2-75
LPCTSTR GetDefaultName (void) .....	2-76
LPCTSTR GetPackageName(void) .....	2-76
LPCTSTR GetAssignedUserName(void) .....	2-77
LPCTSTR GetDescription(void) .....	2-77
HINT32 GetPriority(void) .....	2-78
HINT32 Execute(void) .....	2-78
CaCrossProjectMerge .....	2-79
void SetFromState(LPCTSTR szProjectName, LPCTSTR szStateName) .....	2-79
void SetFromState(HAROBJID szProjectId, HAROBJID szStateId) .....	2-80
void SetMergeRule(LPCTSTR szRule) .....	2-81
void SetPlacement(LPCTSTR szPlacement) .....	2-81
void SetVersionDescription(LPCTSTR szDescription) .....	2-82
HAROBJID GetFromProjectId(void) .....	2-82
LPCTSTR GetFromProjectName(void) .....	2-83
HAROBJID GetFromStateId(void) .....	2-83
LPCTSTR GetFromStateName(void) .....	2-84
LPCTSTR GetMergeRule(void) .....	2-84
LPCTSTR GetPlacement(void) .....	2-85
LPCTSTR GetVersionDescription(void) .....	2-85
HINT32 Execute(void) .....	2-86
CaDeleteVersion .....	2-87
HINT32 Execute(void) .....	2-87

---

---

CaDemote .....	2-88
bool SetCheckPackageDependencies(bool bCheckPackageDependencies) .....	2-88
bool SetCheckPackageGroupBind( bool bFlag) .....	2-89
bool GetCheckPackageDependencies (void) .....	2-89
bool GetCheckPackageGroupBind (void) .....	2-90
HINT32 Execute(void) .....	2-90
CaException .....	2-91
CaException(bool bKey, LPCTSTR szMessage ) .....	2-92
CaException( const CaException &CRight) .....	2-92
CaException(LPCTSTR szKey, ...) .....	2-93
void SetReturnCode(HINT32 iReturnCode) .....	2-93
HINT32 GetReturnCode(void) .....	2-94
CaString GetMessage(void) .....	2-94
void LogMessage(void) .....	2-94
CaHarvest .....	2-95
CaHarvest(void) .....	2-96
CaHarvest(LPCTSTR szBrokerName) .....	2-96
HINT32 Login(LPCTSTR szBrokerName, LPCTSTR szUserName, LPCTSTR szPassword) .....	2-97
HINT32 Login(LPCTSTR szUserName, LPCTSTR szPassword) .....	2-97
LPCTSTR GetLastMessage( ) .....	2-98
HINT32 ChangePassword(LPCTSTR szPassword) .....	2-98
void Logout(void) .....	2-99
LPCTSTR GetBrokerName(void) const .....	2-99
CaContext& GetContext(void) const .....	2-100
CaContainer GetProjectList(void) const .....	2-100
CaContainer GetActiveProjectList(void) const .....	2-101
CaContainer GetInactiveProjectList(void) const .....	2-101
CaContainer GetStateList( HAROBJID iProject ) const .....	2-102
CaContainer GetUserList(void) const .....	2-102
bool SetLog(ostream &rLogStream) .....	2-103
static void SetStaticLog(ostream &rLogStream) .....	2-103
static ostream& GetLog(void) .....	2-103
CaPackageChooser .....	2-105
void Clear(void) .....	2-106
void SetPackageName(LPCTSTR szName, int iIndex = 0) .....	2-106
void SetPackageObjId(HAROBJID iObjId, int iIndex = 0) .....	2-107
void SetProjectObjId(HAROBJID iObjId) .....	2-107
void SetStateObjId (HAROBJID iObjId) .....	2-108
void SetPkgGroupObjId(HAROBJID iObjId) .....	2-108
void SetAssigneeObjId (HAROBJID iObjId) .....	2-108
void SetFromDate(CaTimeStamp& Time) .....	2-109
void SetToDate(CaTimeStamp& Time) .....	2-109

---

void SetCreatorObjId (HAROBJID iObjId) .....	2-110
void SetModifierObjId (HAROBJID iObjId) .....	2-110
void SetApproveUserObjId (HAROBJID iObjId) .....	2-110
void SetPriority(HINT32 iPriority) .....	2-111
void SetSubString(LPCTSTR szString) .....	2-111
void SetPkgStatus(HINT32 iPkgStatus) .....	2-111
void SetPkgEntryFromDate(CaTimeStamp& Time) .....	2-112
void SetPkgEntryToDate(CaTimeStamp& Time) .....	2-112
CaContainer GetPackageList(void) const .....	2-113
HINT32 Execute(bool bUseCurrentContext) .....	2-114
CaPromote .....	2-115
bool SetMergedPkgsOnly(bool bFlag) .....	2-115
bool SetCheckPackageDependencies(bool bCheckPackageDependencies) .....	2-116
bool SetCheckPackageGroupBind( bool bFlag) .....	2-116
bool GetMergedPkgsOnly(void) .....	2-117
bool GetCheckPackageDependencies (void) .....	2-117
bool GetCheckPackageGroupBind (void) .....	2-118
HINT32 Execute(void) .....	2-118
CaRemoveItem .....	2-119
bool SetDescription(LPCTSTR szDesc) .....	2-120
void SetItemObjId(HAROBJID iItemObjId) .....	2-120
void SetItemName(LPCTSTR szItemName) .....	2-121
void SetItemPath(LPCTSTR szItemPath) .....	2-121
void SetRecursive(bool bRecursive) .....	2-122
LPCTSTR GetDescription(void) .....	2-122
HAROBJID GetItemObjId(void) .....	2-123
LPCTSTR GetItemName(void) .....	2-123
LPCTSTR GetItemPath (void) .....	2-124
bool GetRecursive(void) .....	2-124
HINT32 Execute(void) .....	2-125
CaSQL .....	2-125
void SetSQLStatement( LPCTSTR szSQLStr ) .....	2-125
CaContainer GetSQLResult( void ) .....	2-126
HINT32 Execute(void) .....	2-126
CaTakeSnapshot .....	2-127
bool SetCanViewExternallyFlag(bool bFlag) .....	2-127
bool GetCanViewExternallyFlag(bool bFlag) .....	2-128
HINT32 Execute(LPCTSTR szViewName, CaTimeStamp snapshottime=NULL_CATIMESTAMP, LPCTSTR szReferenceSnapshot = NULL) .....	2-128
CaUDP .....	2-129
bool SetUsePackageList(bool bUsePackageChooser) .....	2-129
bool SetUseVersionList(bool bUseVersionChooser) .....	2-130

---

---

void SetInput(LPCTSTR szAddInput) .....	2-131
void SetAddCmdLineArg(LPCTSTR szCmdLineArg) .....	2-131
LPCTSTR GetUDPTType(void) .....	2-132
LPCTSTR GetProgramName(void) .....	2-132
LPCTSTR GetDescription(void) .....	2-133
bool GetUserCanModifyInput(void) .....	2-133
bool GetUsePackageList(void) .....	2-133
bool GetUseVersionList(void) .....	2-134
LPCTSTR GetAddCmdLineArg(void) .....	2-135
LPCTSTR GetInput(void) .....	2-135
HINT32 Execute(void) .....	2-136
CaVersionChooser .....	2-136
void Clear(void) .....	2-137
void SetParentPath(LPCTSTR szParentPath) .....	2-138
void SetItemObjId(HAROBJID iItemObjId, int iIndex) .....	2-138
void SetItemName(LPCTSTR szItemName) .....	2-139
void SetItemPackageObjId(HAROBJID iPackageObjId, int iIndex) .....	2-139
void SetPackageObjId(HAROBJID iPackageObjId, int iIndex) .....	2-139
void SetRecursive(bool bFlag) .....	2-140
void SetBranchOption(HINT32 iOptions) .....	2-140
void SetVersionOption(HINT32 iOptions) .....	2-141
void SetVersionItemOption(HINT32 iOptions) .....	2-141
void SetVersionNumber( LPCTSTR szVersion ) .....	2-142
void SetVersionStatusOption(HINT32 iOptions) .....	2-142
void SetVersionDateOption(HINT32 iOptions) .....	2-143
void SetFromDate(CaTimeStamp& Time) .....	2-144
void SetToDate(CaTimeStamp& Time) .....	2-144
CaContainer GetVersionList(void) const .....	2-145
HINT32 Execute(void) .....	2-145

## Chapter 3: HUtilities Class Descriptions

HUtilities Programming Fundamentals .....	3-1
CaBlob .....	3-2
void CaBlob(const CaBlob& rBlob) .....	3-2
void CaBlob(HINT32 nSize, void * lpData) .....	3-2
HINT32 GetSize(void) const .....	3-3
void * GetData(void) const .....	3-4
CaBlob& operator=( const CaBlob &blob ) .....	3-4
bool operator==( const CaBlob &rSrc) .....	3-4
bool operator!=( const CaBlob &rSrc) .....	3-5

friend ostream& operator<<( ostream &o, const CaBlob &rBlob ) .....	3-5
CaContainer .....	3-6
void Clear(void) .....	3-7
CaData & Props( LPCTSTR strKey, HINT32 iIndex) .....	3-7
HUINT32 GetKeyCount( void ) const .....	3-7
CaStringList GetKeyList( void ) const .....	3-8
HUINT32 GetKeyElementCount(LPCTSTR strKey) const .....	3-9
bool IsKeyExist( LPCTSTR strKey ) const .....	3-9
bool IsEmpty( void ) const .....	3-10
CaDataArray FindData(LPCTSTR strKey, LPCTSTR strValue) const .....	3-10
CaDataArray FindData(LPCTSTR strKey, HINT32 iValue) const .....	3-11
CaContainer & operator=(const CaContainer &CRight) .....	3-11
CaContainer & operator+=(const CaContainer &CRight) .....	3-12
friend ostream& operator<<( ostream &o, const CaContainer &c ) .....	3-12
CaData .....	3-13
void SetNull(void) .....	3-14
bool IsNull(void) const .....	3-14
CaData::Types GetType(void) const .....	3-15
CaData& operator=( const CaData &aData ) .....	3-15
CaData& operator=( HINT32 nValue) .....	3-16
CaData& operator=( LPCTSTR szValue) .....	3-16
CaData& operator=( const CaBlob& rBlob) .....	3-16
CaData& operator=( const CaTimeStamp& rTimeStamp) .....	3-16
bool operator==( const CaData &rSrc) .....	3-17
bool operator!=( const CaData &rSrc) .....	3-17
operator HINT32() .....	3-17
operator LPCTSTR() .....	3-18
operator CaString() .....	3-18
operator (CaBlob& )() .....	3-18
operator (CaTimeStamp& )() .....	3-19
friend ostream& operator<<( ostream &o, const CaData &d ) .....	3-19
CaDataArray .....	3-20
void Clear(void) .....	3-20
int GetSize(void) const .....	3-20
bool Find (const CaData& rData, HUINT32& rIndex) const .....	3-21
CaData & operator[](HUINT32 index) .....	3-21
friend ostream& operator<<( ostream &o, const CaDataArray &a ) .....	3-22
CaString .....	3-23
CaString(void) .....	3-24
CaString(const CaString &src) .....	3-24
CaString(LPCTSTR lpStr) .....	3-25
CaString(char* lpStr) .....	3-25

---

CaString& Append( LPCTSTR src ) .....	3-26
CaString& ToUpper ( void ) .....	3-26
CaString& ToLower ( void ) .....	3-27
bool IsEmpty(void) const .....	3-27
CaString& MakeEmpty( void ) .....	3-28
long GetLength(void) const .....	3-28
CaString& Trim( void ) .....	3-29
bool IsValid(void) const .....	3-29
const char* GetMultiByteString(void) .....	3-30
CaString& operator=( const CaString &src ) .....	3-30
CaString& operator=( LPCTSTR szSrc) .....	3-30
CaString& operator=( char* szSrc) .....	3-31
bool operator==( const CaString &rSrc) .....	3-31
bool operator==( LPCTSTR src) const .....	3-31
bool operator!=( const CaString &rSrc) const .....	3-31
bool operator!=( LPCTSTR src) const .....	3-32
CaString& operator+=( LPCTSTR src ) .....	3-32
CaString& operator-=( LPCTSTR src ) .....	3-32
operator LPCTSTR() .....	3-33
operator char *() .....	3-33
friend ostream& operator<<( ostream &o, const CaString &rString ) .....	3-34
CaStringList .....	3-35
CaStringList(void) .....	3-36
CaStringList(const CaStringList &src) .....	3-36
bool Add(const CaString szString) .....	3-37
CaString & Find(const CaString szString) .....	3-37
bool Remove(const CaString szString) .....	3-38
bool Add(LPCTSTR szString) .....	3-38
CaString & Find(LPCTSTR szString) .....	3-39
bool Remove(LPCTSTR szString) .....	3-39
bool IsEmpty(void) const .....	3-40
void Clean( void ) .....	3-40
int GetSize(void) const .....	3-41
CaString GetAllNames(void) const .....	3-41
Int GetIndex(LPCTSTR szString ) const .....	3-41
CaStringList& operator=( const CaStringList &CRight ) .....	3-42
bool operator==( const CaStringList &CRight) const .....	3-42
bool operator!=( const CaStringList &CRight) const .....	3-42
CaStringList& operator+=(const CaStringList &CRight) .....	3-43
CaStringList& operator-=(const CaStringList &CRight) .....	3-43
CaString & operator[] (int i) .....	3-44
operator bool() const .....	3-44

---

friend ostream& operator<<( ostream &o, const CaStringList &rList ) .....	3-45
CaTimeStamp .....	3-46
void CaTimeStamp(void).....	3-47
void CaTimeStamp(const CaTimeStamp& rTimeStamp) .....	3-47
void CaTimeStamp(HINT32 nYear, HINT32 nMonth, HINT32 nDay, HINT32 nHour, HINT32 nMinute, HINT32 nSecond, HINT32 nFraction) .....	3-48
CaTimeStamp& SetYear( HINT32 iYear ) .....	3-48
CaTimeStamp& SetMonth( HINT32 iMonth ) .....	3-49
CaTimeStamp& SetDay( HINT32 iDay ) .....	3-49
CaTimeStamp& SetHour( HINT32 iHour ).....	3-50
CaTimeStamp& SetMinute( HINT32 iMinute ) .....	3-50
CaTimeStamp& SetSecond( HINT32 iSecond ) .....	3-51
CaTimeStamp& SetFraction( HINT32 iFraction ) .....	3-51
HINT32 GetYear( void ) const .....	3-52
HINT32 GetMonth( void ) const.....	3-52
HINT32 GetDay( void ) const .....	3-53
HINT32 GetHour( void ) const .....	3-53
HINT32 GetMinute( void ) const .....	3-54
HINT32 GetSecond( void ) const .....	3-54
HINT32 GetFraction( void ) const .....	3-55
CPtHTimeStamp& ChangeTime(HINT32 iTimezone).....	3-55
CPtHTimeStamp& ConvertGMTTime(void) .....	3-56
CaTimeStamp& operator=( const CaTimeStamp &rTimeStamp ) .....	3-56
bool operator==( const CaTimeStamp &rTimeStamp) const .....	3-56
bool operator!=( const CaTimeStamp &rTimeStamp) const .....	3-57
friend ostream& operator<<( ostream &o, const CaTimeStamp &rTimeStamp ) .....	3-57

## Chapter 4: HSDK Messages

Messages and Descriptions .....	4-2
---------------------------------	-----



# Using the Harvest Software Development Kit

---

This chapter gives a brief overview of AllFusion Harvest Change Manager Software Development Kit (referred to simply as HSDK) and describes the fundamentals to usage of HSDK. (AllFusion Harvest Change Manager is referred to simply as Harvest throughout this guide.)

## Overview

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

This document assumes a thorough knowledge of the Harvest product and its usage.

## Operating System Requirements

HSDK has the following operating system requirements.

### Requirements for Sun Solaris

SPARCstation or SPARCserver running Solaris 2.5.5 with Sun Workshop 4.2 compiler. The binaries build runs on 2.6, 2.7, or 2.8 (see note) systems.

**Note:** Sun Workshop compiler 5 and Forte 6 require the flag `-compat=4` in the compile stream if built on Solaris 2.8 systems.

### Requirements for IBM AIX

IBM RISC System/6000 series workstations running AIX 4.3.3. IBM CXX compiler 3.6.6 compilers.

### Requirements for HP 9000/800 Series

HP 9000/800 series workstations running HP-UX 11.00. HP aCC compiler A.03.25 or higher.

### Requirements for Linux 2.4

Intel Workstation running Linux with kernel 2.4, glibc 2.2, and gcc 2.96, such as Redhat 7.1.

### Obsolete Versions

HSDK replaces the Harvest SIK released with the 4.x and previous versions of Harvest.

### Compatibility With Earlier Versions

The Harvest SIK was a C language interface to a subset of the Harvest capabilities. HSDK is a C++ class set that presents that functionality to the programmer. SIK functionality can be reprogrammed using HSDK but is not directly compatible.

## HSDK Programming Fundamentals

### Usage

The HSDK is available on both Windows and UNIX platforms.

1. Install Harvest prior to HSDK installation.
2. Install HSDK according to the installation instructions for the target platform. See the Installation section below.
3. Ensure that the platform specific library file is in your build/run path per the installation instructions for the target platform.
4. Include the hsdk.h file in an appropriate file in your application under development.
5. Build your application.

#### Header File

hsdk.h – Each HSDK class has its own header file. The relationships amongst the various HSDK classes make the order of header file inclusion important. hsdk.h includes all required HSDK header files in the proper sequence. It is the only header file required to include in your application files.

## Installation

The HSDK installation comes in a directory structure and includes an installation script to copy required files into the HARVESTHOME directory.

### HSDK Install and Test on Windows

Install the HSDK:

1. Copy the HSDK directory from the product CD into the %HARVESTHOME% directory.
2. Run InstallNT.bat

Establish a development environment:

1. Place the HSDK\include directory in the include path for you development tool.
2. Place the HSDK\lib directory in the library path for your development tool.
3. Include hsdk.h in any application file that uses HSDK classes.
4. Include hsdk.lib in the link command for the application.

Run the HSDK sample test program:

The HSDK comes with a sample program and the Microsoft Developer Studio project files to build and run it. The sample project was established with Developer Studio and Service Pack 3. The project file sets the proper include and lib paths for the build.

The sample program assumes the existence of a specific Harvest project in the Harvest repository. The SDKSampleProject.har file is a Harvest export file of the required project.

1. Execute himpenv.exe to import SDKSampleProject.har.
2. Create the user sdkuser with password sdkpass. Add it to the SDKSampleProject user group and to the approval list on the Approve process in state-1.
3. Create a read/write repository named "\sdkrep" and assign it to the baseline view of SDKSampleProject.
4. Create the directory C:\temp\hsdkdir.
5. Open HSDK\Samples\hsdksample.dsw in Developer Studio.
6. Build the project.
7. Set the program arguments to "-b brokername".
8. Run the program.

9. The file HSDKSample.log has the results.

## Install and Test on UNIX

Install the HSDK:

1. Copy the HSDK.tar file from the product CD into the \$HARVESTHOME location.
2. Extract HSDK.tar into \$HARVESTHOME, creating an HSDK directory in the \$HARVESTHOME location.
3. Execute the installunix.sh script.

Establish a development environment:

1. Place the HSDK/include directory in the include path for your development tool.
2. Place the HSDK/lib directory in the library path for your development tool.
3. Include hsdk.h in any application file that uses HSDK classes.
4. Include -lHSDK in the link command for the application.

Run the HSDK sample test program:

The HSDK comes with a sample program and the necessary Makefile and shell script (hsdksample.sh) to build it. The project file sets the proper include and lib paths for the build.

The sample program assumes the existence of a specific Harvest project in the Harvest repository. The SDKSampleProject.har file is a Harvest export file of the required project. You must setup up the Library path, system variables, and load the initialization script for PEC.

A typical example on a Solaris system:

```
HARVESTHOME=/usr/harvest
. /usr/pec/bin/rtinit.sh
LD_LIBRARY_PATH=/usr/local/Crypto:$LD_LIBRARY_PATH
LD_LIBRARY_PATH=${HARVESTHOME}/lib:$LD_LIBRARY_PATH
PATH=${HARVESTHOME}/bin:$PATH
export HARVESTHOME LD_LIBRARY_PATH PATH
```

1. Execute himpenv to import SDKSampleProject.har.
2. Create the user sdkuser with password sdkpass. Add it to the SDKSampleProject user group and to the approval list on the Approve process in state-1.
3. Create a read/write repository named "\sdkrep" and assign it to the baseline view of SDKSampleProject.

4. HSDK/Release and HSDK/Debug currently contain the compiled version of the sample program. Remove to rebuild.
5. Change directory (cd) to the Samples directory.
6. Execute script with targets DEBUG and/or RELEASE to build:  
**`./hskdsample.sh DEBUG RELEASE`**
7. Results are in HSDK/Release and HSDK/Debug.
8. Set \$HARVESTHOME environmental variable to be /usr/harvest (or equivalent).
9. Execute the files in either HSDK/Release or HSDK/Debug  
**`./hskdsample -b <brokername>`** (HSDK/Release)  
**`./hskdsample_d -b <brokername>`** (HSDK/Debug)
10. The file HSDKSample.log has the results.

## Variable Portability Type Definitions

Harvest 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 within a specified context. Once you log into 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 SIK returned object attributes and operational results in an HLIST structure. The HSDK provides a CaContainer object. The CaContainer is a text-keyed hash map of CaDataArray objects.

## Object Security

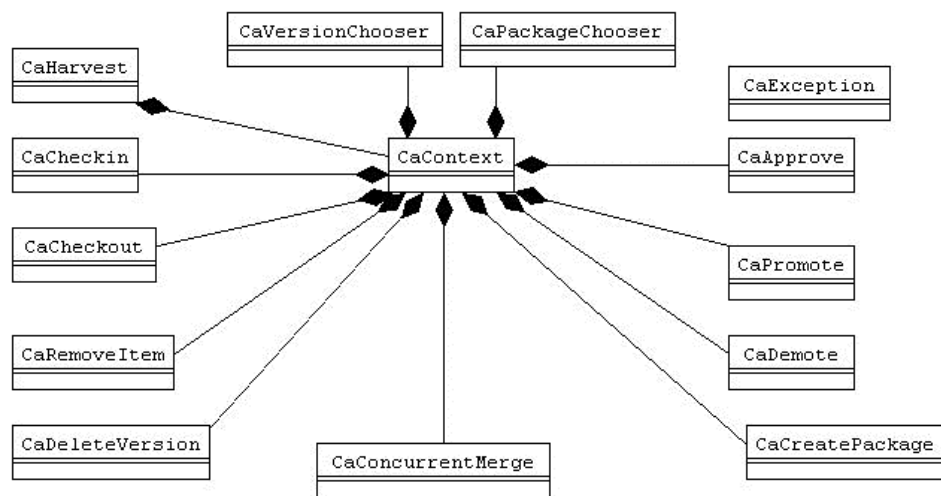
HSDK object security is maintained through the underlying Harvest modules.

## Constructors

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

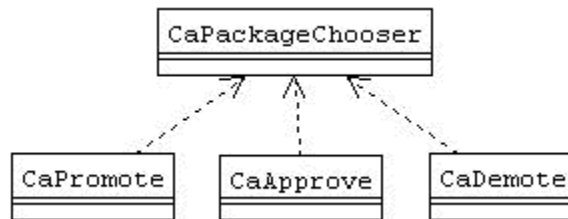
## HSDK Class Descriptions

### HSDK Class Hierarchy

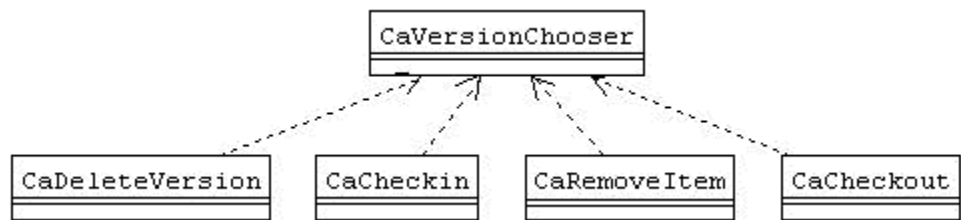


### Additional Class Dependencies

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



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



See Also

[HSDK Programming Fundamentals, Utilities Class Descriptions](#)

## HSDK Sample Program

The HSDK installation includes a sample program, `hdsksample.cpp`. The program comes with build algorithms for both UNIX and Windows environments. `Hdsksample` demonstrates HSDK functionality by exercising an operational scenario of Harvest operations.



# HSDK Class Descriptions

---

Each HSDK class is described in this chapter, including syntax and example.

## CaApprove

### Description

CaApprove encapsulates the Harvest 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 Harvest 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 CaPackageChooser object. Set the Approve attributes. Invoke the Execute function.

### Public Methods

SetApprove

SetDescription

GetApprove

GetDescription

Execute

### See Also

HSDK Programming Fundamentals, CaContext, CaPackageChooser

## Void SetApprove(bool bFlag)

Owning Class	CaApprove
Description	Sets the Approve/Reject flag true or false.
Arguments	bool bFlag – If true Approve the Change Package list, if false Reject it.
Return Value	None.
Notes	The Approve/Reject flag defaults to true.
Example	<pre>CaApprove&amp; rApprove = rContext.GetApprove(); rApprove.SetApprove(true);</pre>
See Also	HSDK Programming Fundamentals, GetApprove

## Void SetDescription(LPCTSTR szDescription)

Owning Class	CaApprove
Description	Sets the Approve/Reject Description used to annotate why the package list was approved or rejected.
Arguments	LPCTSTR szDescription – Description.
Return Value	None.
Notes	The Approve/Reject flag defaults to true.
Example	<pre>CaApprove&amp; rApprove = rContext.GetApprove(); rApprove.SetDescription ("I approve this package");</pre>
See Also	HSDK Programming Fundamentals, GetDescription

## bool GetApprove(void)

Owning Class	CaApprove
Description	Gets the current value of the Approve/Reject flag.
Arguments	None.
Return Value	flag value, true or false.
Notes	The Approve/Reject flag defaults to true.
Example	<pre>CaApprove&amp; rApprove = rContext.GetApprove(); bool bFlag = rApprove.GetApprove();</pre>
See Also	HSDK Programming Fundamentals, SetApprove

## LPCTSTR GetDescription(void)

Owning Class	CaApprove
Description	Gets the current value of the Approve/Reject Description.
Arguments	None.
Return Value	LPCTSTR – Description.
Notes	None.
Example	<pre>CaApprove&amp; rApprove = rContext.GetApprove(); CaString = rApprove.GetDescription();</pre>
See Also	HSDK Programming Fundamentals, SetDescription

## HINT32 Execute(void)

Owning Class	CaApprove
Description	Executes the Approve process on the Package list currently selected using the Package Chooser.
Arguments	None.
Return Value	HINT32 – zero if success. Non-zero if failure.

Notes None.

### Example

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

See Also HSDK Programming Fundamentals, CaContext, CaPackageChooser

## CaCheckin

**Description** CaCheckin encapsulates the Harvest 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 Harvest 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
	SetRemoteUserName
	SetRemotePassword
	SetFileList
	GetItemOption
	GetDescription
	GetParentTrunkMappedVersionList
	ClearParentTrunkMappedVersionList
	GetCheckinMode
	GetPathOption
	GetClientDir
	GetViewPath
	GetDeleteAfterCI
	GetRemoteMachineName
	GetRemoteUserName
	GetRemotePassword
	GetFileList

CreateItemPath

SetCallBack

SetCBUserData

GetCallBack

GetCBUserData

Execute

See Also                HSDK Programming Fundamentals, CaContext, CaVersionChooser, CaStringList

## **bool SetCheckinMode(HINT32 iMode)**

Owning Class            CaCheckin

Description             Sets the Checkin Mode for this process.

Arguments              HINT32 iMode - The Checkin Mode.

Return Value            Returns true if success, false if fail

Notes                    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 - Checkin the file and release the lock.

HAR\_UPDATE\_AND\_KEEP- Checkin the file and reserve it again for update.

HAR\_RELEASE\_ONLY- Release the lock but do not Checkin the file.

Example

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

See Also                HSDK Programming Fundamentals, CaContext, GetCheckinMode

**bool SetPathOption(HINT32 iOption)**

Owning Class	CaCheckin
Description	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.
Arguments	HINT32 iOption – The Path Option.
Return Value	Returns true if success, false if fail
Notes	<p>The following are the allowable values for the Path Option. They are defined in an include file built into the project.</p> <p>CI_OPTION_PRESERVE_DIR – Checkin each new version to its existing view path location that mirrors the location in the client path.</p> <p>CI_OPTION_PRESERVE_AND_CREATE_PATH – 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.</p> <p>CI_OPTION_ALL_FILE_TO_SAME_PATH – Checkin each new version to the single view path location specified.</p>
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); bool bResult = rCheckin.SetPathOption (CI_OPTION_PRESERVE_AND_CREATE_PATH);</pre>
See Also	HSDK Programming Fundamentals, CaContext, GetPathOption, SetClientDir, SetViewPath

## **bool SetItemOption(HINT32 iOption)**

Owning Class	CaCheckin
Description	Sets the Item Option for this process.
Arguments	HINT32 iOption – The Item Option.
Return Value	Returns true if success, false if fail
Notes	<p>The following are the allowable values for the Item Option. They are defined in an include file built into the project.</p> <p>CI_FILTER_NEW_OR_EXISTING_ITEMS – Checkin the file whether it or not it exists in the repository.</p> <p>CI_FILTER_NEW_ITEMS_ONLY – Checkin the file only if it does not yet exist in the repository.</p> <p>CI_FILTER_EXISTING_ITEMS_ONLY – Checkin the file only if it currently exists in the repository.</p>
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); bool bResult = rCheckin. SetItemOption (CI_FILTER_NEW_OR_EXISTING_ITEMS);</pre>
See Also	HSDK Programming Fundamentals, CaContext, GetItemOption

## **bool SetClientDir(LPCTSTR szCDir)**

Owning Class	CaCheckin
Description	Sets the Client Directory anchor for this process.
Arguments	LPCTSTR szCDir – A valid directory.
Return Value	Returns true if success, false if fail
Notes	If the Checkin is from a remote machine then this directory is from the remote machine.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); bool bResult = rCheckin.SetClientDir("c:\\temp\\work");</pre>
See Also	HSDK Programming Fundamentals, CaContext, GetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption

**bool SetViewPath(LPCTSTR szVPath)**

Owning Class	CaCheckin
Description	Sets the repository View Path anchor for this process.
Arguments	LPCTSTR szVPath – A valid repository view path.
Return Value	Returns true if success, false if fail
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); bool bResult = rCheckin.SetViewPath("\\rep\\work");</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetClientDir, SetRemoteMachineName, GetViewPath, SetPathOption

**bool SetDeleteAfterCI(bool bFlag)**

Owning Class	CaCheckin
Description	Sets the Delete After Checkin attribute of this process.
Arguments	bool bFlag – The flag value; true if delete the file, false if not.
Return Value	Returns true if success, false if fail
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); bool bResult = rCheckin.SetDeleteAfterCI(false);</pre>
See Also	HSDK Programming Fundamentals, CaContext, GetDeleteAfterCI

**bool SetDescription(LPCTSTR szDesc)**

Owning Class	CaCheckin
Description	Sets the Description to be associated with versions created by this Checkin.
Arguments	LPCTSTR szDesc – The description.
Return Value	Returns true if success, false if fail

Notes None.

Example

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

See Also HSDK Programming Fundamentals, CaContext, GetDescription

## **void SetRemoteMachineName(LPCTSTR szRemoteMachineName)**

Owning Class CaCheckin

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

Arguments LPCTSTR szRemoteMachineName – The machine name.

Return Value None.

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 Harvest File Agent must be running on the specified machine.

Example

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

See Also HSDK Programming Fundamentals, CaContext, SetClientDir, GetRemoteMachineName, SetViewPath, SetPathOption, SetRemoteUserName, SetRemotePassword

## **void SetRemoteUserName(LPCTSTR szRemoteUserName)**

Owning Class CaCheckin

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

Arguments LPCTSTR szRemoteUserName.

Return Value None.

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

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

See Also HSDK Programming Fundamentals, CaContext, SetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption, GetRemoteUserName, SetRemotePassword

### **void SetRemotePassword(LPCTSTR szRemotePassword)**

Owning Class CaCheckin

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

Arguments LPCTSTR szRemotePassword.

Return Value None.

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

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

See Also HSDK Programming Fundamentals, CaContext, SetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption, SetRemoteUserName, GetRemotePassword

### **void SetFileList(CaStringList &FileList)**

Owning Class CaCheckin

Description 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.

Arguments CaStringList &FileList – A list of file names to check in.

Return Value None.

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

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

See Also HSDK Programming Fundamentals, CaContext, GetFileList, SetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption, SetRemoteUserName, SetRemotePassword, CaStringList

## HINT32 GetItemOption(void)

Owning Class	CaCheckin
Description	Gets the Item Option attribute for this process.
Arguments	None.
Return Value	HINT32 – The value.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); HINT32 iOption = rCheckin.GetItemOption();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetItemOption

## LPCTSTR GetDescription(void)

Owning Class	CaCheckin
Description	Gets the Description attribute for this process.
Arguments	None.
Return Value	LPCTSTR – The value.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); CaString szValue = rCheckin.GetDescription();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetDescription

### HINT32 GetCheckinMode(void)

Owning Class	CaCheckin
Description	Gets the Checkin Mode attribute for this process.
Arguments	None.
Return Value	HINT32 – The value.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); HINT32 iOption = rCheckin.GetCheckinMode();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCheckinMode

### HINT32 GetPathOption(void)

Owning Class	CaCheckin
Description	Gets the Path Option attribute for this process.
Arguments	None.
Return Value	HINT32 – The value.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); HINT32 iOption = rCheckin.GetPathOption();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetPathOption

## LPCTSTR GetClientDir(void)

Owning Class	CaCheckin
Description	Gets the Client Directory anchor for this process.
Arguments	None.
Return Value	LPCTSTR – The value.
Notes	If a remote machine name is set then this is a directory on the remote machine.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); CaString szValue = rCheckin.GetClientDir();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetClientDir, SetRemoteMachineName

## LPCTSTR GetViewPath(void)

Owning Class	CaCheckin
Description	Gets the repository View Path anchor for this process.
Arguments	None.
Return Value	LPCTSTR – The value.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); CaString szValue = rCheckin.GetViewPath();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetViewPath

**bool GetDeleteAfterCI(void)**

Owning Class	CaCheckin
Description	Gets the Delete After Checkin attribute for this process.
Arguments	None.
Return Value	bool – The value.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); bool bFlag = rCheckin.GetDeleteAfterCI();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetDeleteAfterCI

**LPCTSTR GetRemoteMachineName(void)**

Owning Class	CaCheckin
Description	Gets remote machine name used for this process.
Arguments	None.
Return Value	LPCTSTR – The value.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); CaString szValue = rCheckin.GetRemoteMachineName();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetRemoteMachineName, SetRemoteUserName, SetRemotePassword

## LPCTSTR GetRemoteUserName(void)

Owning Class	CaCheckin
Description	Gets remote machine User Name used for this process.
Arguments	None.
Return Value	LPCTSTR – The value.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); CaString szValue = rCheckin.GetRemoteUserName();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetRemoteUserName, SetRemoteMachineName, SetRemotePassword

## LPCTSTR GetRemotePassword(void)

Owning Class	CaCheckin
Description	Gets remote machine Password used for this process.
Arguments	None.
Return Value	LPCTSTR – The value.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); CaString szValue = rCheckin.GetRemotePassword();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetRemotePassword, SetRemoteUserName, SetRemoteMachineName

## CaStringList GetFileList(void)

Owning Class	CaCheckin
Description	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.
Arguments	None.
Return Value	CaStringList – 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	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); CaStringList FileList = rCheckin.FetFileList();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetFileList, SetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption, SetRemoteUserName, SetRemotePassword, CaStringList

## HINT32 CreateItemPath( LPCTSTR szPathName, LPCTSTR szParentPath)

Owning Class	CaCheckin
Description	Creates the path in the current view.
Arguments	LPCTSTR - pathname to be created  LPCTSTR – path of parent which already exists
Return Value	HINT32 – result code for operation
Notes	To Create Item Path, need to have the package set in the context.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); /* To create 'h sdk' in '\HARVEST\src' */ rCheckin.CreateItemPath("h sdk", "\HARVEST\src");</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetFileList, SetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption, SetRemoteUserName, SetRemotePassword

**void SetCallBack( PFN\_CACICB pfCallBack )**

Owning Class	CaCheckin
Description	Sets the callback function for the checkin function, which is called after each item is checked in.
Arguments	PFN_CACICB – function pointer to callback.
Return Value	None.
Notes	<p>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);</p> <p>pBag contains data from checkout operation. The callback function has ownership of pBag and must delete the data it points to.</p>
Example	<pre>void cbFunction(int iBaseCBReturn, CaCheckin* pCheckin, void *pUserData, CaContainer *pBag = NULL) {     CaString* pString = (CaString*)pUserData;     CaString szViewPath = (LPCTSTR)pBag-&gt;Props(ATTR_KEY_VIEWPATH, 0);     cout &lt;&lt; *pString &lt;&lt; szViewPath &lt;&lt; endl; }  CaCheckin&amp; rCheckin = rContext.GetCheckin(); rCheckin.SetCallBack( cbFunction );</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCBUserData, GetCallBack, GetCBUserData

## **void SetCBUserData( void\* pUserData )**

Owning Class	CaCheckin
Description	Stores the user data that will be passed to the callback function on each callback execution.
Arguments	void* - User defined data type that is expected by callback function.
Return Value	None.
Notes	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.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); rCheckin.SetCBUserData( (void*) &amp;usrData );</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCallBack, GetCallBack, GetCBUserData

## **PFN\_CACICB GetCallBack(void);**

Owning Class	CaCheckin
Description	Returns the callback function pointer for the checkin function, which is called after each item is checked in.
Arguments	None.
Return Value	PFN_CACICB – function pointer to callback.
Notes	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);
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); PFN_CACICB pfCallback = rCheckin.GetCallBack();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCBUserData, SetCallBack, GetCBUserData

**void\* GetCBUserData(void);**

Owning Class	CaCheckin
Description	Retrieves the user data that will be passed to the callback function on each callback execution.
Arguments	None.
Return Value	void* - User defined data type that is expected by callback function.
Notes	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.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); void*rCheckin.GetCBUserData( (void*) &amp;usrData );</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCallBack, GetCallBack, GetCBUserData

**HINT32 Execute(bool bFileList)**

Owning Class	CaCheckin
Description	Executes the Checkin process on the Version list currently selected using the Version Chooser.
Arguments	bool bFileList – If true checks in a file list, if false gets the list from the version chooser. The default value is true.
Return Value	HINT32 – zero if success, non-zero if failure.
Notes	None.
Example	<pre>CaCheckin&amp; rCheckin = rContext.GetCheckin(); HINT32 iResult = rCheckin.Execute(false);</pre>
See Also	HSDK Programming Fundamentals, CaContext, CaVersionChooser, CaStringList

## CaCheckout

**Description** CaCheckout encapsulates the Harvest 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 Harvest 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
- SetRemoteUserName
- SetRemotePassword
- SetMVSFileTypes
- GetReplaceFile
- GetShareWorkDir
- GetCheckoutMode
- GetPathOption
- GetClientDir
- GetViewPath
- GetUseCITimeStamp
- GetRemoteMachineName

GetRemoteUserName

GetRemotePassword

GetMVSFileTypes

SetCallBack

SetCBUserData

GetCallBack

GetCBUserData

Execute

See Also                HSDK Programming Fundamentals, CaContext, CaVersionChooser, CaStringList

## **bool SetCheckoutMode(HINT32 iMode)**

Owning Class            CaCheckout

Description             Sets the Checkout Mode for this process.

Arguments              HINT32 iMode - The Checkout Mode.

Return Value            Returns 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.

CO\_MODE\_BROWSE - Checkout the versions for read only. No lock is set.

CO\_MODE\_UPDATE - Checkout 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 - 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.

CO\_MODE\_CONCURRENT - Checkout the versions for concurrent update and still allow other reservations with other packages.

Example

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

See Also                    HSDK Programming Fundamentals, CaContext, GetCheckoutMode

## **bool SetPathOption(HINT32 iOption)**

Owning Class              CaCheckout

Description                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.

Arguments                 HINT32 iOption – The Path Option.

Return Value               Returns true if success, false if fail

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

CO\_OPTION\_PRESERVE – Checkout each new version to its existing client path location that mirrors the location in the view path.

CO\_OPTION\_PRESERVE\_AND\_CREATE – Checkout 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 – Checkout each new version to the single client path location specified.

Example

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

See Also                    HSDK Programming Fundamentals, CaContext, GetPathOption, SetClientDir, SetViewPath

**bool SetReplaceFile(bool bFlag)**

Owning Class	CaCheckout
Description	Sets Replace File attribute for this process.
Arguments	bool bFlag – If true then read only files are replaced, if false they are not.
Return Value	Returns true if success, false if fail
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); bool bResult = rCheckout.SetReplaceFile(true);</pre>
See Also	HSDK Programming Fundamentals, CaContext, GetReplaceFile

**bool SetClientDir(LPCTSTR szCDir)**

Owning Class	CaCheckout
Description	Sets the Client Directory anchor for this process.
Arguments	LPCTSTR szCDir – A valid directory.
Return Value	Returns true if success, false if fail
Notes	If the Checkout is from a remote machine then this directory is from the remote machine.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); bool bResult = rCheckout.SetClientDir("c:\\temp\\work");</pre>
See Also	HSDK Programming Fundamentals, CaContext, GetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption

**bool SetViewPath(LPCTSTR szVPath)**

Owning Class	CaCheckout
Description	Sets the repository View Path anchor for this process.
Arguments	LPCTSTR szVPath – A valid repository view path.
Return Value	Returns true if success, false if fail
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); bool bResult = rCheckout.SetViewPath("\\rep\\work");</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetClientDir, SetRemoteMachineName, GetViewPath, SetPathOption

**bool SetShareWorkDir(bool bFlag)**

Owning Class	CaCheckout
Description	Sets the Delete After Checkout attribute of this process.
Arguments	bool bFlag – The flag value; true if the client directory is shared, false if not.
Return Value	Returns 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	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); bool bResult = rCheckout.SetShareWorkDir(false);</pre>
See Also	HSDK Programming Fundamentals, CaContext, GetShareWorkDir

**bool SetUseCITimeStamp(bool bFlag)**

Owning Class	CaCheckout
Description	Sets the Use Checkin Timestamp attribute to be used by this Checkout.
Arguments	bool bFlag – if true sets time on file to time version was checked in, if false sets it to current time.
Return Value	Returns true if success, false if fail
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); bool bResult = rCheckout.SetUseCITimeStamp(true);</pre>
See Also	HSDK Programming Fundamentals, CaContext, GetUseCITimeStamp

**void SetRemoteMachineName(LPCTSTR szRemoteMachineName)**

Owning Class	CaCheckout
Description	Sets the host name of the remote machine where the files are to be located.
Arguments	LPCTSTR szRemoteMachineName – The machine name.
Return Value	None.
Notes	<p>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.</p> <p>A remote Harvest File Agent must be running on the specified machine.</p>
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); rCheckout.SetRemoteMachineName("MyBuildMachine");</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetClientDir, GetRemoteMachineName, SetViewPath, SetPathOption, SetRemoteUserName, SetRemotePassword

**void SetRemoteUserName(LPCTSTR szRemoteUserName)**

Owning Class	CaCheckout
--------------	------------

Description	Sets the user name for the login to the remote machine where the files are to be located.
Arguments	LPCTSTR szRemoteUserName.
Return Value	None.
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	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout();rCheckout.SetRemoteUserName("George");</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption, GetRemoteUserName, SetRemotePassword

### **void SetRemotePassword(LPCTSTR szRemotePassword)**

Owning Class	CaCheckout
Description	Sets the password for the username used to login to the remote machine where the files are to be located.
Arguments	LPCTSTR szRemotePassword.
Return Value	None.
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	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout();rCheckout.SetRemotePassword("georgespassword");</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption, SetRemoteUserName, GetRemotePassword

### **bool SetMVSFileTypes(HINT32 iOption)**

Owning Class	CaCheckout
Description	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.

Arguments	HINT32 iOption – The MVS File Types Option.
Return Value	Returns true if success, false if fail.
Notes	<p>The following are the allowable values for the Checkout Mode. They are defined in an include file built into the project.</p> <p>HAR_MVS_PDS – The target file location is a PDS.</p> <p>HAR_MVS_SEQUENTIAL – The target file location is a Sequential file.</p> <p>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.</p>
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); rCheckout.SetMVSFileTypes(HAR_MVS_PDS);</pre>
See Also	HSDK Programming Fundamentals, CaContext, GetMVSFileTypes, SetClientDir, SetRemoteMachineName, SetViewPath, SetPathOption, SetRemoteUserName, SetRemotePassword

## HINT32 GetCheckoutMode(void)

Owning Class	CaCheckout
Description	Gets the Checkout Mode attribute for this process.
Arguments	None.
Return Value	HINT32 – The value.
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); HINT32 iOption = rCheckout.GetCheckoutMode();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCheckoutMode

## HINT32 GetPathOption(void)

Owning Class	CaCheckout
Description	Gets the Path Option attribute for this process.
Arguments	None.

Return Value	HINT32 – The value.
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); HINT32 iOption = rCheckout.GetPathOption();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetPathOption

### **LPCTSTR GetClientDir(void)**

Owning Class	CaCheckout
Description	Gets the Client Directory anchor for this process.
Arguments	None.
Return Value	LPCTSTR – The value.
Notes	If a remote machine name is set then this is a directory on the remote machine.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); CaString szValue = rCheckout.GetClientDir();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetClientDir, SetRemoteMachineName

### **LPCTSTR GetViewPath(void)**

Owning Class	CaCheckout
Description	Gets the repository View Path anchor for this process.
Arguments	None.
Return Value	LPCTSTR – The value.
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); CaString szValue = rCheckout.GetViewPath();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetViewPath

## **bool GetReplaceFile(void)**

Owning Class	CaCheckout
Description	Gets the Replace File attribute for this process.
Arguments	None.
Return Value	bool – The value.
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); bool bOption = rCheckout.GetReplaceFile();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetReplaceFile

## **bool GetShareWorkDir(void)**

Owning Class	CaCheckout
Description	Gets the Share Working Directory attribute for this process.
Arguments	None.
Return Value	bool – The value.
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); bool bFlag = rCheckout.GetShareWorkDir();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetShareWorkDir

## **bool GetUseCITimeStamp(void)**

Owning Class	CaCheckout
Description	Gets the Use Checkin Timestamp attribute for this process.
Arguments	None.
Return Value	bool – The value.
Notes	None.

**Example**

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

See Also           HSDK Programming Fundamentals, CaContext, SetUseCITimeStamp

**LPCTSTR GetRemoteMachineName(void)**

Owning Class       CaCheckout

Description       Gets remote machine name used for this process.

Arguments         None.

Return Value      LPCTSTR – The value.

Notes             None.

**Example**

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

See Also           HSDK Programming Fundamentals, CaContext, SetRemoteMachineName,  
SetRemoteUserName, SetRemotePassword

**LPCTSTR GetRemoteUserName(void)**

Owning Class       CaCheckout

Description       Gets remote machine User Name used for this process.

Arguments         None.

Return Value      LPCTSTR – The value.

Notes             None.

**Example**

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

See Also           HSDK Programming Fundamentals, CaContext, SetRemoteUserName,  
SetRemoteMachineName, SetRemotePassword

## LPCTSTR GetRemotePassword(void)

Owning Class	CaCheckout
Description	Gets remote machine Password used for this process.
Arguments	None.
Return Value	LPCTSTR – The value.
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); CaString szValue = rCheckout.GetRemotePassword();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetRemotePassword, SetRemoteUserName, SetRemoteMachineName

## HINT32 GetMVSFileTypes(void)

Owning Class	CaCheckout
Description	Gets the MVSFile Types attribute used by this process.
Arguments	None.
Return Value	HINT32 – the value.
Notes	This attribute is only used if the remote machine name is set to an MVS platform.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout();HINT32 iValue = rCheckout.GetMVSFileTypes();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetMVSFileTypes, SetRemoteMachineName

**void SetCallBack( PFN\_CACOCB pfCallBack )**

Owning Class	CaCheckout
Description	Sets the callback function for the checkout function, which is called after each item is checked in.
Arguments	PFN_CACOCB – function pointer to callback.
Return Value	None.
Notes	<p>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);</p> <p>pBag contains data from checkout operation. The callback function has ownership of pBag and must delete the data it points to.</p>
Example	<pre>void cbFunction(int iBaseCBReturn, CaCheckout* pCheckout, void *pUserData, CaContainer *pBag = NULL) {     CaString* pString = (CaString*)pUserData;     CaString szViewPath = (LPCTSTR)pBag-&gt;Props(ATTR_KEY_VIEWPATH, 0);     cout &lt;&lt; *pString &lt;&lt; szViewPath &lt;&lt; endl; }  CaCheckout&amp; rCheckout = rContext.GetCheckout(); rCheckout.SetCallBack( cbFunction );</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCBUserData, GetCallBack, GetCBUserData

## **void SetCBUserData( void\* pUserData )**

Owning Class	CaCheckout
Description	Stores the user data that will be passed to the callback function on each callback execution.
Arguments	void* - User defined data type that is expected by callback function.
Return Value	None.
Notes	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.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); rCheckout.SetCBUserData( (void*) &amp;usrData );</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCallBack, GetCallBack, GetCBUserData

## **PFN\_CACOCB GetCallBack(void);**

Owning Class	CaCheckout
Description	Returns the callback function pointer for the checkout function, which is called after each item is checked in.
Arguments	None.
Return Value	PFN_CACOCB – function pointer to callback.
Notes	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);
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); PFN_CACOCB pfCallback = rCheckout.GetCallBack();</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCBUserData, SetCallBack, GetCBUserData

**void\* GetCBUserData(void);**

Owning Class	CaCheckout
Description	Retrieves the user data that will be passed to the callback function on each callback execution.
Arguments	None.
Return Value	void* - User defined data type that is expected by callback function.
Notes	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.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); void*rCheckout.GetCBUserData( (void*) &amp;usrData );</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetCallBack, GetCallBack, GetCBUserData

**HINT32 Execute(void)**

Owning Class	CaCheckout
Description	Executes the Checkout process on the Version list currently selected using the Version Chooser.
Arguments	None.
Return Value	HINT32 – zero if success, non-zero if failure.
Notes	None.
Example	<pre>CaCheckout&amp; rCheckout = rContext.GetCheckout(); HINT32 iResult = rCheckout.Execute();</pre>
See Also	HSDK Programming Fundamentals, CaContext, CaVersionChooser

## CaConcurrentMerge

Description	<p>CaConcurrentMerge encapsulates the Harvest 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 Harvest administrator sets default attribute values.</p> <p>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.</p>
Public Methods	<p>SetMergeRule</p> <p>SetVersionDescription</p> <p>GetMergeRule</p> <p>GetVersionDescription</p> <p>Execute</p>
See Also	<p>HSDK Programming Fundamentals, CaContext</p>

**void SetMergeRule( LPCTSTR szRule )**

Owning Class           CaConcurrentMerge

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

Arguments            LPCTSTR szRule – The rule value.

Return Value          None.

Notes                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.

Example

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

See Also            HSDK Programming Fundamentals, GetMergeRule

**void SetVersionDescription( LPCTSTR szDescription )**

Owning Class	CaConcurrentMerge
Description	Provides a description to be associated with the merged version.
Arguments	LPCTSTR szDescription.
Return Value	None.
Notes	None.
Example	<pre>CaConcurrentMerge&amp; rConcurrentMerge = rContext.GetConcurrentMerge(); rConcurrentMerge.SetVersionDescription("My Merged Version");</pre>
See Also	HSDK Programming Fundamentals, GetVersionDescription

**LPCTSTR GetMergeRule(void)**

Owning Class	CaConcurrentMerge
Description	Gets the current value of the Merge Rule Option.
Arguments	None.
Return Value	LPCTSTR – current value of option.
Notes	None.
Example	<pre>CaConcurrentMerge&amp; rConcurrentMerge = rContext.GetConcurrentMerge(); CaString = rConcurrentMerge.GetMergeRule();</pre>
See Also	HSDK Programming Fundamentals, SetMergeRule

### LPCTSTR GetVersionDescription(void)

Owning Class           CaConcurrentMerge

Description           Gets the current value of the Version Description.

Arguments           None.

Return Value          LPCTSTR – current value of description.

Notes                None.

Example

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

See Also            HSDK Programming Fundamentals, SetVersionDescription

### HINT32 Execute(void)

Owning Class           CaConcurrentMerge

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

Arguments           None.

Return Value          HINT32 – zero if success. Non-zero if failure.

Notes                None.

Example

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

See Also            HSDK Programming Fundamentals, CaContext

## CaContext

### Description

CaContext is a smart container that manages the current Harvest 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

GetStateId

GetPackageName

GetPackageId

GetViewName

GetViewId

GetSnapshotViewName

GetSnapshotViewId

GetCheckoutName

GetPromoteName

GetDemoteName

GetSnapshotViewName

GetCheckout

GetCheckin

GetPromote

GetDemote  
GetClientPath  
GetViewPath  
GetSQL  
GetPackageChooser  
GetVersionChooser  
GetStateList  
SetProject  
SetState  
SetCheckout  
SetCheckin  
SetPromote  
SetDemote  
SetPackage  
SetSnapshotView  
SetClientPath  
SetViewPath  
GetApproveName  
GetApprove  
SetApprove  
GetCreatePackageName  
GetCreatePackage  
SetCreatePackage  
SetWebserverName  
GetDeleteVersionName

GetDeleteVersion

SetDeleteVersion

GetRemoveItemName

GetRemoveItem

SetRemoveItemName

GetConcurrentMergeName

GetConcurrentMerge

SetConcurrentMergeName

SetCrossProjectMerge

GetCrossProjectMerge

GetCrossProjectMergeName

SetTakeSnapshot

GetTakeSnapshot

GetTakeSnapshotName

SetUDP

GetUDP

GetUDPName

GetUserName

GetWebserverName

IsPackageSet

See Also

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

### LPCTSTR GetProjectName(void) const

Owning Class            CaContext

Description            Retrieve the name of the current Harvest Project.

Arguments              None.

Return Value           Project name if set, empty string if empty.

Example

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

See Also                HSDK Programming Fundamentals, SetProject, GetProjectId

### HAROBJID GetProjectId(void) const

Owning Class            CaContext

Description            Retrieve the Object Id of the current Harvest Project.

Arguments              None.

Return Value           Project Object Id if set, NULL\_HAROBJID if empty.

Example

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

See Also                HSDK Programming Fundamentals, SetProject, GetProjectName

### LPCTSTR GetStateName(void) const

Owning Class            CaContext

Description            Retrieve the name of the current Harvest State.

Arguments              None.

Return Value           State name if set, empty string if empty.

Example

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

See Also                HSDK Programming Fundamentals, SetState, GetStateId

## HAROBJID GetStateId(void) const

Owning Class	CaContext
Description	Retrieve the Object Id of the current Harvest State.
Arguments	None.
Return Value	State Object Id if set, NULL_HAROBJID if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); HAROBJID iObjId = rContext.GetStateId();</pre>
See Also	HSDK Programming Fundamentals, SetState, GetStateName

## LPCTSTR GetPackageName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest Package.
Arguments	None.
Return Value	Package name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetPackageName();</pre>
See Also	HSDK Programming Fundamentals, SetPackage, GetPackageId

## HAROBJID GetPackageld(void) const

Owning Class	CaContext
Description	Retrieve the Object Id of the current Harvest Package.
Arguments	None.
Return Value	Package Object Id if set, NULL_HAROBJID if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); HAROBJID iObjId = rContext.GetPackageId();</pre>
See Also	HSDK Programming Fundamentals, SetPackage, GetPackageName

## HAROBJID GetViewName(void) const

Owning Class	CaContext
Description	Retrieve the object id of the current Harvest View.
Arguments	None.
Return Value	View object id if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); HAROBJID iId = rContext.GetViewId();</pre>
See Also	HSDK Programming Fundamentals, SetState

## LPCTSTR GetViewName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest View.
Arguments	None.
Return Value	View name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetViewName();</pre>
See Also	HSDK Programming Fundamentals, SetState

## LPCTSTR GetSnapshotViewName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest SnapshotView.
Arguments	None.
Return Value	SnapshotView name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetSnapshotViewName();</pre>
See Also	HSDK Programming Fundamentals, SetSnapshotView, GetSnapshotViewId

## HAROBJID GetSnapshotViewId(void) const

Owning Class	CaContext
Description	Retrieve the object id of the current Harvest SnapshotView.
Arguments	None.
Return Value	SnapshotView id if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); HAROBJID iSnapshot = rContext.GetSnapshotViewId();</pre>
See Also	HSDK Programming Fundamentals, SetSnapshotView, GetSnapshotViewName

## LPCTSTR GetCheckoutName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest Checkout.
Arguments	None.
Return Value	Checkout name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetCheckoutName();</pre>
See Also	HSDK Programming Fundamentals, SetCheckout, GetCheckout

## LPCTSTR GetCheckinName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest Checkin.
Arguments	None.
Return Value	Checkin name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetCheckinName();</pre>
See Also	HSDK Programming Fundamentals, SetCheckin, GetCheckin

### LPCTSTR GetPromoteName(void) const

Owning Class            CaContext

Description            Retrieve the name of the current Harvest Promote.

Arguments              None.

Return Value           Promote name if set, empty string if empty.

Example

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

See Also                HSDK Programming Fundamentals, SetPromote, GetPromote

### LPCTSTR GetDemoteName(void) const

Owning Class            CaContext

Description            Retrieve the name of the current Harvest Demote.

Arguments              None.

Return Value           Demote name if set, empty string if empty.

Example

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

See Also                HSDK Programming Fundamentals, SetDemote, GetDemote

### LPCTSTR GetApproveName(void) const

Owning Class            CaContext

Description            Retrieve the name of the current Harvest Approve.

Arguments              None.

Return Value           Approve name if set, empty string if empty.

Example

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

See Also                HSDK Programming Fundamentals, SetApprove, GetApprove

## LPCTSTR GetCreatePackageName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest CreatePackage.
Arguments	None.
Return Value	CreatePackage name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetCreatePackageName();</pre>
See Also	HSDK Programming Fundamentals, SetCreatePackage, GetCreatePackage

## LPCTSTR GetDeleteVersionName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest DeleteVersion.
Arguments	None.
Return Value	DeleteVersion name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetDeleteVersionName();</pre>
See Also	HSDK Programming Fundamentals, SetDeleteVersion, GetDeleteVersion

## LPCTSTR GetRemoveItemName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest RemoveItem.
Arguments	None.
Return Value	RemoveItem name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetRemoveItemName();</pre>
See Also	HSDK Programming Fundamentals, SetRemoveItem, GetRemoveItem

## LPCTSTR GetConcurrentMergeName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest ConcurrentMerge.
Arguments	None.
Return Value	ConcurrentMerge name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetConcurrentMergeName();</pre>
See Also	HSDK Programming Fundamentals, SetConcurrentMerge, GetConcurrentMerge

## LPCTSTR GetCrossProjectMergeName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest CrossProjectMerge.
Arguments	None.
Return Value	CrossProjectMerge name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetCrossProjectMergeName();</pre>
See Also	HSDK Programming Fundamentals, SetCrossProjectMerge, GetCrossProjectMerge

## LPCTSTR GetTakeSnapshotName(void) const

Owning Class	CaContext
Description	Retrieve the name of the current Harvest TakeSnapshot.
Arguments	None.
Return Value	TakeSnapshot name if set, empty string if empty.
Example	<pre>CaHarvest Harvest; CaContext&amp; rContext = Harvest.GetContext(); CaString szName = rContext.GetTakeSnapshotName();</pre>

See Also [HSDK Programming Fundamentals](#), [SetTakeSnapshot](#), [GetTakeSnapshot](#)

### **LPCTSTR GetUDPName(void) const**

Owning Class [CaContext](#)

Description Retrieve the name of the current Harvest UDP.

Arguments None.

Return Value UDP name if set, empty string if empty.

Example

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

See Also [HSDK Programming Fundamentals](#), [SetUDP](#), [GetUDP](#)

### **LPCTSTR GetUserName(void) const**

Owning Class [CaContext](#)

Description Retrieve the name of the current Harvest user.

Arguments None.

Return Value User name of currently logged in user.

Example

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

See Also [HSDK Programming Fundamentals](#)

### **LPCTSTR GetWebserverName(void)**

Owning Class [CaContext](#)

Description Retrieve the name of the current Harvest Harweb server.

Arguments None.

Return Value Server name of Harweb server.

**Example**

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

See Also           HSDK Programming Fundamentals

**HAROBJID GetUserId(void) const**

Owning Class       CaContext

Description        Retrieve the object id of the current Harvest user.

Arguments          None.

Return Value       Object id of currently logged in user.

**Example**

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

See Also           HSDK Programming Fundamentals

**LPCTSTR GetClientPath(void) const**

Owning Class       CaContext

Description        Retrieve the current Harvest client path.

Arguments          None.

Return Value       Client path if set, empty string if empty.

**Example**

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

See Also           HSDK Programming Fundamentals, SetClientPath

**LPCTSTR GetViewPath(void) const**

Owning Class       CaContext

Description        Retrieve the current Harvest View path.

Arguments          None.

Return Value View path if set, empty string if empty.

Example

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

See Also HSDK Programming Fundamentals, SetViewPath

## CaCheckout& GetCheckout(void) const

Owning Class CaContext

Description Retrieve the current Harvest Checkout object.

Arguments None.

Return Value Reference to the current Checkout object if set.

Exceptions Throws CaException on error.

Example

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

See Also HSDK Programming Fundamentals, CaException, SetCheckout, GetCheckoutName

## CaCheckin& GetCheckin(void) const

Owning Class CaContext

Description Retrieve the current Harvest Checkin object.

Arguments None.

Return Value Reference to the current Checkin object if set.

Exceptions Throws CaException on error.

Example

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

See Also HSDK Programming Fundamentals, CaException, SetCheckin, GetCheckinName

### CaPromote& GetPromote(void) const

Owning Class	CaContext
Description	Retrieve the current Harvest Promote object.
Arguments	None.
Return Value	Reference to the current Promote object if set.
Exceptions	Throws CaException on error.
Example	<pre>CaPromote&amp; rPromote = rContext.GetPromote();</pre>
See Also	HSDK Programming Fundamentals, CaException, SetPromote, GetPromoteName

### CaDemote& GetDemote(void) const

Owning Class	CaContext
Description	Retrieve the current Harvest Demote object.
Arguments	None.
Return Value	Reference to the current Demote object if set.
Exceptions	Throws CaException on error.
Example	<pre>CaDemote&amp; rDemote = rContext.GetDemote();</pre>
See Also	HSDK Programming Fundamentals, CaException, SetDemote, GetDemoteName

### CaApprove& GetApprove(void) const

Owning Class	CaContext
Description	Retrieve the current Harvest Approve object.
Arguments	None.
Return Value	Reference to the current Approve object if set.
Exceptions	Throws CaException on error.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException, SetApprove, GetApproveName

**CaCreatePackage& GetCreatePackage(void) const****Owning Class**

CaContext

**Description**

Retrieve the current Harvest CreatePackage object.

**Arguments**

None.

**Return Value**

Reference to the current CreatePackage object if set.

**Exceptions**

Throws CaException on error.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException, SetCreatePackage, GetCreatePackageName

**CaDeleteVersion& GetDeleteVersion(void) const****Owning Class**

CaContext

**Description**

Retrieve the current Harvest DeleteVersion object.

**Arguments**

None.

**Return Value**

Reference to the current DeleteVersion object if set.

**Exceptions**

Throws CaException on error.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException, SetDeleteVersion, GetDeleteVersionName

### CaRemoveItem& GetRemoveItem(void) const

Owning Class	CaContext
Description	Retrieve the current Harvest RemoveItem object.
Arguments	None.
Return Value	Reference to the current RemoveItem object if set.
Exceptions	Throws CaException on error.
Example	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem();</pre>
See Also	HSDK Programming Fundamentals, CaException, SetRemoveItem, GetRemoveItemName

### CaConcurrentMerge& GetConcurrentMerge(void) const

Owning Class	CaContext
Description	Retrieve the current Harvest ConcurrentMerge object.
Arguments	None.
Return Value	Reference to the current ConcurrentMerge object if set.
Exceptions	Throws CaException on error.
Example	<pre>CaConcurrentMerge&amp; rConcurrentMerge = rContext.GetConcurrentMerge();</pre>
See Also	HSDK Programming Fundamentals, CaException, SetConcurrentMerge, GetConcurrentMergeName

### CaCrossProjectMerge& GetCrossProjectMerge(void) const

Owning Class	CaContext
Description	Retrieve the current Harvest CrossProjectMerge object.
Arguments	None.
Return Value	Reference to the current CrossProjectMerge object if set.

Exceptions Throws CaException on error.

Example

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

See Also HSDK Programming Fundamentals, CaException, SetCrossProjectMerge, GetCrossProjectProjectMergeName

## CaTakeSnapshot& GetTakeSnapshot(void) const

Owning Class CaContext

Description Retrieve the current Harvest TakeSnapshot object.

Arguments None.

Return Value Reference to the current TakeSnapshot object if set.

Exceptions Throws CaException on error.

Example

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

See Also HSDK Programming Fundamentals, CaException, SetTakeSnapshot, GetTakeSnapshotName

## CaUDP& GetUDP(void) const

Owning Class CaContext

Description Retrieve the current Harvest UDP object.

Arguments None.

Return Value Reference to the current UDP object if set.

Exceptions Throws CaException on error.

Example

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

See Also HSDK Programming Fundamentals, CaException, SetUDP, GetUDPName

### CaSQL& GetSQL(void) const

Owning Class	CaContext
Description	Retrieve the Harvest SQL object.
Arguments	None.
Return Value	Reference to the SQL object.
Exceptions	Throws CaException on error.
Example	<pre>CaSQL&amp; rSQL = rContext.GetSQL();</pre>
See Also	HSDK Programming Fundamentals, CaException

### CaPackageChooser& GetPackageChooser(void) const

Owning Class	CaContext
Description	Retrieve the Harvest PackageChooser object.
Arguments	None.
Return Value	Reference to the PackageChooser object.
Exceptions	Throws CaException on error.
Example	<pre>CaPackageChooser&amp; rPackageChooser = rContext.GetPackageChooser();</pre>
See Also	HSDK Programming Fundamentals, CaException

### CaVersionChooser& GetVersionChooser(void) const

Owning Class	CaContext
Description	Retrieve the Harvest VersionChooser object.
Arguments	None.
Return Value	Reference to the VersionChooser object.
Exceptions	Throws CaException on error.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException

## CaHarvest& GetHarvestSession (void) const

**Owning Class**

CaContext

**Description**

Retrieve the current CaHarvest object.

**Arguments**

None.

**Return Value**

Reference to the CaHarvest object.

**Exceptions**

Throws CaException on error.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException

## CaContainer GetStateList(void) const

**Owning Class**

CaContext

**Description**

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

**Arguments**

None.

**Return Value**

Returns a CaContainer object.

**Exceptions**

Throws CaException on 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**

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

See Also                    HSDK Programming Fundamentals, CaException, CaContainer

## CaContainer GetViewList(void) const

Owning Class	CaContext
Description	Retrieve a CaContainer object that contains the attributes for each Harvest View in the list.
Arguments	None.
Return Value	Returns a CaContainer object.
Exceptions	Throws CaException on error.
Notes	List attributes are stored with the following hash keys:  <i>CA_ATTRKEY_NAME, CA_ATTRKEY_OBJID, CA_ATTRKEY_PROJECTID, CA_ATTRKEY_VIEW_TYPE</i>

### Example

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

See Also                    HSDK Programming Fundamentals, CaException, CaContainer

## CaContainer GetProcessList(void) const

Owning Class	CaContext
Description	Retrieve a CaContainer object that contains the attributes for each Harvest Process in the list.
Arguments	None.
Return Value	Returns a CaContainer object.
Exceptions	Throws CaException on error.
Notes	List attributes are stored with the following hash keys:  <i>CA_ATTRKEY_NAME, CA_ATTRKEY_OBJID, CA_ATTRKEY_PROJECTID, CA_ATTRKEY_STATEID, CA_ATTRKEY_PROCTYPE, CA_ATTRKEY_EXECUTEACCESS</i>

### Example

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

See Also [HSDK Programming Fundamentals](#), [CaException](#), [CaContainer](#)

### **bool SetProject(LPCTSTR szProjectName)**

Owning Class [CaContext](#)

Description Set the specified Project as the current context Project.

Arguments LPCTSTR szProjectName

Return Value true if success, false if fails.

Example

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

See Also [HSDK Programming Fundamentals](#), [CaException](#), [GetProjectName](#), [GetProject](#), [CaHarvest](#)

### **bool SetProject(HAROBJID iObjId)**

Owning Class [CaContext](#)

Description Set the specified Project as the current context Project.

Arguments HAROBJID iObjId

Return Value true if success, false if fails.

Example

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

See Also [HSDK Programming Fundamentals](#), [CaException](#), [GetProjectName](#), [GetProject](#), [CaHarvest](#)

### **bool SetState(LPCTSTR szStateName)**

Owning Class [CaContext](#)

Description Set the specified State as the current context State.

Arguments LPCTSTR szStateName

Return Value true if success, false if fails.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException, GetStateName, GetState, GetStateId

**bool SetState(HAROBJID iObjId)****Owning Class**

CaContext

**Description**

Set the specified State as the current context State.

**Arguments**

HAROBJID iObjId

**Return Value**

true if success, false if fails.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException, GetStateName, GetState, GetStateId

**bool SetCheckout(LPCTSTR szCheckoutName)****Owning Class**

CaContext

**Description**

Set the specified Checkout as the current context Checkout.

**Arguments**

LPCTSTR szCheckoutName

**Return Value**

true if success, false if fails.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException, GetCheckoutName, GetCheckout, GetCheckoutId

**bool SetCheckout(HAROBJID iObjId)****Owning Class**

CaContext

**Description**

Set the specified Checkout as the current context Checkout.

**Arguments**

HAROBJID iObjId

Return Value            true if success, false if fails.

Example

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

See Also                HSDK Programming Fundamentals, CaException, GetCheckoutName, GetCheckout, GetCheckoutId

## **bool SetCheckin(LPCTSTR szCheckinName)**

Owning Class           CaContext

Description            Set the specified Checkin as the current context Checkin.

Arguments              LPCTSTR szCheckinName

Return Value           true if success, false if fails.

Example

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

See Also                HSDK Programming Fundamentals, CaException, GetCheckinName, GetCheckin, GetCheckinId

## **bool SetCheckin(HAROBJID iObjId)**

Owning Class           CaContext

Description            Set the specified Checkin as the current context Checkin.

Arguments              HAROBJID iObjId

Return Value           true if success, false if fails.

Example

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

See Also                HSDK Programming Fundamentals, CaException, GetCheckinName, GetCheckin, GetCheckinId

## **bool SetPromote(LPCTSTR szPromoteName)**

Owning Class           CaContext

Description            Set the specified Promote as the current context Promote.

Arguments	LPCTSTR szPromoteName
Return Value	true if success, false if fails.
Example	<pre>rContext.SetPromote(szName);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetPromoteName, GetPromote, GetPromoteId

### **bool SetPromote(HAROBJID iObjId)**

Owning Class	CaContext
Description	Set the specified Promote as the current context Promote.
Arguments	HAROBJID iObjId
Return Value	true if success, false if fails.
Example	<pre>rContext.SetPromote(iObjId);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetPromoteName, GetPromote, GetPromoteId

### **bool SetDemote(LPCTSTR szDemoteName)**

Owning Class	CaContext
Description	Set the specified Demote as the current context Demote.
Arguments	LPCTSTR szDemoteName
Return Value	true if success, false if fails.
Example	<pre>rContext.SetDemote(szName);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetDemoteName, GetDemote, GetDemoteId

**bool SetDemote(HAROBJID iObjId)**

Owning Class	CaContext
Description	Set the specified Demote as the current context Demote.
Arguments	HAROBJID iObjId
Return Value	true if success, false if fails.
Example	<pre>rContext.SetDemote(iObjId);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetDemoteName, GetDemote, GetDemoteId

**bool SetCrossProjectMerge(LPCTSTR szCrossProjectMergeName)**

Owning Class	CaContext
Description	Set the specified CrossProjectMerge as the current context CrossProjectMerge.
Arguments	LPCTSTR szCrossProjectMerge
Return Value	true if success, false if fails.
Example	<pre>rContext.CrossProjectMerge(szName);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetCrossProjectMergeName, GetCrossProjectMerge

**bool SetCrossProjectMerge(HAROBJID iObjId)**

Owning Class	CaContext
Description	Set the specified CrossProjectMerge as the current context CrossProjectMerge.
Arguments	HAROBJID iObjId
Return Value	true if success, false if fails.
Example	<pre>rContext.CrossProjectMerge( iObjId );</pre>

See Also                      HSDK Programming Fundamentals, CaException, GetCrossProjectMergeName, GetCrossProjectMerge

### **bool SetTakeSnapshot(LPCTSTR szTakeSnapshotName)**

Owning Class                CaContext

Description                Set the specified TakeSnapshot as the current context TakeSnapshot.

Arguments                  LPCTSTR szTakeSnapshotName

Return Value                true if success, false if fails.

Example

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

See Also                      HSDK Programming Fundamentals, CaException, GetTakeSnapshotName, GetTakeSnapshot

### **bool SetTakeSnapshot(HAROBJID iObjId)**

Owning Class                CaContext

Description                Set the specified TakeSnapshot as the current context TakeSnapshot.

Arguments                  HAROBJID iObjId

Return Value                true if success, false if fails.

Example

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

See Also                      HSDK Programming Fundamentals, CaException, GetTakeSnapshotName, GetTakeSnapshot

### **bool SetUDP(LPCTSTR szUDPName)**

Owning Class                CaContext

Description                Set the specified UDP as the current context UDP.

Arguments                  LPCTSTR szUDPName

Return Value                true if success, false if fails.

### Example

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

See Also            HSDK Programming Fundamentals, CaException, GetUDPName, GetUDP

## **bool SetUDP(HAROBJID iObjId)**

Owning Class        CaContext

Description         Set the specified UDP as the current context UDP.

Arguments           HAROBJID iObjId

Return Value        true if success, false if fails.

### Example

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

See Also            HSDK Programming Fundamentals, CaException, GetUDPName, GetUDP

## **bool SetApprove(LPCTSTR szApproveName)**

Owning Class        CaContext

Description         Set the specified Approve as the current context Approve.

Arguments           LPCTSTR szApproveName

Return Value        true if success, false if fails.

### Example

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

See Also            HSDK Programming Fundamentals, CaException, GetApproveName, GetApprove, GetApproveId

## **bool SetApprove(HAROBJID iObjId)**

Owning Class        CaContext

Description         Set the specified Approve as the current context Approve.

Arguments           HAROBJID iObjId

Return Value        true if success, false if fails.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException, GetApproveName, GetApprove, GetApproveId

**void SetWebserverName( LPCTSTR szWebserverName )****Owning Class**

CaContext

**Description**

Store the address of the Harweb server in the current context.

**Arguments**

LPCTSTR szWebserverName

**Return Value**

None.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException, GetWebserverName

**bool SetCreatePackage(LPCTSTR szCreatePackageName)****Owning Class**

CaContext

**Description**

Set the specified CreatePackage as the current context CreatePackage.

**Arguments**

LPCTSTR szCreatePackageName

**Return Value**

true if success, false if fails.

**Example**

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

**See Also**

HSDK Programming Fundamentals, CaException, GetCreatePackageName, GetCreatePackage, GetCreatePackageId

**bool SetCreatePackage(HAROBJID iObjId)****Owning Class**

CaContext

**Description**

Set the specified CreatePackage as the current context CreatePackage.

**Arguments**

HAROBJID iObjId

Return Value            true if success, false if fails.

Example

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

See Also                HSDK Programming Fundamentals, CaException, GetCreatePackageName, GetCreatePackage, GetCreatePackageId

## **bool SetDeleteVersion(LPCTSTR szDeleteVersionName)**

Owning Class            CaContext

Description             Set the specified DeleteVersion as the current context DeleteVersion.

Arguments               LPCTSTR szDeleteVersionName

Return Value            true if success, false if fails.

Example

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

See Also                HSDK Programming Fundamentals, CaException, GetDeleteVersionName, GetDeleteVersion, GetDeleteVersionId

## **bool SetDeleteVersion(HAROBJID iObjId)**

Owning Class            CaContext

Description             Set the specified DeleteVersion as the current context DeleteVersion.

Arguments               HAROBJID iObjId

Return Value            true if success, false if fails.

Example

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

See Also                HSDK Programming Fundamentals, CaException, GetDeleteVersionName, GetDeleteVersion, GetDeleteVersionId

## **bool SetRemoveItem(LPCTSTR szRemoveItemName)**

Owning Class            CaContext

Description             Set the specified RemoveItem as the current context RemoveItem.

Arguments	LPCTSTR szRemoveItemName
Return Value	true if success, false if fails.
Example	<pre>rContext.SetRemoveItem(szName);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetRemoveItemName, GetRemoveItem, GetRemoveItemId

### **bool SetRemoveItem(HAROBJID iObjId)**

Owning Class	CaContext
Description	Set the specified RemoveItem as the current context RemoveItem.
Arguments	HAROBJID iObjId
Return Value	true if success, false if fails.
Example	<pre>rContext.SetRemoveItem(iObjId);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetRemoveItemName, GetRemoveItem, GetRemoveItemId

### **bool SetConcurrentMerge(LPCTSTR szConcurrentMergeName)**

Owning Class	CaContext
Description	Set the specified ConcurrentMerge as the current context ConcurrentMerge.
Arguments	LPCTSTR szConcurrentMergeName
Return Value	true if success, false if fails.
Example	<pre>rContext.SetConcurrentMerge(szName);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetConcurrentMergeName, GetConcurrentMerge, GetConcurrentMergeId

**bool SetConcurrentMerge(HAROBJID iObjId)**

Owning Class	CaContext
Description	Set the specified ConcurrentMerge as the current context ConcurrentMerge.
Arguments	HAROBJID iObjId
Return Value	true if success, false if fails.
Example	<pre>rContext.SetConcurrentMerge(iObjId);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetConcurrentMergeName, GetConcurrentMerge, GetConcurrentMergeId

**bool SetPackage(LPCTSTR szPackageName)**

Owning Class	CaContext
Description	Set the specified Package as the current context Package.
Arguments	LPCTSTR szPackageName
Return Value	true if success, false if fails.
Example	<pre>rContext.SetPackage(szName);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetPackageName, GetPackage, GetPackageId

**bool SetPackage(HAROBJID iObjId)**

Owning Class	CaContext
Description	Set the specified Package as the current context Package.
Arguments	HAROBJID iObjId
Return Value	true if success, false if fails.
Example	<pre>rContext.SetPackage(iObjId);</pre>

See Also                    HSDK Programming Fundamentals, CaException, GetPackageName, GetPackage, GetPackageId

### **bool SetSnapshotView(LPCTSTR szSnapshotViewName)**

Owning Class	CaContext
Description	Set the specified SnapshotView as the current context SnapshotView.
Arguments	LPCTSTR szSnapshotViewName
Return Value	true if success, false if fails.
Example	<pre>rContext.SetSnapshotView(szName);</pre>
See Also	HSDK Programming Fundamentals, CaException, GetSnapshotViewName, GetSnapshotViewId

### **bool SetSnapshotView( HAROBJID iObjid )**

Owning Class	CaContext
Description	Set the specified SnapshotView as the current context SnapshotView.
Arguments	HAROBJID iObjid
Return Value	true if success, false if fails.
Example	<pre>rContext.SetSnapshotView( iObjId );</pre>
See Also	HSDK Programming Fundamentals, CaException, GetSnapshotViewName, GetSnapshotViewId

### **void SetClientPath(LPCTSTR szClientPath)**

Owning Class	CaContext
Description	Set the specified Client Path as the current context Client Path.
Arguments	LPCTSTR szClientPath
Return Value	None.

Exceptions                      Throws CaException on error.

Example

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

See Also                      HSDK Programming Fundamentals, CaException, GetClientPath

### **void SetViewPath(LPCTSTR szViewPath)**

Owning Class                  CaContext

Description                   Set the specified View Path as the current context View Path.

Arguments                   LPCTSTR szViewPath

Return Value                  None.

Exceptions                   Throws CaException on error.

Example

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

See Also                      HSDK Programming Fundamentals, CaException, GetViewPath

### **bool IsPackageSet()**

Owning Class                  CaContext

Description                   Determines if the context currently has a package set.

Arguments                   None.

Return Value                  bool - true if package set, false if not set.

Exceptions                   Throws CaException on error.

Example

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

See Also                      HSDK Programming Fundamentals, CaException, SetPackage, GetPackage

## CaCreatePackage

Description	<p>CaCreatePackage encapsulates the Harvest 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 Harvest administrator sets default attribute values.</p> <p>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.</p> <p>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.</p> <p>Usage: Set the Project, State, and CreatePackage Context. Set the CreatePackage attributes. Invoke the Execute function.</p>
Public Methods	<p>SetPackageName</p> <p>SetAssignedUserName</p> <p>SetDescription</p> <p>SetPriority</p> <p>GetPackageName</p> <p>GetAssignedUserName</p> <p>GetPriority</p> <p>GetDescription</p> <p>GetDefaultName</p> <p>Execute</p>
See Also	<p>HSDK Programming Fundamentals, CaContext</p>

**void SetPackageName(LPCTSTR szPackageName)**

Owning Class	CaCreatePackage
Description	Set a name for the new Package.
Arguments	LPCTSTR szPackageName.
Return Value	None.
Notes	If the package name attribute is set then it is used without regard to the default name expression.
Example	<pre>CaCreatePackage&amp; rCreatePackage = rContext.GetCreatePackage();rCreatePackage. SetPackageName("Package - 12");</pre>
See Also	HSDK Programming Fundamentals, GetPackageName

**void SetAssignedUserName( LPCTSTR szAssignedUserName)**

Owning Class	CaCreatePackage
Description	Assign a user to the new Package.
Arguments	LPCTSTR szAssignedUserName – username for the assigned user
Return Value	None.
Notes	None.
Example	<pre>CaCreatePackage&amp; rCreatePackage = rContext.GetCreatePackage();rCreatePackage. SetAssignedUserName ("George");</pre>
See Also	HSDK Programming Fundamentals, GetAssignedUserName

**void SetDescription(LPCTSTR szDescription)**

Owning Class            CaCreatePackage

Description            Sets a description for the new package.

Arguments              LPCTSTR szDescription.

Return Value           None.

Notes                   None.

Example

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

See Also                HSDK Programming Fundamentals, GetDescription

**void SetPriority(HINT32 iPriority)**

Owning Class            CaCreatePackage

Description            Sets the Package priority attribute.

Arguments              HINT32 iPriority - Priorities value.

Return Value           None.

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

Example

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

See Also                HSDK Programming Fundamentals, GetPriority

## LPCTSTR GetDefaultName (void)

Owning Class	CaCreatePackage
Description	Retrieves the default name expression for the new package.
Arguments	None.
Return Value	LPCTSTR – The name expression.
Notes	The Harvest Administrator sets the default name expression as an attribute of the Harvest 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.
Example	<pre>CaCreatePackage&amp; rCreatePackage = rContext.GetCreatePackage();CaString szName = rCreatePackage. GetDefaultName();</pre>
See Also	HSDK Programming Fundamentals, SetPackageName, GetDefaultName

## LPCTSTR GetPackageName(void)

Owning Class	CaCreatePackage
Description	Retrieves the name of the new package.
Arguments	None.
Return Value	LPCTSTR – The name.
Notes	None.
Example	<pre>CaCreatePackage&amp; rCreatePackage = rContext.GetCreatePackage();CaString szName = rCreatePackage. GetPackageName(true);</pre>
See Also	HSDK Programming Fundamentals, SetPackageName, GetDefaultName

## LPCTSTR GetAssignedUserName(void)

Owning Class            CaCreatePackage

Description            Retrieves the current value of the Assigned User Name attribute.

Arguments              None.

Return Value           LPCTSTR – current Assigned User Name value.

Notes                   None.

Example

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

See Also                HSDK Programming Fundamentals, SetItemName

## LPCTSTR GetDescription(void)

Owning Class            CaCreatePackage

Description            Retrieves the current value of the description attribute.

Arguments              None.

Return Value           LPCTSTR – current description value.

Notes                   None.

Example

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

See Also                HSDK Programming Fundamentals, SetDescription

## HINT32 GetPriority(void)

Owning Class	CaCreatePackage
Description	Retrieves the current value of the Priority attribute.
Arguments	None.
Return Value	HINT32– current Priority value.
Notes	None.
Example	<pre>CaCreatePackage&amp; rCreatePackage = rContext.GetCreatePackage();HINT32 iPriority = rCreatePackage.GetPriority();</pre>
See Also	HSDK Programming Fundamentals, SetPriority

## HINT32 Execute(void)

Owning Class	CaCreatePackage
Description	Executes the CreatePackage process on the Package list currently selected using the Package Chooser.
Arguments	None.
Return Value	HINT32 – zero if success. Non-zero if failure.
Notes	None.
Example	<pre>CaCreatePackage&amp; rCreatePackage = rContext.GetCreatePackage(); HINT32 iResult = rCreatePackage.Execute();</pre>
See Also	HSDK Programming Fundamentals, CaContext

## CaCrossProjectMerge

### Description

CaCrossProjectMerge encapsulates the Harvest 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 Harvest 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

See Also      HSDK Programming Fundamentals, CaContext

### **void SetFromState(LPCTSTR szProjectName, LPCTSTR szStateName)**

Owning Class      CaCrossPackageMerge

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

Arguments	<p>LPCTSTR szProjectName – name of the project to merge with.</p> <p>LPCTSTR szStateName – name of a state in szProjectName that contains the desired packages.</p>
Return Value	None.
Notes	A list of available projects may be retrieved from a CaHarvest object using GetProjectList(). From that list, select a project by name and extract it's 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.
Example	<pre>CaCrossProjectMerge&amp; xprojmerge = CaContext.GetCrossProjectMerge(); xprojmerge.SetFromState( szProject, szState );</pre>
See Also	HSDK Programming Fundamentals, CaHarvest, GetProjectList, GetStateList

### **void SetFromState(HAROBJID szProjectId, HAROBJID szStateId)**

Owning Class	CaCrossPackageMerge
Description	Set the Project and State that contain the source package(s) to merge with. These must differ from the Project & State set in the Context.
Arguments	<p>LPCTSTR szProjectName – name of the project to merge with.</p> <p>LPCTSTR szStateName – name of a state in szProjectName that contains the desired packages.</p>
Return Value	None.
Notes	A list of available projects may be retrieved from a CaHarvest object using GetProjectList(). From that list, select a project by name and extract it's 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.
Example	<pre>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&amp; xprojmerge = CaContext.GetCrossProjectMerge(); xprojmerge.SetFromState( iProject, iState);</pre>
See Also	HSDK Programming Fundamentals, CaHarvest, GetProjectList, GetStateList

**void SetMergeRule(LPCTSTR szRule)**

Owning Class	CaCrossPackageMerge
Description	Sets Rule for how to merge the versions of files in the packages.
Arguments	LPCTSTR szRule – The rule value.
Return Value	None.
Notes	<p>The following are the allowable values for the Merge Rule. They are defined in an include file built into the project.</p> <p>MERGE_RULE_TRUNK_MERGE – Take the entire trunk version and ignore the version on the branch. This corresponds to the GUI TakeTrunk Version option.</p> <p>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.</p> <p>MERGE_RULE_CONFLICT_TAG – Create a merge tag only if changes in branch and trunk conflict. This corresponds to the GUI Merge Aggressively option.</p> <p>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.</p>
Example	<pre>CaCrossProjectMerge&amp; xprojmrg = rContext.GetCrossProjectMerge(); xprojmrg.SetMergeRule("MERGE_RULE_CONFLICT_TAG");</pre>
See Also	HSDK Programming Fundamentals, GetMergeRule

**void SetPlacement(LPCTSTR szPlacement)**

Owning Class	CaCrossPackageMerge
Description	Sets Rule for how the merged version appears in the destination state.
Arguments	LPCTSTR szRule – The rule value.
Return Value	None.
Notes	<p>The following are the allowable values for the Placement. They are defined in an include file built into the project.</p>

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.

#### Example

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

See Also                      HSDK Programming Fundamentals, GetPlacement

### **void SetVersionDescription(LPCTSTR szDescription)**

Owning Class	CaCrossPackageMerge
Description	Sets text description of merged version.
Arguments	LPCTSTR szDescription – the description itself
Return Value	None.
Notes	None.

#### Example

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

See Also                      HSDK Programming Fundamentals, GetVersionDescription

### **HAROBJID GetFromProjectId(void)**

Owning Class	CaCrossPackageMerge
Description	Returns the ObjID of the Project set by SetFromState.
Arguments	None.
Return Value	HAROBJID – Project Id containing source state & packages

Notes None.

Example

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

See Also HSDK Programming Fundamentals, SetFromState, CaHarvest

## LPCTSTR GetFromProjectName(void)

Owning Class CaCrossPackageMerge

Description Returns the Name of the Project set by SetFromState.

Arguments None.

Return Value LPCTSTR – Project Name containing source state & packages

Notes None.

Example

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

See Also HSDK Programming Fundamentals, SetFromState, CaHarvest

## HAROBJID GetFromStateId(void)

Owning Class CaCrossPackageMerge

Description Returns the ObjID of the State set by SetFromState.

Arguments None.

Return Value HAROBJID – State Id containing source state & packages

Notes None.

Example

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

See Also HSDK Programming Fundamentals, SetFromState, CaHarvest

## LPCTSTR GetFromStateName(void)

Owning Class	CaCrossPackageMerge
Description	Returns the Name of the State set by SetFromState.
Arguments	None.
Return Value	LPCTSTR – State Name containing source state & packages
Notes	None.
Example	<pre>CaCrossProjectMerge&amp; xprojmrg = rContext.GetCrossProjectMerge(); LPCTSTR szFromState = xprojmrg.GetFromStateName();</pre>
See Also	HSDK Programming Fundamentals, SetFromState, CaHarvest

## LPCTSTR GetMergeRule(void)

Owning Class	CaCrossPackageMerge
Description	Gets Rule for how to merge the versions of files in the packages.
Arguments	None.
Return Value	LPCTSTR szRule – The rule value.
Notes	<p>The following are the allowable values for the Merge Rule. They are defined in an include file built into the project.</p> <p>MERGE_RULE_TRUNK_MERGE – Take the entire trunk version and ignore the version on the branch. This corresponds to the GUI TakeTrunk Version option.</p> <p>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.</p> <p>MERGE_RULE_CONFLICT_TAG – Create a merge tag only if changes in branch and trunk conflict. This corresponds to the GUI Merge Aggressively option.</p> <p>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.</p>
Example	<pre>CaCrossProjectMerge&amp; xprojmrg = rContext.GetCrossProjectMerge(); LPCTSTR szMergeRule = xprojmrg.GetMergeRule();</pre>

See Also                      HSDK Programming Fundamentals, SetMergeRule

## **LPCTSTR GetPlacement(void)**

Owning Class                CaCrossPackageMerge

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

Arguments                 None.

Return Value              LPCTSTR szRule - The rule value.

Notes                      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.

Example

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

See Also                      HSDK Programming Fundamentals, SetPlacement

## **LPCTSTR GetVersionDescription(void)**

Owning Class                CaCrossPackageMerge

Description                Gets text description of merged version.

Arguments                 None.

Return Value              LPCTSTR szDescription - the description itself.

Notes                      None.

**Example**

```
CaCrossProjectMerge& xprojmrg = rContext.GetCrossProjectMerge(); LPCTSTR  
szDescription = xprojmrg.GetDescription();
```

See Also                HSDK Programming Fundamentals, [GetVersionDescription](#)

**HINT32 Execute(void)**

Owning Class            CaCrossPackageMerge

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

Arguments              None.

Return Value            HINT32 – zero if success. Non-zero 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 CaPackageChooser will search with CaPackageChooser::SetProjectObjId( iProject ) and CaPackageChooser::SetStateObjId( iState ), and then CaPackageChooser::Execute (false) to allow searching outside of context.

**Example**

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

See Also                HSDK Programming Fundamentals, [CaContext](#), [CaPackageChooser](#), [SetProjectObjId](#), [SetStateId](#), [CaHarvest](#), [GetProjectList](#), [GetStateList](#)

## CaDeleteVersion

Description	<p>CaDeleteVersion encapsulates the Harvest 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 Harvest administrator sets default attribute values.</p> <p>Usage: Set the Project, State, and DeleteVersion Context. Select a set Versions to delete using the CaVersionChooser object. Invoke the Execute function.</p>
Public Methods	Execute
See Also	HSDK Programming Fundamentals, CaContext, CaVersionChooser

### HINT32 Execute(void)

Owning Class	CaDeleteVersion
Description	Executes the DeleteVersion process on the Package list currently selected using the Package Chooser.
Arguments	None.
Return Value	HINT32 - zero if success. Non-zero if failure.
Notes	None.
Example	<pre>CaDeleteVersion&amp; rDeleteVersion = rContext.GetDeleteVersion(); HINT32 iResult = rDeleteVersion.Execute();</pre>
See Also	HSDK Programming Fundamentals, CaContext, CaVersionChooser

## CaDemote

Description	<p>CaDemote encapsulates the Harvest 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 Harvest administrator sets default attribute values.</p> <p>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.</p>
Public Methods	<p>SetCheckPackageDependencies</p> <p>SetCheckPackageGroupBind</p> <p>GetCheckPackageDependencies</p> <p>GetCheckPackageGroupBind</p> <p>Execute</p>
See Also	HSDK Programming Fundamentals, CaContext, CaPackageChooser

### **bool SetCheckPackageDependencies(bool bCheckPackageDependencies)**

Owning Class	CaDemote
Description	Sets Check Package Dependencies flag.
Arguments	bool bCheckPackageDependencies – true if verify dependencies prior to Demote, false if not.
Return Value	bool – true if set succeeded, false if failed.
Notes	None.
Example	<pre>CaDemote&amp; rDemote = rContext.GetDemote(); rDemote.SetCheckPackageDependencies(true);</pre>
See Also	HSDK Programming Fundamentals, GetCheckPackageDependencies

## **bool SetCheckPackageGroupBind( bool bFlag)**

Owning Class	CaDemote
Description	Specifies whether or not packages in a package group must all be demoted together.
Arguments	bool bFlag – true if yes, false if no.
Return Value	bool – true if set succeeded, false if failed.
Notes	None.
Example	<pre>CaDemote&amp; rDemote = rContext.GetDemote(); bool bFlag = rDemote. SetCheckPackageGroupBind(true);</pre>
See Also	HSDK Programming Fundamentals, GetCheckPackageGroupBind

## **bool GetCheckPackageDependencies (void)**

Owning Class	CaDemote
Description	Gets the current value of the Check Package Dependencies flag.
Arguments	None.
Return Value	bool – current value of flag.
Notes	None.
Example	<pre>CaDemote&amp; rDemote = rContext.GetDemote(); CaString = rDemote. GetCheckPackageDependencies ();</pre>
See Also	HSDK Programming Fundamentals, SetCheckPackageDependencies

**bool GetCheckPackageGroupBind (void)**

Owning Class	CaDemote
Description	Gets the current value of the Check Package Group Bind flag.
Arguments	None.
Return Value	bool – current value of flag.
Notes	None.
Example	<pre>CaDemote&amp; rDemote = rContext.GetDemote(); CaString = rDemote. GetCheckPackageGroupBind ();</pre>
See Also	HSDK Programming Fundamentals, SetCheckPackageGroupBind

**HINT32 Execute(void)**

Owning Class	CaDemote
Description	Executes the Demote process on the Package list currently selected using the Package Chooser.
Arguments	None.
Return Value	HINT32 – zero if success. Non-zero if failure.
Notes	None.
Example	<pre>CaDemote&amp; rDemote = rContext.GetDemote(); HINT32 iResult = rDemote.Execute();</pre>
See Also	HSDK Programming Fundamentals, CaContext, CaPackageChooser

## CaException

### Description

CaException provides keyed message generation and logging facilities. It uses the underlying Harvest 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 Harvest 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

### See Also

HSDK Programming Fundamentals, CaException, CaHarvest, CaContainer

## CaException(bool bKey, LPCTSTR szMessage )

Owning Class	CaException
Description	Constructs an exception object. The bKey argument determines whether szMessage is treated as a message key or as simple message text.
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.
Notes	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.
Example	<pre>CaException ex(false, "message text");</pre>
See Also	HSDK Programming Fundamentals, CaContainer

## CaException( const CaException &CRight)

Owning Class	CaException
Description	Copy constructor. Duplicates one exception object into another.
Arguments	const CaException &Cright – The object to be copied
Return Value	None.
Example	<pre>CaException ex(false, "message text"); CaException excopy(ex);</pre>
See Also	HSDK Programming Fundamentals, CaContainer

## CaException(LPCTSTR szKey, ...)

Owning Class	CaException
Description	Constructs an exception object with the message indexed by the specified key.
Arguments	<p>LPCTSTR szKey – The message key.</p> <p>... - 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.</p>
Return Value	None.
Notes	If the message key is invalid then the object is created with a message-key error message.
Example	
See Also	<pre>CaException ex(HSDKKEYVALUE, 1, "arg2", "arg3", 4, NULL);</pre> <p>HSDK Programming Fundamentals, CaContainer</p>

## void SetReturnCode(HINT32 iReturnCode)

Owning Class	CaException
Description	Sets an optional return code into the exception object.
Arguments	HINT32 iReturnCode – The value.
Return Value	None.
Example	
See Also	<pre>CaException ex(HSDKKEYVALUE, 1, "arg2", "arg3", 4, NULL); ex.SetReturnCode(54);</pre> <p>HSDK Programming Fundamentals</p>

### **HINT32 GetReturnCode(void)**

Owning Class            CaException

Description            Retrieves the optional return code from the exception object.

Arguments              None.

Return Value           Code value.

Example

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

See Also                HSDK Programming Fundamentals

### **CaString GetMessage(void)**

Owning Class            CaException

Description            Retrieves the full formatted message from the exception object.

Arguments              None.

Return Value           Message value.

Example

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

See Also                HSDK Programming Fundamentals, CaString

### **void LogMessage(void)**

Owning Class            CaException

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

Arguments              None.

Return Value           None.

Example

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

See Also                HSDK Programming Fundamentals, CaHarvest

## CaHarvest

Description	CaHarvest is the top level HSDK object that provides the Harvest 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	<div>CaHarvest</div> <div>Login</div> <div>Logout</div> <div>GetLastMessage</div> <div>ChangePassword</div> <div>GetProjectList</div> <div>GetActiveProjectList</div> <div>GetInactiveProjectList</div> <div>GetStateList</div> <div>GetUserList</div> <div>SetLog</div> <div>SetStaticLog</div> <div>GetLog</div> <div>GetContext</div> <div>GetBrokerName</div>
See Also	HSDK Programming Fundamentals, CaException, CaContext, CaContainer

## CaHarvest(void)

Owning Class	CaHarvest
Description	This constructor instantiates a CaHarvest.
Arguments	None.
Return Value	None.
Exceptions	Throws CaException on error.
Example	<pre>CaHarvest Harvest;</pre>
See Also	HSDK Programming Fundamentals, CaException, SetBroker, GetBrokerName

## CaHarvest(LPCTSTR szBrokerName)

Owning Class	CaHarvest
Description	This constructor instantiates a CaHarvest object and specifies a Harvest service broker.
Arguments	LPCTSTR szBrokerName – Name of the Harvest service broker for connections.
Return Value	None
Exceptions	Throws CaException on error.
Example	<pre>CaHarvest Harvest("MyBrokerName");</pre>
See Also	HSDK Programming Fundamentals, CaException, SetBroker, GetBrokerName

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

Owning Class	CaHarvest
Description	Logs into the specified Harvest Broker.
Arguments	LPCTSTR szBrokerName – The name of the Harvest Broker.  LPCTSTR szUserName – The User Name.  LPCTSTR szPassword – The Password.
Return Value	Returns zero if success, non-zero if failure or password policy invocation.
Notes	If password has expired, the return code will be non-zero and no valid context will be available. However, when password policy allows the changing of an expired password, the ChangePassword method will work. Once the password is successfully changed, a valid context and session are available. GetLastMessage will retrieve the message from the server for why login failed
Example	<pre>CaHarvest Harvest; if(Harvest.Login("CMBroker", "CMUser", "MyPassword")) throw CaException(false, "Bad Login");</pre>
See Also	HSDK Programming Fundamentals, CaException, SetBroker, Logout, GetLastMessage, ChangePassword

## HINT32 Login(LPCTSTR szUserName, LPCTSTR szPassword)

Owning Class	CaHarvest
Description	Logs into the Harvest Broker specified in constructor.
Arguments	LPCTSTR szUserName – The User Name.  LPCTSTR szPassword – The Password.
Return Value	Returns zero if success, non-zero if failure.
Notes	If password has expired, the return code will be non-zero and no valid context will be available. However, when password policy allows the changing of an expired password, the ChangePassword method will work. Once the password is successfully changed, a valid context and session are available. GetLastMessage will retrieve the message from the server for why login failed
Example	

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

See Also            HSDK Programming Fundamentals, CaException, SetBroker, Logout,  
GetLastMessage, ChangePassword

## LPCTSTR GetLastMessage( )

Owning Class	CaHarvest
Description	Returns the last message from the server for Login or ChangePassword requests..
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;
```

See Also            HSDK Programming Fundamentals, CaException, SetBroker, Login, Logout,  
ChangePassword

## HINT32 ChangePassword(LPCTSTR szPassword)

Owning Class	CaHarvest
Description	Changes the Harvest password of the currently logged in user.
Arguments	LPCTSTR szPassword – The Password.
Return Value	Returns zero if success, non-zero if failure.
Notes	If new password does not match local password policy, a non-zero result code will be returned. Check with GetLastMessage to see why changing the password failed.

### Example

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

See Also            HSDK Programming Fundamentals, CaException, SetBroker, Login, Logout,  
GetLastMessage

**void Logout(void)**

Owning Class	CaHarvest
Description	Log out from the Harvest Broker. Clear the context.
Arguments	None.
Return Value	None.
Example	<pre>Harvest.Logout();</pre>
See Also	HSDK Programming Fundamentals, Login

**LPCTSTR GetBrokerName(void) const**

Owning Class	CaHarvest
Description	the GetBrokerName function retrieves the name of the current Harvest service broker.
Arguments	None.
Return Value	Returns Broker name, empty string if not set.
Example	<pre>CaString szName = Harvest.GetBrokerName();</pre>
See Also	HSDK Programming Fundamentals, SetBroker

## CaContext& GetContext(void) const

Owning Class	CaHarvest
Description	Retrieve the current context object.
Arguments	None.
Return Value	Returns a reference to the context object.
Exceptions	Throws CaException on error.
Example	<pre>CaContext&amp; rContext = Harvest.GetContext();</pre>
See Also	HSDK Programming Fundamentals, CaException, CaContext

## CaContainer GetProjectList(void) const

Owning Class	CaHarvest
Description	Retrieve a CaContainer object that contains the attributes for each Harvest Project in the list.
Arguments	None.
Return Value	Returns a CaContainer object.
Exceptions	Throws CaException on error.
Notes	List attributes are stored with the following hash keys:  <i>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</i>
Example	<pre>CaContainer Container = Harvest.GetProjectList();</pre>
See Also	HSDK Programming Fundamentals, CaException, CaContainer

---

## CaContainer GetActiveProjectList(void) const

Owning Class	CaHarvest
Description	Retrieve a CaContainer object that contains the attributes for each active Harvest Project in the list.
Arguments	None.
Return Value	Returns a CaContainer object.
Exceptions	Throws CaException on error.
Notes	List attributes are stored with the following hash keys:  <i>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</i>
Example	<pre>CaContainer Container = Harvest.GetActiveProjectList ();</pre>
See Also	HSDK Programming Fundamentals, CaException, CaContainer

## CaContainer GetInactiveProjectList(void) const

Owning Class	CaHarvest
Description	Retrieve a CaContainer object that contains the attributes for each inactive Harvest Project in the list.
Arguments	None.
Return Value	Returns a CaContainer object.
Exceptions	Throws CaException on error.
Notes	List attributes are stored with the following hash keys:  <i>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</i>
Example	<pre>CaContainer Container = Harvest.GetInactiveProjectList ();</pre>

See Also                      HSDK Programming Fundamentals, CaException, CaContainer

## CaContainer GetStateList( HAROBJID iProject ) const

Owning Class	CaHarvest
Description	Retrieve a CaContainer object that contains the attributes for each Harvest State in the Harvest Project specified by the HAROBJID iProject argument.
Arguments	HAROBJID iProject – ObjId of the desired project that contains this statelist.
Return Value	Returns a CaContainer object.
Exceptions	Throws CaException on error.
Notes	List attributes are stored with the following hash keys:  <i>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</i>

### Example

```
CaContainer Container = Harvest.GetStateList ( iProject );
```

## CaContainer GetUserList(void) const

Owning Class	CaHarvest
Description	Retrieve a CaContainer object that contains the attributes for each Harvest User in the list.
Arguments	None.
Return Value	Returns a CaContainer object.
Exceptions	Throws CaException on error.
Notes	List attributes are stored with the following hash keys:  <i>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</i>

### Example

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

See Also            HSDK Programming Fundamentals, CaException, CaContainer

## **bool SetLog(ostream &rLogStream)**

Owning Class	CaHarvest
Description	the SetLog function sets the specified ostream as the default log stream for the Harvest program, both HSDK and HAPI instances.
Arguments	ostream &rLogStream – reference to log stream.
Return Value	Returns true if success, false if fail.
Example	<pre>ofstream log("HarvestLog.txt") bool bResult = Harvest.SetLog(log); if(!bResult) throw exception;</pre>
See Also	HSDK Programming Fundamentals, SetLog, GetLog

## **static void SetStaticLog(ostream &rLogStream)**

Owning Class	CaHarvest
Description	the SetStaticLog function sets the specified ostream as the default static log stream for the HSDK program but not for the HAPI instance.
Arguments	ostream &rLogStream – reference to log stream.
Return Value	None.
Example	<pre>ofstream log("HarvestLog.txt")         Harvest.SetStaticLog(log);</pre>
See Also	HSDK Programming Fundamentals, SetLog, GetLog

## **static ostream& GetLog(void)**

Owning Class	CaHarvest
Description	the static GetLog function retrieves the static log ostream.
Arguments	None.
Return Value	Returns reference to the static log ostream.

Example

```
ofstream &rLog = Harvest.GetLog(log);  
if(!rLog) throw exception;
```

See Also

[HSDK Programming Fundamentals](#), [SetLog](#), [GetLog](#)

## CaPackageChooser

Description	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

Execute

See Also HSDK Programming Fundamentals, CaContext, CaContainer, CaCheckin, CaCheckout, CaDeleteVersion

## **void Clear(void)**

Owning Class CaPackageChooser

Description Removes all selection criteria.

Arguments None.

Return Value None.

Notes None.

Example

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

See Also HSDK Programming Fundamentals, CaContext

## **void SetPackageName(LPCTSTR szName, int iIndex = 0)**

Owning Class CaPackageChooser

Description Set a name expression for search by name pattern.

Arguments LPCTSTR szName - Name expression to search for package names.

int iIndex - Index value for selecting multiple package names.

Return Value None.

Notes None.

Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetPackageName(?QA: Test?, 0); rPackageChooser.SetPackageName(?QA:  
Document?, 1);
```

See Also HSDK Programming Fundamentals, CaContext

**void SetPackageObjId(HAROBJID iObjId, int iIndex = 0)**

Owning Class	CaPackageChooser
Description	Set a name expression for search by name pattern.
Arguments	HAROBJID iObjId - Object Id to search for package with matching ids.  int iIndex - Index value for selecting multiple package ids.
Return Value	None.
Notes	None.
Example	<pre>CaPackageChooser&amp; rPackageChooser = rContext.GetPackageChooser(); rPackageChooser.SetPackageObjId(101, 0);rPackageChooser.SetPackageObjId(202, 1);</pre>
See Also	HSDK Programming Fundamentals, CaContext

**void SetProjectObjId(HAROBJID iObjId)**

Owning Class	CaPackageChooser
Description	Sets Project Object Id if not searching within current context.
Arguments	HAROBJID iObjId - Project Object Id.
Return Value	None.
Notes	None.
Example	<pre>CaPackageChooser&amp; rPackageChooser = rContext.GetPackageChooser(); rPackageChooser.SetProjectObjId(10024);</pre>
See Also	HSDK Programming Fundamentals, CaContext, Execute

**void SetStateObjId (HAROBJID iObjId)**

Owning Class	CaPackageChooser
Description	Sets State Object Id if not searching within current context.
Arguments	HAROBJID iObjId – State Id.
Return Value	None.
Notes	None.
Example	<pre>CaPackageChooser&amp; rPackageChooser = rContext.GetPackageChooser(); rPackageChooser.SetStateObjId(10024);</pre>
See Also	HSDK Programming Fundamentals, CaContext, Execute

**void SetPkgGroupObjId(HAROBJID iObjId)**

Owning Class	CaPackageChooser
Description	Set Package Group Object Id.
Arguments	HAROBJID iObjId – Package Group Id.
Return Value	None.
Notes	None.
Example	<pre>CaPackageChooser&amp; rPackageChooser = rContext.GetPackageChooser(); rPackageChooser.SetPkgGroupObjId(10024);</pre>
See Also	HSDK Programming Fundamentals, CaContext

**void SetAssigneeObjId (HAROBJID iObjId)**

Owning Class	CaPackageChooser
Description	Set Package Assignee Object Id.
Arguments	HAROBJID iObjId – Assignee Id.
Return Value	None.
Notes	None.

### Example

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetAssigneeObjId(10024);
```

## **void SetFromDate(CaTimeStamp& Time)**

Owning Class	CaPackageChooser
Description	Sets the From Date value of the date range.
Arguments	CaTimeStamp& Time – the time stamp value.
Return Value	None.
Notes	None.

### Example

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

See Also            HSDK Programming Fundamentals, CaContext, CaTimeStamp, SetToDate

## **void SetToDate(CaTimeStamp& Time)**

Owning Class	CaPackageChooser
Description	Sets the To Date value of the date range.
Arguments	CaTimeStamp& Time – the time stamp value.
Return Value	None.
Notes	None.

### Example

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

See Also            HSDK Programming Fundamentals, CaContext, CaTimeStamp, SetFromDate

**void SetCreatorObjId (HAROBJID iObjId)**

Owning Class	CaPackageChooser
Description	Set Package Creator Object Id.
Arguments	HAROBJID iObjId – Creator Id.
Return Value	None.
Notes	None.

**Example**

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetCreatorObjId(10024);
```

**void SetModifierObjId (HAROBJID iObjId)**

Owning Class	CaPackageChooser
Description	Set Package Modifier Object Id.
Arguments	HAROBJID iObjId – Most recent Modifier Id.
Return Value	None.
Notes	None.

**Example**

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetModifierObjId(10024);
```

**void SetApproveUserObjId (HAROBJID iObjId)**

Owning Class	CaPackageChooser
Description	Set Package Approve User Object Id.
Arguments	HAROBJID iObjId – Id of user who has approved packages.
Return Value	None.
Notes	None.

**Example**

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetApproveUserObjId(10024);
```

**void SetPriority(HINT32 iPriority)**

Owning Class	CaPackageChooser
Description	Set Package Priority.
Arguments	HINT32 iPriority – User defined priority value to search for.
Return Value	None.
Notes	None.

**Example**

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetPriority(1);
```

**void SetSubString(LPCTSTR szString)**

Owning Class	CaPackageChooser
Description	Set substring expression for searching the Note attribute of packages.
Arguments	LPCTSTR szString – A substring expression.
Return Value	None.
Notes	None.

**Example**

```
CaPackageChooser& rPackageChooser = rContext.GetPackageChooser();  
rPackageChooser.SetSubString("*required for Version 1.5*");
```

**void SetPkgStatus(HINT32 iPkgStatus)**

Owning Class	CaPackageChooser
Description	Sets the Package Status Option to search by Package approval status.
Arguments	HINT32 iPkgStatus – Package Approval Status attribute.
Return Value	None.
Notes	<p>The following are the allowable values for the Package Status Option. They are defined in an include file built into the project.</p> <p>APPROVE_STATUS_APPROVED – Get only approved packages.</p>

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.

#### Example

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

See Also                   HSDK Programming Fundamentals, CaContext

### **void SetPkgEntryFromDate(CaTimeStamp& Time)**

Owning Class	CaPackageChooser
Description	Sets the State Entry From Date value of the date range.
Arguments	CaTimeStamp& Time – the time stamp value.
Return Value	None.
Notes	None.

#### Example

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

See Also                   HSDK Programming Fundamentals, CaContext, CaTimeStamp, SetPkgEntryToDate

### **void SetPkgEntryToDate(CaTimeStamp& Time)**

Owning Class	CaPackageChooser
Description	Sets the State Entry To Date value of the date range.
Arguments	CaTimeStamp& Time – the time stamp value.
Return Value	None.
Notes	None.

#### Example

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

See Also                    HSDK Programming Fundamentals, CaContext, CaTimeStamp,  
SetPkgEntryFromDate

## CaContainer GetPackageList(void) const

Owning Class	CaPackageChooser
Description	Retrieve a CaContainer object that contains the attributes for each Harvest Package in the list.
Arguments	None.
Return Value	Returns a CaContainer object.
Exceptions	Throws CaException on 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

```
CaContainer Container = rPackageChooser.GetPackageList();
```

See Also                    HSDK Programming Fundamentals, CaException, CaContainer

**HINT32 Execute(bool bUseCurrentContext)**

Owning Class	CaPackageChooser
Description	Executes the PackageChooser based on the set of package selection criteria set in the package chooser attributes.
Arguments	bool bUseCurrentContext – If true restrict package search to current context, if false use Project Id and State Id if set.
Return Value	HINT32 – zero if success, non-zero if failure.
Notes	Default value is true.
Example	<pre>CaPackageChooser&amp; rPackageChooser = rContext.GetPackageChooser(); HINT32 iResult = rPackageChooser.Execute(false);</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetProjectObjId, SetStateObjId
Notes	None.
Example	<pre>CaStringList d; cout &lt;&lt; d;</pre>
See Also	HUtilities Programming Fundamentals

## CaPromote

Description	<p>CaPromote encapsulates the Harvest 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 Harvest administrator sets default attribute values.</p> <p>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.</p>
Public Methods	<p>SetMergedPkgsOnly</p> <p>SetCheckPackageDependencies</p> <p>SetCheckPackageGroupBind</p> <p>GetMergedPkgsOnly</p> <p>GetCheckPackageDependencies</p> <p>GetCheckPackageGroupBind</p> <p>Execute</p>
See Also	HSDK Programming Fundamentals, CaContext, CaPackageChooser

### **bool SetMergedPkgsOnly(bool bFlag)**

Owning Class	CaPromote
Description	Sets the Merged Packages Only flag.
Arguments	bool bFlag – If 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.
Notes	None.
Example	<pre>CaPromote&amp; rPromote = rContext.GetPromote(); rPromote.SetMergedPkgsOnly(true);</pre>
See Also	HSDK Programming Fundamentals, GetMergedPkgsOnly

**bool SetCheckPackageDependencies(bool bCheckPackageDependencies)**

Owning Class	CaPromote
Description	Sets Check Package Dependencies flag.
Arguments	bool bCheckPackageDependencies – true if verify dependencies prior to promote, false if not.
Return Value	bool – true if set succeeded, false if failed.
Notes	None.
Example	<pre>CaPromote&amp; rPromote = rContext.GetPromote(); rPromote. SetCheckPackageDependencies (true);</pre>
See Also	HSDK Programming Fundamentals, GetCheckPackageDependencies

**bool SetCheckPackageGroupBind( bool bFlag)**

Owning Class	CaPromote
Description	Specifies whether or not packages in a package group must all be promoted together.
Arguments	bool bFlag – true if yes, false if no.
Return Value	bool – true if set succeeded, false if failed.
Notes	None.
Example	<pre>CaPromote&amp; rPromote = rContext.GetPromote(); bool bFlag = rPromote. SetCheckPackageGroupBind(true);</pre>
See Also	HSDK Programming Fundamentals, GetCheckPackageGroupBind

**bool GetMergedPkgsOnly(void)**

Owning Class	CaPromote
Description	Gets the current value of the Merged Packages Only flag.
Arguments	None.
Return Value	bool – current value of flag.
Notes	None.
Example	<pre>CaPromote&amp; rPromote = rContext.GetPromote(); CaString = rPromote. GetMergedPkgsOnly();</pre>
See Also	HSDK Programming Fundamentals, SetMergedPkgsOnly

**bool GetCheckPackageDependencies (void)**

Owning Class	CaPromote
Description	Gets the current value of the Check Package Dependencies flag.
Arguments	None.
Return Value	bool – current value of flag.
Notes	None.
Example	<pre>CaPromote&amp; rPromote = rContext.GetPromote(); CaString = rPromote. GetCheckPackageDependencies ();</pre>
See Also	HSDK Programming Fundamentals, SetCheckPackageDependencies

**bool GetCheckPackageGroupBind (void)**

Owning Class	CaPromote
Description	Gets the current value of the Check Package Group Bind flag.
Arguments	None.
Return Value	bool – current value of flag.
Notes	None.
Example	<pre>CaPromote&amp; rPromote = rContext.GetPromote(); CaString = rPromote. GetCheckPackageGroupBind ();</pre>
See Also	HSDK Programming Fundamentals, SetCheckPackageGroupBind

**HINT32 Execute(void)**

Owning Class	CaPromote
Description	Executes the Promote process on the Package list currently selected using the Package Chooser.
Arguments	None.
Return Value	HINT32 – zero if success. Non-zero if failure.
Notes	None.
Example	<pre>CaPromote&amp; rPromote = rContext.GetPromote(); HINT32 iResult = rPromote.Execute();</pre>
See Also	HSDK Programming Fundamentals, CaContext, CaPackageChooser

## CaRemoveItem

Description	<p>CaRemoveItem encapsulates the Harvest 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 Harvest 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.</p> <p>Usage: Set the Project, State, and RemoveItem Context. Set the RemoveItem attributes, including the item to be removed. Invoke the Execute function.</p>
Public Methods	<p>SetDescription</p> <p>SetItemObjId</p> <p>SetItemName</p> <p>SetItemPath</p> <p>SetRecursive</p> <p>GetDescription</p> <p>GetItemObjId</p> <p>GetItemName</p> <p>GetItemPath</p> <p>GetRecursive</p> <p>Execute</p>
See Also	<p>HSDK Programming Fundamentals, CaContext</p>

**bool SetDescription(LPCTSTR szDesc)**

Owning Class	CaRemoveItem
Description	Sets a description to annotate why the item is being removed.
Arguments	LPCTSTR szDesc – The explanation.
Return Value	bool – true if set succeeded, false if failed.
Notes	None.
Example	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem(); rRemoveItem. SetDescription("Obsolete");</pre>
See Also	HSDK Programming Fundamentals, GetDescription

**void SetItemId(HAROBJID itemId)**

Owning Class	CaRemoveItem
Description	Identifies the item to be removed by its object id.
Arguments	HAROBJID itemId.
Return Value	None.
Notes	Takes precedence over name. If present then name, path, and recursive are ignored.
Example	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem(); rRemoveItem. SetItemId(12);</pre>
See Also	HSDK Programming Fundamentals, GetItemId, SetItemName

**void SetItemName(LPCTSTR szItemName)**

Owning Class	CaRemoveItem
Description	Identifies the item to be removed by its name.
Arguments	LPCTSTR szItemName.
Return Value	None.
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	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem();rRemoveItem. SetItemName("Item1.txt");</pre>
See Also	HSDK Programming Fundamentals, GetItemName, SetItemObjId, SetItemPath, SetRecursive

**void SetItemPath(LPCTSTR szItemPath)**

Owning Class	CaRemoveItem
Description	Sets the view path of the item to be removed.
Arguments	LPCTSTR szItemPath – a valid view path.
Return Value	None.
Notes	None.
Example	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem();rRemoveItem. SetItemPath("\\rep\\code");</pre>
See Also	HSDK Programming Fundamentals, SetItemName, SetItemObjId, GetItemPath, SetRecursive

**void SetRecursive(bool bRecursive)**

Owning Class	CaRemoveItem
Description	Specifies whether or not to look recursively below the current item path for the item.
Arguments	bool bRecursive – true if yes, false if no.
Return Value	None.
Notes	Used in conjunction with SetItemName and SetItemPath.
Example	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem();rRemoveItem. SetRecursive(true);</pre>
See Also	HSDK Programming Fundamentals, SetItemName, SetItemObjId, SetItemPath, GetRecursive

**LPCTSTR GetDescription(void)**

Owning Class	CaRemoveItem
Description	Retrieves the current value of the description attribute.
Arguments	None.
Return Value	LPCTSTR – current description value.
Notes	None.
Example	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem();CaString szDesc = rRemoveItem. GetDescription();</pre>
See Also	HSDK Programming Fundamentals, SetDescription

## **HAROBJID GetItemObjId(void)**

Owning Class            CaRemoveItem

Description            Retrieves the current value of the item object id attribute.

Arguments              None.

Return Value           HAROBJID – minus one if not set otherwise the current id.

Notes                   None.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();HAROBJID iItemId =  
rRemoveItem. GetItemObjId();
```

See Also                HSDK Programming Fundamentals, SetItemObjId

## **LPCTSTR GetItemName(void)**

Owning Class            CaRemoveItem

Description            Retrieves the current value of the Item Name attribute.

Arguments              None.

Return Value           LPCTSTR – current Item Name value.

Notes                   None.

Example

```
CaRemoveItem& rRemoveItem = rContext.GetRemoveItem();CaString szName =  
rRemoveItem. GetItemName ();
```

See Also                HSDK Programming Fundamentals, SetItemName

## LPCTSTR GetItemPath (void)

Owning Class	CaRemoveItem
Description	Retrieves the current value of the Item Path attribute.
Arguments	None.
Return Value	LPCTSTR – current Item Path value.
Notes	None.
Example	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem();CaString szName = rRemoveItem. GetItemPath ();</pre>
See Also	HSDK Programming Fundamentals, SetItemPath

## bool GetRecursive(void)

Owning Class	CaRemoveItem
Description	Retrieves the current value of the Recursive attribute.
Arguments	None.
Return Value	bool – current Recursive value.
Notes	None.
Example	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem();bool bRecursive = rRemoveItem. GetRecursive();</pre>
See Also	HSDK Programming Fundamentals, SetRecursive

## HINT32 Execute(void)

Owning Class	CaRemoveItem
Description	Executes the RemoveItem process on the Package list currently selected using the Package Chooser.
Arguments	None.
Return Value	HINT32 - zero if success. Non-zero if failure.
Notes	None.
Example	<pre>CaRemoveItem&amp; rRemoveItem = rContext.GetRemoveItem(); HINT32 iResult = rRemoveItem.Execute();</pre>
See Also	HSDK Programming Fundamentals, CaContext, CaPackageChooser

## CaSQL

Description	<p>CaSQL encapsulates the Harvest ExecuteSQL process. It cannot be instantiated directly. The CaSQL object can only be retrieved from the CaContext, but there is not other setup necessary. The CaSQL object is instantiated when the CaContext is created.</p> <p>Usage: Set the SQL string value. Invoke the Execute function.</p>
Public Methods	<p>SetSQLStatement</p> <p>GetSQLResult</p> <p>Execute</p>
See Also	HSDK Programming Fundamentals, CaContext

## void SetSQLStatement( LPCTSTR szSQLStr )

Owning Class	CaSQL
Description	Sets the SQL text to execute in the Harvest database.
Arguments	LPCTSTR szSQLStr - The SQL statement.

Return Value           None.

Notes                   None.

Example

```
CaSQL& rSQL= rContext.GetSQL(); rSQL.SetSQLStatement("Select * from HARUSER");  
rSQL.Execute();
```

See Also               HSDK Programming Fundamentals, CaString

## CaContainer GetSQLResult( void )

Owning Class           CaSQL

Description           Retrieves the results from the execution of the SQL statement.

Arguments             None.

Return Value           CaContainer – Results

Notes                  None.

Example

```
CaSQL& rSQL= rContext.GetSQL(); rSQL.SetSQLStatement("Select * from HARUSER");  
rSQL.Execute(); CaContainer sqldata = rSQL.GetSQLResult();
```

See Also               HSDK Programming Fundamentals, CaContainer

## HINT32 Execute(void)

Owning Class           CaSQL

Description           Executes the SQL statement supplied by SetSQLStatement against the Harvest database.

Arguments             None.

Return Value           HINT32 – zero if success. Non-zero if failure.

Notes                  None.

Example

```
CaSQL& rSQL= rContext.GetSQL(); rSQL.SetSQLStatement("Select * from HARUSER");  
rSQL.Execute();
```

See Also               HSDK Programming Fundamentals, CaContext

## CaTakeSnapshot

**Description** CaTakeSnapshot encapsulates the Harvest 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 Harvest 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

**See Also** HSDK Programming Fundamentals, CaContext, CaTimeStamp, CaPackageChooser

## **bool SetCanViewExternallyFlag(bool bFlag)**

**Owning Class** CaTakeSnapshot

**Description** Sets the CanViewExternally flag true or false.

**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.

**Notes** The CanViewExternally flag defaults to true.

### Example

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

**See Also** HSDK Programming Fundamentals, GetCanViewExternallyFlag

**bool GetCanViewExternallyFlag(bool bFlag)**

Owning Class	CaTakeSnapshot
Description	Gets the CanViewExternally flag true or false.
Arguments	None
Return Value	bool bFlag – If true snapshot is visible to other projects, if false only current project has visibility.
Notes	The CanViewExternally flag defaults to true.
Example	<pre>CaTakeSnapshot&amp; rTakeSnapshot = rContext.GetTakeSnapshot(); bool bViewExternal = rTakeSnapshot.GetCanViewExternallyFlag(true);</pre>
See Also	HSDK Programming Fundamentals, SetCanViewExternallyFlag

**HINT32 Execute(LPCTSTR szViewName, CaTimeStamp snapshottime=NULL\_CATIMESTAMP, LPCTSTR szReferenceSnapshot = NULL)**

Owning Class	CaTakeSnapshot
Description	Executes the TakeSnapshot process on one of the following: <ul style="list-style-type: none"><li>■ The State selected in the CaContext.</li><li>■ The State selected in the CaContext at a given time.</li><li>■ The packages selected in the package chooser overlaid on a previous Snapshot view</li></ul>
Arguments	LPCTSTR szViewName, CaTimeStamp snapshottime, LPCTSTR szReferenceSnapshot
Return Value	HINT32 – zero if success. Non-zero if failure.
Notes	None.
Example	<pre>CaApprove&amp; rApprove = rContext.GetApprove(); HINT32 iResult = rApprove.Execute();</pre>
See Also	HSDK Programming Fundamentals, CaContext, CaPackageChooser

## CaUDP

**Description** CaUDP encapsulates the Harvest 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 Harvest 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

GetUDPTYPE

GetProgramName

GetDescription

GetUserCanModifyInput

GetUsePackageList

GetUseVersionList

GetAddCmdLineArg

GetInput

Execute

**See Also** HSDK Programming Fundamentals, CaContext

### **bool SetUsePackageList(bool bUsePackageChooser)**

**Owning Class** CaUDP

**Description** Sets a boolean value for whether to pass the UDP process a Package List which would be expanded by the [PACKAGE] system variable.

Arguments	bool bUsePackageChooser – To use packages in package chooser or not.
Return Value	bool – What was set for this value.
Notes	<p>To pass a list of packages to a UDP process, this boolean is only part of the process.</p> <p>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.</p> <p>Second, this boolean must be set before Execute() is called.</p> <p>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.</p>
Example	<pre>CaUDP&amp; rUDP = rContext.GetUDP(); rUDP.SetUsePackageList(true); CaPackageChooser &amp;rPackageUDP = rContext.GetPackageChooser(); rPackageUDP.Clear(); rPackageUDP.SetPackageName( rContext.GetPackageName() ); rPackageUDP.Execute(); rUDP.Execute();</pre>
See Also	HSDK Programming Fundamentals, CaPackageChooser, GetUsePackageList

## **bool SetUseVersionList(bool bUseVersionChooser)**

Owning Class	CaUDP
Description	Sets a boolean value for whether to pass the UDP process a Version List which would be expanded by the [VERSION] system variable.
Arguments	bool bUseVersionChooser – To use versions in version chooser or not.
Return Value	bool – What was set for this value.
Notes	<p>To pass a list of versions to a UDP process, this boolean is only part of the process.</p> <p>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.</p> <p>Second, this boolean must be set before Execute() is called.</p> <p>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.</p>

**Example**

```
CaUDP& rUDP = rContext.GetUDP(); rUDP.SetUseVersionList(true); CaVersionChooser
&rVersionUDP = rContext.GetVersionChooser(); rVersionUDP.Clear();
rVersionUDP.SetItemName( "*.cpp" ); rVesrionUDP.Execute(); rUDP.Execute();
```

**See Also**

HSDK Programming Fundamentals, CaVersionChooser, GetUserVersionList

**void SetInput(LPCTSTR szAddInput)****Owning Class**

CaUDP

**Description**

Sets the string that will be piped to stdin for the UDP process.

**Arguments**

LPCTSTR szAddInput – The string to be passed to stdin.

**Return Value**

None.

**Notes**

Permission must be given by Harvest 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**

```
CaUDP& rUDP = rContext.GetUDP(); if( rUDP.GetUserCanModifyInput() == true )
rUDP.SetInput("Input string"); rUDP.Execute();
```

**See Also**

HSDK Programming Fundamentals, GetUserCanModifyInput, GetInput

**void SetAddCmdLineArg(LPCTSTR szCmdLineArg)****Owning Class**

CaUDP

**Description**

Sets additional command line arguments to be passed to the UDP executing process.

**Arguments**

LPCTSTR szCmdLineArg – 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 whitespace character.

**Example**

```
CaUDP& rUDP = rContext.GetUDP(); rUDP.SetAddCmdLineArg("-v -h"); rUDP.Execute();
```

**See Also**

HSDK Programming Fundamentals, [GetAddCmdLineArg](#)

**LPCTSTR GetUDPType(void)****Owning Class**

CaUDP

**Description**

Gets the string describing the type of UDP that this process defines.

**Arguments**

None.

**Return Value**

LPCTSTR – string description of UDP type.

**Notes**

There are two possible types of UDP:

Client – Client side UDP execution (local machine)

Server – Server side UDP execution (Harvest server)

**Example**

```
CaUDP& rUDP = rContext.GetUDP(); LPCTSTR szUDPTYPE = rUDP.GetUDPTYPE();
```

**See Also**

HSDK Programming Fundamentals

**LPCTSTR GetProgramName(void)****Owning Class**

CaUDP

**Description**

Gets the complete program name and argument for the UDP process specified by the Harvest Administrator.

**Arguments**

None.

**Return Value**

LPCTSTR – string containing command and arguments for UDP process

**Notes**

None.

**Example**

```
CaUDP& rUDP = rContext.GetUDP(); LPCTSTR szProgramName = rUDP.GetProgramName();
```

**See Also**

HSDK Programming Fundamentals

## LPCTSTR GetDescription(void)

Owning Class	CaUDP
Description	Returns the description for the UDP process set by the Harvest Administrator.
Arguments	LPCTSTR szDescription – the description for this process.
Return Value	None.
Notes	None.
Example	<pre>CaUDP&amp; rUDP = rContext.GetUDP(); LPCTSTR szDescription = rUDP.GetDescription();</pre>
See Also	HSDK Programming Fundamentals

## bool GetUserCanModifyInput(void)

Owning Class	CaUDP
Description	Returns the boolean value which specifies whether “Secure Input” was set for the UDP process by the Harvest Administrator.
Arguments	bool – whether input is user modifiable.
Return Value	None.
Notes	<p>true – Using SetInput(), the user can replace the default values for the string that is passed to stdin of the executed UDP.</p> <p>false – User cannot modify input, using SetInput() will fail.</p>
Example	<pre>CaUDP&amp; rUDP = rContext.GetUDP(); if( rUDP.GetUserCanModifyInput() == true ) rUDP.SetInput(“Input String”);</pre>
See Also	HSDK Programming Fundamentals, SetInput, GetInput

## bool GetUsePackageList(void)

Owning Class	CaUDP
Description	Returns the current boolean setting for whether on not the packages in the PackageChooser will be sent to the UDP process during Execute().

Arguments	None.
Return Value	bool – the current setting for using the package list
Notes	<p>To pass a list of packages to a UDP process, this boolean is only part of the process.</p> <p>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.</p> <p>Second, this boolean must be set before Execute() is called.</p> <p>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 desire package list.</p>
Example	<pre>CaUDP&amp; rUDP = rContext.GetUDP(); bool usepackage = rUDP.GetUsePackageList();</pre>
See Also	HSDK Programming Fundamentals, SetUsePackageList, CaPackageChooser

## **bool GetUseVersionList(void)**

Owning Class	CaUDP
Description	Returns the current boolean setting for whether on not the versions in the VersionChooser will be sent to the UDP process during Execute().
Arguments	None.
Return Value	bool – the current setting for using the version list
Notes	<p>To pass a list of versions to a UDP process, this boolean is only part of the process.</p> <p>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.</p> <p>Second, this boolean must be set before Execute() is called.</p> <p>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 desire version list.</p>
Example	

```
CaUDP& rUDP = rContext.GetUDP(); bool useversion = rUDP.GetUseVersionList();
```

See Also                      HSDK Programming Fundamentals, SetUsePackageList, CaVersionChooser

## LPCTSTR GetAddCmdLineArg(void)

Owning Class	CaUDP
Description	Retrieves the additional command line arguments that will be passed to the executing UDP.
Arguments	None.
Return Value	LPCTSTR – string of arguments.
Notes	<p>SetAddCmdLineArg appends the arguments specified to the entire UDP command. Any arguments specified by the Administrator during the setup of the UDP are retained.</p> <p>Arguments that are specified are prepended with a white space character.</p>

### Example

```
CaUDP& rUDP = rContext.GetUDP(); LPCTSTR szCmdArg = rUDP.GetAddCmdLineArg();
```

See Also                      HSDK Programming Fundamentals, SetAddCmdLineArg

## LPCTSTR GetInput(void)

Owning Class	CaUDP
Description	Returns the current string that will be passed to the stdin of the process on execution.
Arguments	None.
Return Value	LPCTSTR – string that will be passed to input of executing process.
Notes	<p>Permission must be given by Harvest Administer to allow user-modifiable input. See GetUserCanModifyInput() to determine if the Administrator has secured input.</p> <p>GetInput returns the default string specified by the Harvest administrator if not yet modified by SetInput(); otherwise returns the current value for the input.</p> <p>Output from the process will be sent to the CaHarvest log.</p>

### Example

```
CaUDP& rUDP = rContext.GetUDP(); LPCTSTR szInput = rUDP.GetInput();
```

See Also            HSDK Programming Fundamentals, SetInput, CaHarvest, GetLog

## HINT32 Execute(void)

Owning Class	CaUDP
Description	Executes the UDP process.
Arguments	None.
Return Value	HINT32 – zero if success. Non-zero if failure.
Notes	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 Harvest object log.

### Example

```
CaUDP& rUDP = rContext.GetUDP(); HINT32 iResult = rUDP.Execute();
```

See Also            HSDK Programming Fundamentals, CaHarvest, GetLog, CaPackageChooser, CaVersionChooser

# CaVersionChooser

## Description

CaVersionChooser cannot be instantiated directly. The CaVersionChooser object can only be retrieved from the CaContext after.

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.

Usage: Set the Project and State Context. If the State is a Snapshot State then set a Snapshot View in the current context. If the State is not a Shapshot 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  
SetItemObjId  
SetItemName  
SetItemPackageObjId  
SetPackageObjId  
SetRecursive  
SetBranchOption  
SetVersionNumber  
SetVersionOption  
SetVersionItemOption  
SetVersionStatusOption  
SetVersionDateOption  
SetFromDate  
SetToDate  
GetVersionList  
Execute

See Also      HSDK Programming Fundamentals, CaContext, CaContainer, CaCheckin, CaCheckout, CaDeleteVersion

### **void Clear(void)**

Owning Class	CaVersionChooser
Description	Removes all selection criteria.
Arguments	None.
Return Value	None.
Notes	None.

### Example

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

See Also            HSDK Programming Fundamentals, CaContext

## **void SetParentPath(LPCTSTR szParentPath)**

Owning Class            CaVersionChooser

Description            Removes all selection criteria.

Arguments              LPCTSTR szParentPath – Parent repository view path to begin search.

Return Value            None.

Notes                    None.

### Example

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetParentPath(“\\rep1\\mydir”);
```

See Also            HSDK Programming Fundamentals, CaContext, SetRecursive

## **void SetItemId(HAROBJID iItemObjId, int iIndex)**

Owning Class            CaVersionChooser

Description            Sets one in a set of Item Object Ids whose versions will be searched for.

Arguments              HAROBJID iItemObjId – Item Object Id.

int iIndex – 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

```
CaVersionChooser& rVersionChooser = rContext.GetVersionChooser();  
rVersionChooser.SetItemId(10024, 1);
```

See Also            HSDK Programming Fundamentals, CaContext

**void SetItemName(LPCTSTR szItemName)**

Owning Class	CaVersionChooser
Description	Sets a name expression to search for versions.
Arguments	LPCTSTR szItemName - a name expression.
Return Value	None.
Notes	None.
Example	<pre>CaVersionChooser&amp; rVersionChooser = rContext.GetVersionChooser(); rVersionChooser.SetItemName("ca*.h");</pre>
See Also	HSDK Programming Fundamentals, CaContext

**void SetItemPackageObjId(HAROBJID iPackageObjId, int iIndex)**

Owning Class	CaVersionChooser
Description	Sets one in a set of Package Object Ids whose items will be searched for, and returns the latest version in the view.
Arguments	<p>HAROBJID iPackageObjId - Package Id whose items to search for.</p> <p>int iIndex - The set of ids is stored in an indexed array. Defaults to zero.</p>
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	<pre>CaVersionChooser&amp; rVersionChooser = rContext.GetVersionChooser(); rVersionChooser.SetItemPackageObjId(10024, 1);</pre>
See Also	HSDK Programming Fundamentals, CaContext

**void SetPackageObjId(HAROBJID iPackageObjId, int iIndex)**

Owning Class	CaVersionChooser
Description	Sets one in a set of Package Object Ids whose versions will be searched for.

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.
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	<pre>CaVersionChooser&amp; rVersionChooser = rContext.GetVersionChooser(); rVersionChooser.SetPackageObjId(10024, 1);</pre>
See Also	HSDK Programming Fundamentals, CaContext

### **void SetRecursive(bool bFlag)**

Owning Class	CaVersionChooser
Description	Sets recursive flag to search below the parent path.
Arguments	bool bFlag – true if recursive, false if no.
Return Value	None.
Notes	None.
Example	<pre>CaVersionChooser&amp; rVersionChooser = rContext.GetVersionChooser(); rVersionChooser.SetItemName("ca*.h");</pre>
See Also	HSDK Programming Fundamentals, CaContext, SetItemName

### **void SetBranchOption(HINT32 iOptions)**

Owning Class	CaVersionChooser
Description	Sets the Branch option to specify whether trunk versions, branch versions, or both are to be retrieved.
Arguments	HINT32 iOption – The Branch Option.
Return Value	None.

Notes	<p>The following are the allowable values for the Branch Option. They are defined in an include file built into the project.</p> <p>BRANCH_FILTER_TRUNK_ONLY - Only get trunk versions.</p> <p>BRANCH_FILTER_BRANCH_ONLY- Only get branch versions.</p> <p>BRANCH_FILTER_UNMERGED_ONLY- Only get unmerged branches.</p> <p>BRANCH_FILTER_MERGED_ONLY- Only get merged branches.</p> <p>BRANCH_FILTER_TRUNK_AND_BRANCH- Get trunk and branched versions.</p>
Example	<pre>CaVersionChooser&amp; rVersionChooser = rContext.GetVersionChooser(); rVersionChooser.SetBranchOption(BRANCH_FILTER_TRUNK_AND_BRANCH);</pre>
See Also	HSDK Programming Fundamentals, CaContext

### **void SetVersionOption(HINT32 iOptions)**

Owning Class	CaVersionChooser
Description	Sets the Version Option to specify whether to get only the latest or multiple versions of the same item.
Arguments	HINT32 iOption - The Version Option.
Return Value	None.
Notes	<p>The following are the allowable values for the Version Option. They are defined in an include file built into the project.</p> <p>VERSION_FILTER_ALL_IN_VIEW- Get all version of an item in the view.</p> <p>VERSION_FILTER_LATEST_IN_VIEW- Get only the latest version of an item in the view.</p>
Example	<pre>CaVersionChooser&amp; rVersionChooser = rContext.GetVersionChooser(); rVersionChooser.SetVersionOption(VERSION_FILTER_LATEST_IN_VIEW);</pre>
See Also	HSDK Programming Fundamentals, CaContext

### **void SetVersionItemOption(HINT32 iOptions)**

Owning Class	CaVersionChooser
--------------	------------------

Description	Sets the Version Item Option to specify whether to base versions of an item, modified versions of an item, or both.
Arguments	HINT32 iOption - The Version Item Option.
Return Value	None.
Notes	<p>The following are the allowable values for the Version Item Option. They are defined in an include file built into the project.</p> <p>VERSION_FILTER_ITEM_MODIFIED- Get only modified versions of an item in the view.</p> <p>VERSION_FILTER_ITEM_BASELINE- Get only base versions of an item in the view.</p> <p>VERSION_FILTER_ITEM_BOTH- Get both base and modified versions of an item in the view.</p>
Example	<pre>CaVersionChooser&amp; rVersionChooser = rContext.GetVersionChooser(); rVersionChooser.SetVersionItemOption(VERSION_FILTER_ITEM_BOTH);</pre>
See Also	HSDK Programming Fundamentals, CaContext

### **void SetVersionNumber( LPCTSTR szVersion )**

Owning Class	CaVersionChooser
Description	Sets the version number to be selected by version chooser.
Arguments	LPCTSTR szVersion - The version number as a string.
Return Value	None.
Notes	Must be used with SetVersionOption(VERSION_FILTER_ALL_IN_VIEW).
Example	<pre>CaVersionChooser&amp; rVerChoose= rContext.GetVersionChooser(); rVerChoose.SetVersionNumber("3.1.1");</pre>
See Also	HSDK Programming Fundamentals, SetVersionOption.

### **void SetVersionStatusOption(HINT32 iOptions)**

Owning Class	CaVersionChooser
--------------	------------------

Description	Sets the Version Status Option to specify what types of versions to retrieve.
Arguments	HINT32 iOption – The Version Status Option.
Return Value	None.
Notes	<p>The following are the allowable values for the Version Status Option. They are defined in an include file built into the project.</p> <p>VERSION_FILTER_NORMAL_VERSION– Get only versions with a Normal tag.</p> <p>VERSION_FILTER_RESERVED_VERSION– Get only versions with a Reserve tag.</p> <p>VERSION_FILTER_MERGED_VERSION– Get only versions with a Merge tag.</p> <p>VERSION_FILTER_DELETED_VERSION– Get only versions with a Delete tag.</p> <p>VERSION_FILTER_FROZEN_VERSION– Get only versions with a Frozen tag.</p> <p>VERSION_FILTER_ALL_TAG– Get versions with any tag.</p> <p>VERSION_FILTER_ANY_TAG– Get versions with any tag except a Normal tag.</p>
Example	<pre>CaVersionChooser&amp; rVersionChooser = rContext.GetVersionChooser(); rVersionChooser.SetVersionStatusOption(VERSION_FILTER_ALL_TAG);</pre>
See Also	HSDK Programming Fundamentals, CaContext

### **void SetVersionDateOption(HINT32 iOptions)**

Owning Class	CaVersionChooser
Description	Sets the Version Date Option to specify date ranges.
Arguments	HINT32 iOption – The Version Date Option.
Return Value	None.
Notes	<p>The following are the allowable values for the Version Date Option. They are defined in an include file built into the project.</p> <p>VERSION_OPTION_DATE_ALL– Don't filter on dates.</p> <p>VERSION_OPTION_DATE_BETWEEN– Get only versions created between the from and to dates.</p>

**Example**

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

See Also            HSDK Programming Fundamentals, CaContext, SetFromDate, SetToDate

**void SetFromDate(CaTimeStamp& Time)**

Owning Class        CaVersionChooser

Description         Sets the From Date value of the date range.

Arguments           CaTimeStamp& Time – the time stamp value.

Return Value        None.

Notes                Only used if the Version Date Option is set.

**Example**

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

See Also            HSDK Programming Fundamentals, CaContext, CaTimeStamp,  
SetVersionDateOption, SetToDate

**void SetToDate(CaTimeStamp& Time)**

Owning Class        CaVersionChooser

Description         Sets the To Date value of the date range.

Arguments           CaTimeStamp& Time – the time stamp value.

Return Value        None.

Notes                Only used if the Version Date Option is set.

**Example**

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

See Also            HSDK Programming Fundamentals, CaContext, CaTimeStamp,  
SetVersionDateOption, SetFromDate

## CaContainer GetVersionList(void) const

Owning Class	CaVersionChooser
Description	Retrieve a CaContainer object that contains the attributes for each Harvest Version in the list.
Arguments	None.
Return Value	Returns a CaContainer object.
Exceptions	Throws CaException on error.
Notes	List attributes are stored with the following hash keys:  <i>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_DESCRIPTION</i>
Example	<pre>CaContainer Container = rVersionChooser.GetVersionList ();</pre>
See Also	HSDK Programming Fundamentals, CaException, CaContainer

## HINT32 Execute(void)

Owning Class	CaVersionChooser
Description	Executes the VersionChooser based on the set of version selection criteria set in the version chooser attributes.
Arguments	None.
Return Value	HINT32 - zero if success, non-zero if failure.
Notes	None.
Example	<pre>CaVersionChooser&amp; rVersionChooser = rContext.GetVersionChooser(); HINT32 iResult = rVersionChooser.Execute();</pre>

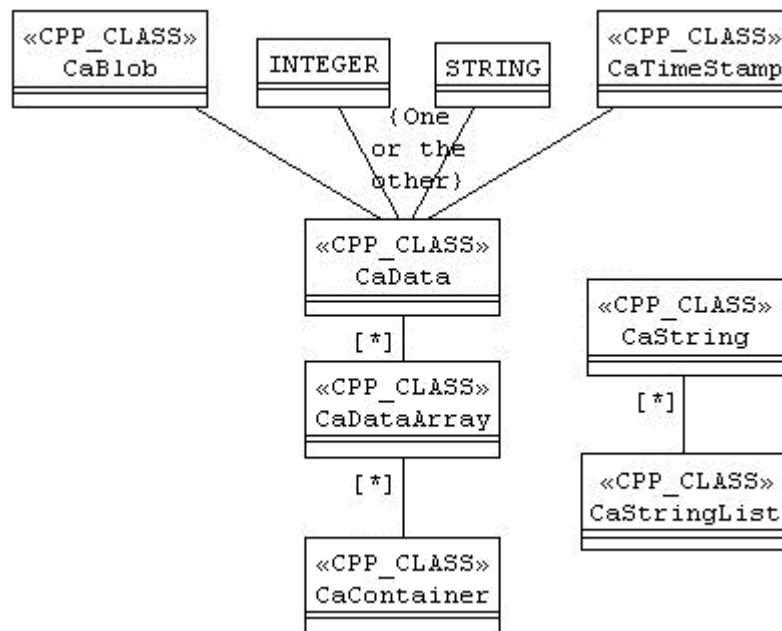
See Also                      HSDK Programming Fundamentals, CaContext

# HUtilities Class Descriptions

Each Harvest HUtilities class is described in this chapter, including syntax and example.

## HUtilities Programming Fundamentals

Hutilities is a lower-level Harvest module. It defines a set of cross-platform classes that encapsulate data and data containers used throughout Harvest. 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.



See Also

HSDK Programming Fundamentals

## CaBlob

### Description

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<<

### See Also

None.

## **void CaBlob(const CaBlob& rBlob)**

### Owning Class

CaBlob

### Description

Copy constructor sets one blob to another. They both point to the same data.

### Arguments

const CaBlob& rBlob – the blob to copy.

### Return Value

None.

### Notes

Blob does not own this data.

### Example

```
CaBlob b1(b2);
```

### See Also

HUtilities Programming Fundamentals

## **void CaBlob(HINT32 nSize, void \* lpData)**

### Owning Class

CaBlob

Description	Constructor.
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	<pre>CaBlob b1(5, ptr);</pre>
See Also	HUtilities Programming Fundamentals

### **HINT32 GetSize(void) const**

Owning Class	CaBlob
Description	Retrieves size in bytes of data stored.
Arguments	None.
Return Value	HINT32 – number of bytes stored.
Notes	Blob does not own this data.
Example	<pre>CaBlob b1(5, ptr); HINT32 iSize = b1.GetSize();</pre>
See Also	HUtilities Programming Fundamentals, GetData

**void \* GetData(void) const**

Owning Class	CaBlob
Description	Retrieves the data pointer stored in the object.
Arguments	None.
Return Value	void* - pointer stored in blob
Notes	Blob does not own this data
Example	<pre>CaBlob b1(5, ptr); void* pData = b1.GetData();</pre>
See Also	HUtilities Programming Fundamentals, GetSize

**CaBlob& operator=( const CaBlob &blob )**

Owning Class	CaBlob
Description	Assignment operator. Copies one data object to another.
Arguments	const CaBlob &aData – the object to copy.
Return Value	CaBlob & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals

**bool operator==( const CaBlob &rSrc)**

Owning Class	CaBlob
Description	Compares the blob with the specified object.
Arguments	const CaBlob &rSrc.
Return Value	bool – true if equal, false if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

**bool operator!=( const CaBlob &rSrc)**

Owning Class	CaBlob
Description	Compares the blob with the specified object.
Arguments	const CaBlob &rSrc.
Return Value	bool – false if equal, true if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

**friend ostream& operator<<( ostream &o, const CaBlob &rBlob )**

Owning Class	CaBlob
Description	Writes a description of the contents onto an ostream.
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 debug purposes.
Example	<pre>CaBlob b(5, ptr); cout &lt;&lt; b;</pre>
See Also	HUtilities Programming Fundamentals

## CaContainer

Description	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	<div>Clear</div> <div>Props</div> <div>GetKeyCount</div> <div>GetKeyList</div> <div>GetKeyElementCount</div> <div>IsKeyExist</div> <div>IsEmpty</div> <div>FindData</div> <div>IsValid</div> <div>operator=</div> <div>operator+=</div> <div>operator&lt;&lt;</div>
See Also	CaDataArray, CaData, CaString

## **void Clear(void)**

Owning Class	CaContainer
Description	Empties the container.
Arguments	None.
Return Value	None.
Notes	None.
Example	<pre>CaContainer bag; bag.clear();</pre>
See Also	HUtilities Programming Fundamentals

## **CaData & Props( LPCTSTR strKey, HINT32 iIndex)**

Owning Class	CaContainer
Description	Retrieves a referent to an object within the container. The contents of the object can be read or altered.
Arguments	<p>LPCTSTR strKey - They key value of the CaDataArray to locate.</p> <p>HINT32 iIndex- the location within the CaDataArray where the CaData object resides.</p>
Return Value	CPtHData & - a referent to the located object.
Notes	<p>iIndex defaults to zero.</p> <p>A lower-level exception occurs if the key is not found or if the index is out of range.</p>
Example	<pre>CaContainer bag; CaString = (LPCTSTR)bag.Props("MyKey", 2);</pre>
See Also	HUtilities Programming Fundamentals, IsKeyExist, GetKeyElementCount, CaData

## **HUINT32 GetKeyCount( void ) const**

Owning Class	CaContainer
--------------	-------------

Description	Retrieves the quantity of hash-map keys currently in the container.
Arguments	None.
Return Value	HUINT32 – number of keys as an unsigned 32-bit integer.
Notes	None.
Example	<pre>CaContainer bag; HUINT32 iNumKeys = bag.GetKeyCount();</pre>
See Also	HUtilities Programming Fundamentals, IsKeyExist, GetKeyElementCount

### **CaStringList GetKeyList( void ) const**

Owning Class	CaContainer
Description	Retrieves a list of the hash-map keys currently in the container.
Arguments	None.
Return Value	CaStringList – The key list.
Notes	None.
Example	<pre>CaContainer bag; CaStringList list = bag.GetKeyList();</pre>
See Also	HUtilities Programming Fundamentals, CaStringList, CaString

## HUINT32 GetKeyElementCount(LPCTSTR strKey) const

Owning Class	CaContainer
Description	Retrieves the quantity of CaData objects in the CaDataArray indexed by the specified hash-map key value.
Arguments	LPCTSTR strKey – The key value to check.
Return Value	HUINT32 – number of CaData objects as an unsigned 32-bit integer.
Notes	None.
Example	<pre>CaContainer bag; HUINT32 iNumElements = bag.GetKeyElementCount("MyKey");</pre>
See Also	HUtilities Programming Fundamentals, IsKeyExist, CaDataArray, CaData

## bool IsKeyExist( LPCTSTR strKey ) const

Owning Class	CaContainer
Description	Checks for the existence of the specified key value.
Arguments	LPCTSTR strKey – The key value to check.
Return Value	bool – true if it exists, false if not.
Notes	None.
Example	<pre>CaContainer bag; bool bHaveKey = bag.IsKeyExist("MyKey");</pre>
See Also	HUtilities Programming Fundamentals, Props

## **bool IsEmpty( void ) const**

Owning Class	CaContainer
Description	Checks if the container is empty.
Arguments	None.
Return Value	bool – true if it empty, false if not.
Notes	None.
Example	<pre>CaContainer bag; bool bHaveKey = bag.IsEmpty();</pre>
See Also	HUtilities Programming Fundamentals

## **CaDataArray FindData(LPCTSTR strKey, LPCTSTR strValue) const**

Owning Class	CaContainer
Description	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.
Arguments	LPCTSTR strKey – Key value  LPCTSTR strValue – string to search for.
Return Value	CaDataArray – array of matching indexes.
Notes	None.
Example	<pre>CaContainer bag; CaDataArray a = bag.FindData(CA_ATTRKEY_NAME, (LPCTSTR)gszProjectName); HAROBJID iObjid = (HINT32)bag.Props(CA_ATTRKEY_OBJID,a[0]);</pre>
See Also	HUtilities Programming Fundamentals, Props, CaDataArray, CaData

## CaDataArray FindData(LPCTSTR strKey, HINT32 iValue) const

Owning Class	CaContainer
Description	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.
Arguments	LPCTSTR strKey – Key value  HINT32 iValue – Integer value to search for.
Return Value	CaDataArray – array of matching indexes.
Notes	None.
Example	<pre>CaContainer bag; CaDataArray a = bag.FindData(CA_ATTRKEY_PRIORITY, 1); HINT32 iPriority = (HINT32)bag.Props(CA_ATTRKEY_PRIORITY,a[0]);</pre>
See Also	HUtilities Programming Fundamentals, Props, CaDataArray, CaData

## CaContainer & operator=(const CaContainer &CRight)

Owning Class	CaContainer
Description	Clears the left-hand container and then copies all data from the right-hand container into it.
Arguments	CaContainer &Cright – reference to the right-hand container.
Return Value	CaContainer & - reference to the left-hand container.
Notes	None.
Example	<pre>CaContainer bag; CaContainer bag2; bag2.Props("mykey", 0) = 3; bag = bag2;</pre>
See Also	HUtilities Programming Fundamentals, Props, CaDataArray, CaData

## CaContainer & operator+=(const CaContainer &CRight)

Owning Class	CaContainer
Description	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.
Arguments	CaContainer &Cright – reference to the right-hand container.
Return Value	CaContainer & – reference to the left-hand container.
Notes	None.
Example	<pre>CaContainer bag; CaContainer bag2; bag2.Props("mykey", 0) = 3; bag += bag2;</pre>
See Also	HUtilities Programming Fundamentals, Props, CaDataArray, CaData

## friend ostream& operator<<( ostream &o, const CaContainer &c )

Owning Class	CaContainer
Description	Writes the key values and their associated data values onto an ostream.
Arguments	<p>ostream &amp;o – Ostream on which to write.</p> <p>CaContainer &amp;c – reference to a container.</p>
Return Value	ostream & – reference to the ostream.
Notes	This is normally used for debug purposes.
Example	<pre>CaContainer bag; bag.Props("mykey", 0) = 3; cout &lt;&lt; bag;</pre>
See Also	HUtilities Programming Fundamentals, Props, CaDataArray, CaData

## CaData

Description	CaData objects provide a type-safe container for LPCTSTR, CaTimeStamp, HINT32, and CaBlob 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.
Public Methods	<div>GetType</div> <div>IsNull</div> <div>SetNull</div> <div>Operator=</div> <div>Operator==</div> <div>Operator!=</div> <div>Operator HINT32()</div> <div>Operator LPCTSTR()</div> <div>Operator CaString()</div> <div>Operator (CaTimeStamp&amp;())</div> <div>Operator (CaBlob&amp;())</div> <div>operator&lt;&lt;</div>
See Also	CaString, CaTimeStamp, CaBlob

**void SetNull(void)**

Owning Class	CaData
Description	Sets the Data object's value to NULL but does not change its type.
Arguments	None.
Return Value	None.
Notes	None.
Example	<pre>CaData d; d.SetNull();</pre>
See Also	HUtilities Programming Fundamentals, CaString, CaTimeStamp, CaBlob, IsNull

**bool IsNull(void) const**

Owning Class	CaData
Description	Checks to see whether the value of the CaData object is NULL.
Arguments	None.
Return Value	bool – true if NULL, false if not.
Notes	None.
Example	<pre>CaData d; bool bResult = d.IsNull();</pre>
See Also	HUtilities Programming Fundamentals, CaString, CaTimeStamp, CaBlob, SetNull

## CaData::Types GetType(void) const

Owning Class	CaData
Description	Retrieves the data type of the object.
Arguments	None.
Return Value	CaData::Types – Data type of object
Notes	<p>CaData::Types is an enumeration that can have the following values:</p> <p>NO_TYPE – Value has not been set.</p> <p>INTEGER– Integer value accessed as HINT32.</p> <p>BLOB– Binary Large Object value accessed as CaBlob.</p> <p>TIMESTAMP– TimeStamp object accessed as CaTimeStamp, LPCTSTR, or CaString.</p>
Example	<pre>CaData a; a = 3; CaData::Types iType = a.GetType();</pre>
See Also	HUtilities Programming Fundamentals, CaString, CaTimeStamp, CaBlob

## CaData& operator=( const CaData &aData )

Owning Class	CaData
Description	Copies one data object to another.
Arguments	const CaData &aData – the object to copy.
Return Value	CaData & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals

**CaData& operator=( HINT32 nValue)**

Owning Class	CaData
Description	Sets the data value to the specified integer value.
Arguments	HINT32 nValue.
Return Value	CaData & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals

**CaData& operator=( LPCTSTR szValue)**

Owning Class	CaData
Description	Sets the data value to the specified string value.
Arguments	LPCTSTR szValue.
Return Value	CaData & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals

**CaData& operator=( const CaBlob& rBlob)**

Owning Class	CaData
Description	Sets the data value to the specified blob value.
Arguments	const CaBlob& rBlob.
Return Value	CaData & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals, CaBlob

**CaData& operator=( const CaTimeStamp& rTimeStamp)**

Owning Class	CaData
--------------	--------

Description	Sets the data value to the specified time stamp value.
Arguments	const CaTimeStamp& rTimeStamp.
Return Value	CaData & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals, CaTimeStamp

### **bool operator==( const CaData &rSrc)**

Owning Class	CaData
Description	Compares the data object with the specified object.
Arguments	const CaData &rSrc.
Return Value	bool – true if equal, false if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

### **bool operator!=( const CaData &rSrc)**

Owning Class	CaData
Description	Compares the data object with the specified object.
Arguments	const CaData &rSrc.
Return Value	bool – false if equal, true if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

### **operator HINT32()**

Owning Class	CaData
--------------	--------

Description	Conversion operator that returns the value of the CaData object as an HINT32 object.
Arguments	None.
Return Value	None.
Notes	Errors if CaData value is not of type INTEGER.
See Also	HUtilities Programming Fundamentals

### **operator LPCTSTR()**

Owning Class	CaData
Description	Conversion operator that returns the value of the CaData object as an LPCTSTR object.
Arguments	None.
Return Value	None.
Notes	Errors if CaData value is not of type STRING or TIMESTAMP.
See Also	HUtilities Programming Fundamentals

### **operator CaString()**

Owning Class	CaData
Description	Conversion operator that returns the value of the CaData object as a CaString object.
Arguments	None.
Return Value	None.
Notes	Errors if CaData value is not of type STRING or TIMESTAMP.
See Also	HUtilities Programming Fundamentals, CaString

### **operator (CaBlob&)()**

Owning Class	CaData
Description	Conversion operator that returns the value of the CaData object as blob object.

Arguments	None.
Return Value	None.
Notes	Errors if CaData value is not of type BLOB.
See Also	HUtilities Programming Fundamentals, CaBlob

### **operator (CaTimeStamp& )()**

Owning Class	CaData
Description	Conversion operator that returns the value of the CaData object as timestamp object.
Arguments	None.
Return Value	None.
Notes	Errors if CaData value is not of type TIMESTAMP.
See Also	HUtilities Programming Fundamentals, CaTimeStamp

### **friend ostream& operator<<( ostream &o, const CaData &d )**

Owning Class	CaData
Description	Writes the array values onto an ostream.
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 debug purposes.
Example	<pre>CaData d; d = 3; cout &lt;&lt; d;</pre>
See Also	HUtilities Programming Fundamentals

## CaDataArray

Description	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	<div>clear</div> <div>size</div> <div>Find</div> <div>Operator[]</div> <div>operator&lt;&lt;</div>
See Also	CaData

### void Clear(void)

Owning Class	CaDataArray
Description	Empties the DataArray.
Arguments	None.
Return Value	None.
Notes	None.
Example	<pre>CaDataArray bag; bag.clear();</pre>
See Also	HUtilities Programming Fundamentals

### int GetSize(void) const

Owning Class	CaDataArray
Description	Retrieves the quantity of CaData objects in the array.
Arguments	None.
Return Value	int – number of objects in array.
Notes	None.

**Example**

```
CaDataArray bag; int iSize = bag.size();
```

**See Also**

HUtilities Programming Fundamentals

**bool Find (const CaData& rData, HUINT32& rIndex) const****Owning Class**

CaDataArray

**Description**

Searches all elements for one whose value matches the supplied object. Returns the array index of the found object in an argument.

**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);
```

**See Also**

HUtilities Programming Fundamentals, Props, CaData

**CaData & operator[] (HUINT32 index)****Owning Class**

CaDataArray

**Description**

Retrieves the indexed CaData object.

**Arguments**

HUINT32 index – index value.

**Return Value**

CaData & - reference to the indexed object.

**Notes**

None.

**Example**

```
CaDataArray bag; bag[0] = 3; CaData& rData = bag[0];
```

**See Also**

HUtilities Programming Fundamentals, Props, CaDataArray, CaData

**friend ostream& operator<<( ostream &o, const CaDataArray &a )**

Owning Class	CaDataArray
Description	Writes the key values and their associated data values onto an ostream.
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 debug purposes.
Example	<pre>CaDataArray bag; bag[0] = 3; cout &lt;&lt; bag;</pre>
See Also	HUtilities Programming Fundamentals, Props, CaData

## CaString

Description	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	<div>CaString</div> <div>Append</div> <div>ToUpper</div> <div>ToLower</div> <div>IsEmpty</div> <div>MakeEmpty</div> <div>GetLength</div> <div>Trim</div> <div>IsValid</div> <div>GetMultiByteString</div> <div>Operator=</div> <div>Operator+=</div> <div>Operator-=</div> <div>Operator==</div> <div>Operator!=</div> <div>Operator LPCTSTR()</div> <div>Operator char* ()</div> <div>operator&lt;&lt;</div>
See Also	HUtilities Programming Fundamentals

## CaString(void)

Owning Class	CaString
Description	Default constructor. Initialize new string value to NULL.
Arguments	None.
Return Value	None.
Notes	Value is set to NULL.
Example	<pre>CaString d;</pre>
See Also	HUtilities Programming Fundamentals

## CaString(const CaString &src)

Owning Class	CaString
Description	Copy Constructor – Initialize new string with source string.
Arguments	const CaString &src – value for new string.
Return Value	None.
Notes	None.
Example	<pre>CaString d("My String"); CaString newString(d);</pre>
See Also	HUtilities Programming Fundamentals

## CaString(LPCTSTR lpStr)

Owning Class	CaString
Description	Constructor – Initialize new string with this value.
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	<pre>CaString d(_T("My String"));</pre>
See Also	HUtilities Programming Fundamentals

## CaString(char\* lpStr)

Owning Class	CaString
Description	Constructor – Initialize new string with this value.
Arguments	char* lpStr – value for new string.
Return Value	None.
Notes	char* is a Non-Unicode string regardless of how it is compiled.
Example	<pre>CaString d("My String");</pre>
See Also	HUtilities Programming Fundamentals

## CaString& Append( LPCTSTR src )

Owning Class            CaString

Description            Appends a string to this one.

Arguments              LPCTSTR src – String to append.

Return Value           CaString& - reference to this string.

Notes                   None.

Example

```
CaString d("My String"); d.Append(_T("Appended String"));
```

See Also                HUtilities Programming Fundamentals

## CaString& ToUpper ( void )

Owning Class            CaString

Description            Converts this string to upper case.

Arguments              None.

Return Value           CaString& - reference to this string.

Notes                   None.

Example

```
CaString d("My String"); d.ToUpper();
```

See Also                HUtilities Programming Fundamentals

## CaString& ToLower ( void )

Owning Class	CaString
Description	Converts this string to lower case.
Arguments	None.
Return Value	CaString& - reference to this string.
Notes	None.
Example	<pre>CaString d("My String"); d.ToLower();</pre>
See Also	HUtilities Programming Fundamentals

## bool IsEmpty(void) const

Owning Class	CaString
Description	Determines if value is set to NULL, an empty string, or data.
Arguments	None.
Return Value	bool - true if NULL or empty, false if length > 0.
Notes	None.
Example	<pre>CaString d("My String"); bool bResult = d.IsEmpty();</pre>
See Also	HUtilities Programming Fundamentals

## CaString& MakeEmpty( void )

Owning Class	CaString
Description	Sets string value to NULL.
Arguments	None.
Return Value	CaString& - reference to this string.
Notes	None.
Example	<pre>CaString d("My String"); d.MakeEmpty();</pre>
See Also	HUtilities Programming Fundamentals

## long GetLength(void) const

Owning Class	CaString
Description	Returns the number of characters in this string.
Arguments	None.
Return Value	long – string length.
Notes	<p>This function returns number characters not number of bytes. If Unicode, then each character is two bytes wide.</p> <p>If string value is NULL or the empty string then return value is zero.</p>
Example	<pre>CaString d("My String"); long length = d.GetLength();</pre>
See Also	HUtilities Programming Fundamentals

## CaString& Trim( void )

Owning Class	CaString
Description	Removes white space from both ends of the string.
Arguments	None.
Return Value	CaString& - reference to this string.
Notes	None.
Example	<pre>CaString d(" My String "); d.Trim();</pre>
See Also	HUtilities Programming Fundamentals

## bool IsValid(void) const

Owning Class	CaString
Description	Determines whether the pointer to this string is NULL or not.
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	<pre>CaString d(" My String "); bool bResult = d.IsValid();</pre>
See Also	HUtilities Programming Fundamentals

**const char\* GetMultiByteString(void)**

Owning Class	CaString
Description	If compiled with Unicode this returns the Non-Unicode version of the data.
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	<pre>CaString d(_T(" My String ")); const char* p = d.GetMultiByteString();</pre>
See Also	HUtilities Programming Fundamentals

**CaString& operator=( const CaString &src )**

Owning Class	CaString
Description	Copies one object to another.
Arguments	const CaString &src - the object to copy.
Return Value	CaString & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals

**CaString& operator=( LPCTSTR szSrc)**

Owning Class	CaString
Description	Sets the data value to the specified string value.
Arguments	LPCTSTR szSrc.
Return Value	CaString & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals

**CaString& operator=( char\* szSrc)**

Owning Class	CaString
Description	Sets the data value to the specified string value.
Arguments	LPCTSTR szSrc.
Return Value	CaString & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals

**bool operator==( const CaString &rSrc)**

Owning Class	CaString
Description	Compares the data object with the specified object.
Arguments	const CaString &rSrc.
Return Value	bool - true if equal, false if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

**bool operator==( LPCTSTR src) const**

Owning Class	CaString
Description	Compares the string value with the specified string.
Arguments	LPCTSTR src.
Return Value	bool - true if equal, false if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

**bool operator!=( const CaString &rSrc) const**

Owning Class	CaString
--------------	----------

Description	Compares the object with the specified object.
Arguments	const CaString &rSrc.
Return Value	bool – false if equal, true if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

### **bool operator!=( LPCTSTR src) const**

Owning Class	CaString
Description	Compares the string value with the specified string.
Arguments	LPCTSTR src.
Return Value	bool – false if equal, true if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

### **CaString& operator+=( LPCTSTR src )**

Owning Class	CaString
Description	Appends a string to this one.
Arguments	LPCTSTR src – String to append.
Return Value	CaString& - reference to this string.
Notes	None.
Example	<pre>CaString d("My String"); d += _T("Appended String");</pre>
See Also	HUtilities Programming Fundamentals

### **CaString& operator-=( LPCTSTR src )**

Owning Class	CaString
--------------	----------

Description	Subtracts a string from this one.
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	
	<pre>CaString d("My String"); d -= _T("String");</pre>
See Also	HUtilities Programming Fundamentals

### **operator LPCTSTR()**

Owning Class	CaString
Description	Conversion operator that returns the value of the CaString object as an LPCTSTR object.
Arguments	None.
Return Value	LPCTSTR.
Notes	If value is NULL returns empty string.
See Also	HUtilities Programming Fundamentals

### **operator char \*()**

Owning Class	CaString
Description	Conversion operator that returns the value of the CaString object as a char* object.
Arguments	None.
Return Value	char*.
Notes	If value is NULL returns empty string.
See Also	HUtilities Programming Fundamentals, CaString

**friend ostream& operator<<( ostream &o, const CaString &rString )**

Owning Class	CaString
Description	Writes string onto an ostream.
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	<pre>CaString d("My String"); cout &lt;&lt; d;</pre>
See Also	HUtilities Programming Fundamentals

## CaStringList

**Description**                      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<<

**See Also**                          HUtilities Programming Fundamentals, CaString

## CaStringList(void)

Owning Class	CaStringList
Description	Default constructor. Initialize empty list.
Arguments	None.
Return Value	None.
Notes	None.
Example	<pre>CaStringList d;</pre>
See Also	HUtilities Programming Fundamentals

## CaStringList(const CaStringList &src)

Owning Class	CaStringList
Description	Copy Constructor – Initialize new string list from source string list.
Arguments	const CaStringList &src – source list.
Return Value	None.
Notes	None.
Example	<pre>CaStringList d; CaStringList newStringList(d);</pre>
See Also	HUtilities Programming Fundamentals

**bool Add(const CaString szString)**

Owning Class	CaStringList
Description	Adds the string to the list.
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);
```

See Also            HUtilities Programming Fundamentals

**CaString & Find(const CaString szString)**

Owning Class	CaStringList
Description	Search for the specified string in the list.
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);
```

See Also            HUtilities Programming Fundamentals

## **bool Remove(const CaString szString)**

Owning Class	CaStringList
Description	Find matching string in list and remove it.
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	<pre>CaStringList d; CaString s("My String"); CaString&amp; rString = d.Remove(s);</pre>
See Also	HUtilities Programming Fundamentals

## **bool Add(LPCTSTR szString)**

Owning Class	CaStringList
Description	Adds the string to the list.
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	<pre>CaStringList d; bool bResult = d.Add("My String");</pre>
See Also	HUtilities Programming Fundamentals

## CaString & Find(LPCTSTR szString)

Owning Class	CaStringList
Description	Search for the specified string in the list.
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	<pre>CaStringList d; bool bResult = d.Find("My String");</pre>
See Also	HUtilities Programming Fundamentals

## bool Remove(LPCTSTR szString)

Owning Class	CaStringList
Description	Find matching string in list and remove it.
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	<pre>CaStringList d; bool bResult = d.Remove("My String");</pre>
See Also	HUtilities Programming Fundamentals

**bool IsEmpty(void) const**

Owning Class	CaStringList
Description	Determines if list is empty.
Arguments	None.
Return Value	bool – true if NULL or empty, false if length > 0.
Notes	None.
Example	<pre>CaStringList d; bool bResult = d.IsEmpty();</pre>
See Also	HUtilities Programming Fundamentals

**void Clean( void )**

Owning Class	CaStringList
Description	Removes all elements from the list.
Arguments	None.
Return Value	None.
Notes	None.
Example	<pre>CaStringList d; d.Clean();</pre>
See Also	HUtilities Programming Fundamentals

### **int GetSize(void) const**

Owning Class	CaStringList
Description	Returns the number of elements in this list.
Arguments	None.
Return Value	int – number of strings in list.
Notes	None.
Example	<pre>CaStringList d; int length = d.GetSize();</pre>
See Also	HUtilities Programming Fundamentals

### **CaString GetAllNames(void) const**

Owning Class	CaStringList
Description	Concatenates all strings in the list into a new string separated by “\n\t”.
Arguments	None.
Return Value	CaString - reference to this string.
Notes	None.
Example	<pre>CaStringList d; CaString szString = d.GetAllNames();</pre>
See Also	HUtilities Programming Fundamentals

### **Int GetIndex(LPCTSTR szString ) const**

Owning Class	CaStringList
Description	Finds and returns the index value for the string that matches szString.
Arguments	LPCTSTR szString – string to find
Return Value	int – index value for matching string.
Notes	None.

**Example**

```
CaStringList d; int iIndex = d.GetIndex("Matching String");
```

See Also                      HUtilities Programming Fundamentals

**CaStringList& operator=( const CaStringList & CRight )**

Owning Class	CaStringList
Description	Copies one object to another.
Arguments	const CaStringList & CRight – the list to copy.
Return Value	CaStringList & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals

**bool operator==( const CaStringList & CRight) const**

Owning Class	CaStringList
Description	Compares the list with the specified list.
Arguments	const CaStringList & CRight.
Return Value	bool – true if equal, false if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

**bool operator!=( const CaStringList & CRight) const**

Owning Class	CaStringList
--------------	--------------

Description	Compares the list with the specified list.
Arguments	const CaStringList & CRight.
Return Value	bool – false if equal, true if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

### **CaStringList& operator+=(const CaStringList & CRight)**

Owning Class	CaStringList
Description	Appends a string list to this one.
Arguments	const CaStringList & CRight – String list to append.
Return Value	CaStringList& - reference to this string list.
Notes	Duplicates are ignored.
Example	<pre>CaStringList d, e; d += e;</pre>
See Also	HUtilities Programming Fundamentals

### **CaStringList& operator-=(const CaStringList & CRight)**

Owning Class	CaStringList
Description	Subtracts a string list from this one.
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	<pre>CaStringList d; d -= e;</pre>
See Also	HUtilities Programming Fundamentals

## CaString & operator[] (int i)

Owning Class	CaStringList
Description	Retrieves the string from the list at the specified index.
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	<pre>CaStringList d; CaString&amp; r5 = d[0];</pre>
See Also	HUtilities Programming Fundamentals

## operator bool() const

Owning Class	CaStringList
Description	Determines whether this is valid or not. If the “this” pointer is NULL list is invalid.
Arguments	None.
Return Value	bool – true if valid, false if not.
Notes	None.
Example	<pre>CaStringList d; if(d) d.Add("MyString");</pre>
See Also	HUtilities Programming Fundamentals

**friend ostream& operator<<( ostream &o, const CaStringList &rList )**

Owning Class	CaStringList
Description	Writes strings from list onto an ostream.
Arguments	ostream &o – Ostream on which to write. CaStringList &rList – reference to a String List.
Return Value	ostream & - reference to the ostream.

## CaTimeStamp

Description 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<<

See Also None.

**void CaTimeStamp(void)**

Owning Class	CaTimeStamp
Description	Default constructor. Creates an empty TimeStamp.
Arguments	None.
Return Value	None.
Notes	None.
Example	<pre>CaTimeStamp t1;</pre>
See Also	HUtilities Programming Fundamentals

**void CaTimeStamp(const CaTimeStamp& rTimeStamp)**

Owning Class	CaTimeStamp
Description	Copy constructor sets one TimeStamp to another.
Arguments	const CaTimeStamp& rTimeStamp – the TimeStamp to copy.
Return Value	None.
Notes	None.
Example	<pre>CaTimeStamp t1(t2);</pre>
See Also	HUtilities Programming Fundamentals

**void CaTimeStamp(HINT32 nYear, HINT32 nMonth, HINT32 nDay, HINT32 nHour, HINT32 nMinute, HINT32 nSecond, HINT32 nFraction)**

Owning Class	CaTimeStamp
Description	Constructor. Sets initial values.
Arguments	HINT32 nYear, HINT32 nMonth, HINT32 nDay, HINT32 nHour, HINT32 nMinute, HINT32 nSecond, HINT32 nFraction.
Return Value	None.
Notes	nFraction defaults to zero.
Example	<pre>CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4);</pre>
See Also	HUtilities Programming Fundamentals

**CaTimeStamp& SetYear( HINT32 iYear )**

Owning Class	CaTimeStamp
Description	Set Year value.
Arguments	HINT32 iYear – Year value.
Return Value	CaTimeStamp& - reference to this object so functions can be chained together.
Notes	None.
Example	<pre>CaTimeStamp t1; t1.SetYear(2001);</pre>
See Also	HUtilities Programming Fundamentals, GetYear

### CaTimeStamp& SetMonth( HINT32 iMonth )

Owning Class	CaTimeStamp
Description	Set Month value.
Arguments	HINT32 iMonth – Month value.
Return Value	CaTimeStamp& - reference to this object so functions can be chained together.
Notes	None.
Example	<pre>CaTimeStamp t1; t1.SetMonth(2001);</pre>
See Also	HUtilities Programming Fundamentals, GetMonth

### CaTimeStamp& SetDay( HINT32 iDay )

Owning Class	CaTimeStamp
Description	Set Day value.
Arguments	HINT32 iDay – Day value.
Return Value	CaTimeStamp& - reference to this object so functions can be chained together.
Notes	None.
Example	<pre>CaTimeStamp t1; t1.SetDay(2001);</pre>
See Also	HUtilities Programming Fundamentals, GetDay

**CaTimeStamp& SetHour( HINT32 iHour )**

Owning Class	CaTimeStamp
Description	Set Hour value.
Arguments	HINT32 iHour – Hour value.
Return Value	CaTimeStamp& - reference to this object so functions can be chained together.
Notes	None.
Example	<pre>CaTimeStamp t1; t1.SetHour(2001);</pre>
See Also	HUtilities Programming Fundamentals, GetHour

**CaTimeStamp& SetMinute( HINT32 iMinute )**

Owning Class	CaTimeStamp
Description	Set Minute value.
Arguments	HINT32 iMinute – Minute value.
Return Value	CaTimeStamp& - reference to this object so functions can be chained together.
Notes	None.
Example	<pre>CaTimeStamp t1; t1.SetMinute(2001);</pre>
See Also	HUtilities Programming Fundamentals, GetMinute

**CaTimeStamp& SetSecond( HINT32 iSecond )**

Owning Class	CaTimeStamp
Description	Set Second value.
Arguments	HINT32 iSecond – Second value.
Return Value	CaTimeStamp& - reference to this object so functions can be chained together.
Notes	None.
Example	<pre>CaTimeStamp t1; t1.SetSecond(2001);</pre>
See Also	HUtilities Programming Fundamentals, GetSecond

**CaTimeStamp& SetFraction( HINT32 iFraction )**

Owning Class	CaTimeStamp
Description	Set Fraction value.
Arguments	HINT32 iFraction – Fraction value.
Return Value	CaTimeStamp& - reference to this object so functions can be chained together.
Notes	None.
Example	<pre>CaTimeStamp t1; t1.SetFraction(2001);</pre>
See Also	HUtilities Programming Fundamentals, GetFraction

**HINT32 GetYear( void ) const**

Owning Class            CaTimeStamp

Description            Get Year value.

Arguments              None.

Return Value           HINT32 – Year value of TimeStamp.

Notes                   None.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iYear = t1.GetYear();
```

See Also                HUtilities Programming Fundamentals, SetYear

**HINT32 GetMonth( void ) const**

Owning Class            CaTimeStamp

Description            Get Month value.

Arguments              None.

Return Value           HINT32 – Month value of TimeStamp.

Notes                   None.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iMonth = t1.GetMonth();
```

See Also                HUtilities Programming Fundamentals, SetMonth

**HINT32 GetDay( void ) const**

Owning Class            CaTimeStamp

Description            Get Day value.

Arguments              None.

Return Value           HINT32 – Day value of TimeStamp.

Notes                   None.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iDay = t1.GetDay();
```

See Also                HUtilities Programming Fundamentals, SetDay

**HINT32 GetHour( void ) const**

Owning Class            CaTimeStamp

Description            Get Hour value.

Arguments              None.

Return Value           HINT32 – Hour value of TimeStamp.

Notes                   None.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iHour = t1.GetHour();
```

See Also                HUtilities Programming Fundamentals, SetHour

### HINT32 GetMinute( void ) const

Owning Class            CaTimeStamp

Description            Get Minute value.

Arguments              None.

Return Value           HINT32 – Minute value of TimeStamp.

Notes                   None.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iMinute = t1.GetMinute();
```

See Also                HUtilities Programming Fundamentals, SetMinute

### HINT32 GetSecond( void ) const

Owning Class            CaTimeStamp

Description            Get Second value.

Arguments              None.

Return Value           HINT32 – Second value of TimeStamp.

Notes                   None.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iSecond = t1.GetSecond();
```

See Also                HUtilities Programming Fundamentals, SetSecond

**HINT32 GetFraction( void ) const**

Owning Class            CaTimeStamp

Description            Get Fraction value.

Arguments              None.

Return Value           HINT32 – Fraction value of TimeStamp.

Notes                   None.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); HINT32 iFraction = t1.GetFraction();
```

See Also                HUtilities Programming Fundamentals, SetFraction

**CPtHTimeStamp& ChangeTime(HINT32 iTimezone)**

Owning Class            CaTimeStamp

Description            Convert time to specified timezone.

Arguments              HINT32 iTimezone – Difference in hours between new time and GMT time.

Return Value           CPtHTimeStamp& - reference to this TimeStamp.

Notes                   Adjusts month and day as well.

Example

```
CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); t1.ChangeTime(3);
```

See Also                HUtilities Programming Fundamentals

## CPtHTimeStamp& ConvertGMTime(void)

Owning Class	CaTimeStamp
Description	Convert time to specified timezone.
Arguments	None.
Return Value	CPtHTimeStamp& - reference to this TimeStamp.
Notes	None.
Example	<pre>CaTimeStamp t1(2001, 5, 29, 18, 5, 32, 4); t1. ConvertGMTime();</pre>
See Also	HUtilities Programming Fundamentals

## CaTimeStamp& operator=( const CaTimeStamp &rTimeStamp )

Owning Class	CaTimeStamp
Description	Assignment operator. Copies one Time Stamp object to another.
Arguments	const CaTimeStamp & rTimeStamp – the object to copy.
Return Value	CaTimeStamp & - reference to left-hand object.
Notes	None.
See Also	HUtilities Programming Fundamentals

## bool operator==( const CaTimeStamp &rTimeStamp) const

Owning Class	CaTimeStamp
Description	Compares the TimeStamp with the specified object.
Arguments	const CaTimeStamp &rTimeStamp.
Return Value	bool – true if equal, false if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

**bool operator!=( const CaTimeStamp &rTimeStamp) const**

Owning Class	CaTimeStamp
Description	Compares the TimeStamp with the specified object.
Arguments	const CaTimeStamp &rTimeStamp.
Return Value	bool – false if equal, true if not.
Notes	None.
See Also	HUtilities Programming Fundamentals

**friend ostream& operator<<( ostream &o, const CaTimeStamp &rTimeStamp )**

Owning Class	CaTimeStamp
Description	Writes a description of the contents onto an ostream.
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	<pre>CaTimeStamp b(5, ptr); cout &lt;&lt; b;</pre>
See Also	HUtilities Programming Fundamentals



# HSDK Messages

---

This chapter lists of all HSDK messages displayed during the execution of HSDK functions. Other Harvest messages may be displayed. Refer to the *Messages Guide* to view information regarding additional messages.

Messages are preceded by an encoded 32-bit hexadecimal number.

- The upper eight bits comprise the severity code. Zero is Informational. One is Warning. Two and Three are Error.
- The second eight bits comprise the component code. A value of nine corresponds to the HSDK. Other values correspond to lower-level Harvest components.
- The next sixteen bits comprise a sequential message number.

This document only discusses HSDK messages.

- Messages that begin with an I are informational. They are commonly used to indicate successful completion of a function, or give you additional information about it.
- Messages that begin with a W indicate a warning condition. The function can continue but some action may be required to return the system to full operation.
- Messages that begin with an E indicate an error condition. The function cannot be completed until this condition is corrected.

## Messages and Descriptions

### **W01090001 HSDK\_W\_NO\_MESSAGEKEY WARNING: No Message Key Found in Container.**

**Reason:**

This is an internal error.

The message handler was not properly initialized.

**Action:**

Verify that the file hSDKenglish.msg is in the Harvest installation directory.

### **I0009002D HSDK\_I\_LOGGEDIN Information: %s1 has successfully logged into Harvest broker %s2.**

**Reason:**

The user, %s1, has successfully logged into the broker %s2.

**Action:**

None. This is an informational message.

### **I0009002E HSDK\_I\_LOGGEDOUT Information: %s1 has successfully logged out from Harvest broker %s2.**

**Reason:**

The user, %s1, has successfully logged out from the broker %s2.

**Action:**

None. This is an informational message.

### **E03090002 HSDK\_E\_NO\_HARVEST ERROR: Could not instantiate CaHarvest Object.**

**Reason:**

This is an internal run-time error.

**Action:**

Verify available system memory.

Verify that the Harvest client installation and the HSDK installation are both complete.

**E03090003 HSDK\_E\_NO\_BROKER\_CONNECT ERROR: Could not connect to Broker: %s1.**

**Reason:**

Could not connect to the specified broker.

**Action:**

Verify that the broker name is entered correctly.

Verify broker is running.

Verify that your client machine can communicate with the broker machine.

**E03090004 HSDK\_E\_NO\_LOGIN ERROR: Login failed.**

**Reason:**

Could not log in to the specified broker.

**Action:**

Verify that the user name and password are entered correctly.

Verify that there are Harvest servers running.

**E03090005 HSDK\_E\_NO\_CONTEXT\_GET ERROR: Could not get Harvest Context.**

**Reason:**

Could not get CaContext object from CaHarvest object.

**Action:**

Verify that the user has successfully logged into the broker.

Verify that there are servers running.

**E03090006 HSDK\_E\_NO\_CONTEXT\_SET ERROR: Could not set Harvest Context: %s1.**

**Reason:**

Could not add %s1 to the current context.

**Action:**

Verify that parent objects are already set (for example, Project, State).

Verify that the object id or object name is correct.

Verify that the user has successfully logged into the broker.

Verify that the user has sufficient access rights.

Verify that there are servers running.

**E03090007 HSDK\_E\_NO\_LOGGEDIN ERROR: Not Currently Logged in to Harvest.**

**Reason:**

User is not logged in to Harvest.

**Action:**

Log in.

**E03090008 HSDK\_E\_NO\_PROJECT\_LIST ERROR: Could not retrieve Project list.**

**Reason:**

Could not retrieve the Project list from the CaHarvest Object, possibly while trying to add a Project to the current context.

**Action:**

Verify that user is currently logged in to Harvest.

Verify that user has sufficient access rights.

Verify that there are servers running.

**E03090009 HSDK\_E\_NO\_PROJECT ERROR: Could not get Project.**

**Reason:**

Could not retrieve a Project, either from the CaHarvest object while trying to retrieve the Project list or from the CaContext Object while trying to set or get the current context.

**Action:**

If trying to get the Project from the current context ensure that it was properly set.

If trying to set the Project ensure that the name or id is correct.

Verify that user is currently logged in to Harvest.

Verify that the user has sufficient access rights.

Verify that there are servers running.

**E0309000A HSDK\_E\_NO\_STATE\_LIST ERROR: Could not retrieve State list.****Reason:**

Could not retrieve the State list from the CaContext Object, possibly while trying to add a State to the current context.

**Action:**

Verify that the Project is currently set.

Verify that the State name or Id is correct.

Verify that the user is currently logged in to Harvest.

Verify that user has sufficient access rights.

Verify that there are servers running.

**HSDK\_E\_NO\_STATE E0309000B ERROR: Could not get State.****Reason:**

Could not retrieve a State from the CaContext Object while trying to set or get the current context.

**Action:**

Verify that the Project is properly set.

If trying to get the State from the current context ensure that it was properly set.

If trying to set the State ensure that the name or id is correct.

Verify that user is currently logged in to Harvest.

Verify that the user has sufficient access rights.

Verify that there are servers running.

#### **E0309000C HSDK\_E\_NO\_PROCESS\_LIST ERROR: Could not retrieve Process list.**

##### **Reason:**

Could not retrieve a Process list from the CaContext Object while trying to set or get the current context.

##### **Action:**

Verify that the State is properly set.

If trying to get the Process from the current context ensure that it was properly set.

If trying to set the Process ensure that the name or id is correct.

Verify that user is currently logged in to Harvest.

Verify that the user has sufficient access rights.

Verify that there are servers running.

#### **E0309000D HSDK\_E\_NO\_PROCESS ERROR: Could not get Process.**

##### **Reason:**

Could not retrieve a Process list from the CaContext Object while trying to set or get the current context.

##### **Action:**

Verify that the State is properly set.

If trying to get the Process from the current context ensure that it was properly set.

If trying to set the Process ensure that the name or id is correct.

Verify that user is currently logged in to Harvest.

Verify that the user has sufficient access rights.

Verify that there are servers running.

**E0309000E HSDK\_E\_UNEXPECTED\_ERROR: Unexpected error during HSDK operation.**

**Reason:**

This is usually a programming error. Generally the programmer has tried to perform actions in some illegal sequence. It can also be that the broker or server died while the program was executing.

**Action:**

Verify that the broker and server are currently running.

Verify that the user is currently logged in.

Verify that the context is properly set for the attempted operation.

**E0309000F HSDK\_E\_NO\_CONTEXT\_SET\_BYID ERROR: Could not set Harvest Context by ID: %d1.**

**Reason:**

Could not add the specified object, %d1, to the current context.

**Action:**

Verify that parent objects are already set (for example, Project, State).

Verify that the object id is correct.

Verify that user is currently logged in to Harvest.

Verify that the user has sufficient access rights.

Verify that there are servers running.

**E03090010 HSDK\_E\_NO\_CHECKOUT\_CONTEXT ERROR: Could not get Checkout Process from Context.**

**Reason:**

Could not retrieve the Checkout Process from the current context.

**Action:**

Verify that the Checkout Process was properly set.

**E03090011 HSDK\_E\_INVALID\_PROCESS ERROR: Invalid Process: %s1.**

**Reason:**

A process of the name %s1 was found in the State's process list but the process type is not what was expected.

**Action:**

Verify that the process name is of the requested type.

**E03090010 HSDK\_E\_NO\_PROMOTE\_CONTEXT ERROR: Could not get Promote Process from Context.**

**Reason:**

Could not retrieve the Promote Process from the current context.

**Action:**

Verify that the Promote Process was properly set.

**E03090011 HSDK\_E\_NO\_USER\_LIST ERROR: Could not retrieve User list.**

**Reason:**

Could not retrieve the User list from the CaHarvest Object.

**Action:**

Verify that user is currently logged in to Harvest.

Verify that user has sufficient access rights.

Verify that there are servers running.

**E03090012 HSDK\_E\_NO\_USER ERROR: Could not get User.**

**Reason:**

Could not retrieve a User from the CaHarvest Object while trying to get the User list.

**Action:**

Verify that user is currently logged in to Harvest.

Verify that user has sufficient access rights.

Verify that there are servers running.

**E03090013 HSDK\_E\_NO\_PACKAGE\_LIST ERROR: Could not retrieve Package list.**

**Reason:**

Could not retrieve the Package list from the CaContext Object while trying to set the Package context.

**Action:**

Verify that the State context is properly set.

Verify that user is currently logged in to Harvest.

Verify that user has sufficient access rights.

Verify that there are servers running.

**E03090014 HSDK\_E\_NO\_PACKAGE ERROR: Could not get Package.**

**Reason:**

Could not retrieve the Package from the CaContext Object while trying to set the Package context.

**Action:**

Verify that the State context is properly set.

Verify that the Package name is correct and that it exists in the State.

Verify that user is currently logged in to Harvest.

Verify that user has sufficient access rights.

Verify that there are servers running.

**E03090015 HSDK\_E\_INVALID\_PACKAGE ERROR: Package, %s1, not found in State, %s2.**

**Reason:**

A Package operation (e.g. Approve, Promote, Demote) was attempted from State %s2 using Package %s1. The Package was not found in the State.

**Action:**

Verify that the State context is properly set.

Verify that the Package name is correct and that it exists in the State.

**E03090016 HSDK\_E\_NO\_VIEW\_LIST ERROR: Could not retrieve View list.**

**Reason:**

Could not retrieve the View list either from the current Project or while trying to add a Snapshot View to the context.

**Action:**

If trying to get a Project's View list verify that the Project is set.

If trying to set the context Snapshot View verify that the State context is properly set.

**E03090017 HSDK\_E\_NO\_VIEW ERROR: Could not get View.****Reason:**

Could not retrieve a View either from the current Project or while trying to add a Snapshot View to the context.

**Action:**

If trying to get a Project's View list verify that the Project is set.

If trying to set the context Snapshot View verify that the State context is properly set.

**E03090018 HSDK\_E\_NOT\_SSVIEW ERROR: %s1 should be a Snapshot View but it is a %s2.****Reason:**

Tried to add a Snapshot View%s1 to the context but it is not a Snapshot view. It is a %s2.

**Action:**

Verify the Snapshot View name or id.

**E03090019 HSDK\_E\_INVALID\_VIEWPATH ERROR: View Path, %s1, not found in View, %s2.****Reason:**

Tried to addView Path %s1 to the context but it is not a View Path in View %s2.

**Action:**

Verify the View Path name and that it exists in the View.

**E0309001A HSDK\_E\_NO\_VERSION\_LIST ERROR: Could not retrieve Version list.**

**Reason:**

An unexpected failure occurred while trying to execute the version chooser.

**Action:**

Verify that the User is logged in to Harvest and that the State context is properly set. If the State is a Snapshot State verify the Snapshot View is set.

Verify that servers are running.

**E0309001B HSDK\_E\_NO\_VERSION ERROR: Could not get Version.**

**Reason:**

An unexpected failure occurred while trying to execute the version chooser.

**Action:**

Verify that the User is logged in to Harvest and that the State context is properly set. If the State is a Snapshot State verify the Snapshot View is set.

Verify that servers are running.

**E0309001C HSDK\_E\_NO\_FILEAGENT ERROR: No Remote File Agent found on %s1.**

**Reason:**

A Checkin or Checkout was attempted using remote machine, %s1. No Harvest File Agent was found running on the remote machine.

**Action:**

Verify that a remote agent is running.

**E0309001D HSDK\_E\_NO\_FILEAGENT\_LOGIN ERROR: Could not Login to Remote FileAgent on %s1.****Reason:**

A Checkin or Checkout was attempted using remote machine, %s1. Could not log in to the Harvest File Agent on the remote machine.

**Action:**

Verify the user name and password for the remote machine.

Verify that the user has sufficient access rights to log in to a remote agent on the remote machine.

**E0309001E HSDK\_E\_NO\_FILEAGENT\_SETDIR ERROR: Could not set File Agent directory to %s1.****Reason:**

A Checkin or Checkout was attempted using a remote machine. The Client Directory, %s1 was not found or not accessible.

**Action:**

Verify the directory exists on the remote machine and that the user can access it.

**E0309001F HSDK\_E\_INVALID\_VERSION\_LIST ERROR: Invalid Version List.****Reason:**

Tried to Checkin or Checkout an empty set of versions.

**Action:**

Verify the Version Chooser selection criteria and execute the Version Chooser.

**E03090020 HSDK\_E\_INVALID\_PACKAGE\_LIST ERROR: Invalid Package List.****Reason:**

Tried to perform a Package operation on an empty set of packages.

**Action:**

Verify the Package Chooser selection criteria and execute the Package Chooser.

**E03090021 HSDK\_E\_NO\_DEMOTE\_CONTEXT ERROR: Could not get Demote Process from Context.**

**Reason:**

Could not retrieve the Demote Process from the current context.

**Action:**

Verify that the Demote Process context is properly set.

**E03090022 HSDK\_E\_NO\_CHECKIN\_CONTEXT ERROR: Could not get Checkin Process from Context.**

**Reason:**

Could not retrieve the Checkin Process from the current context.

**Action:**

Verify that the Checkin Process context is properly set.

**E03090023 HSDK\_E\_INVALID\_FILE\_LIST ERROR: Invalid File List.**

**Reason:**

Attempted to Checkin an empty file list.

**Action:**

Verify the file list.

**E03090024 HSDK\_E\_INVALID\_ATTRIBUTES ERROR: Invalid Attributes passed to function.**

**Reason:**

Attempted to create a Package with invalid attributes.

**Action:**

Verify that the attributes set in the CaCreatePackage object are correct.

**E03090025 HSDK\_E\_NO\_CREATEPACKAGE\_CONTEXT ERROR: Could not get Create Package Process from Context.**

**Reason:**

Could not retrieve the Create Package Process from the current context.

**Action:**

Verify that the Create Package Process context is properly set.

**E03090026 HSDK\_E\_NO\_PACKAGE\_CREATE ERROR: Could not Create Package: %s1.**

**Reason:**

Could not Create the Package %s1.

**Action:**

Verify that the %s1 does not already exist.

Verify that the user has sufficient access rights to create a package.

**E03090027 HSDK\_E\_NO\_APPROVE\_CONTEXT ERROR: Could not get Approve Process from Context.**

**Reason:**

Could not retrieve the Approve Process from the current context.

**Action:**

Verify that the Approve Process context is properly set.

**E03090028 HSDK\_E\_NO\_DELETEVERSION\_CONTEXT ERROR: Could not get Delete Version Process from Context.**

**Reason:**

Could not retrieve the Delete Version Process from the current context.

**Action:**

Verify that the Delete Version Process context is properly set.

**E03090029 HSDK\_E\_NO\_REMOVEITEM\_CONTEXT ERROR: Could not get Remove Item Process from Context.**

**Reason:**

Could not retrieve the Remove Item Process from the current context.

**Action:**

Verify that the Remove Item Process context is properly set.

**E0309002A HSDK\_E\_INVALID\_ITEM ERROR: Invalid Item.**

**Reason:**

Attempted to execute a Remove Item process on an empty Item list.

**Action:**

Verify the Item list.

**E0309002B HSDK\_E\_MULTIPLE\_ITEMS ERROR: Multiple matching Items named: %s1, found in path: %s2.**

**Reason:**

Attempted to execute a Remove Item process on an Item list that has multiple Items of the same name, %s1, in with the View Path structure below %s2.

**Action:**

Try removing the Recursive attribute or adjust the View Path.

**E0309002C HSDK\_E\_NO\_CONCURRENTMERGE\_CONTEXT ERROR: Could not get Concurrent Merge Process from Context.**

**Reason:**

Could not retrieve the Concurrent Merge Process from the current context.

**Action:**

Verify that the Concurrent Merge Process context is properly set.

**E0309002F HSDK\_E\_INVALID\_SNAPSHOT\_OPTIONS ERROR: Take Snapshot Process called with conflicting options.**

**Reason:**

Tried to execute Snapshot with conflicting options by specifying a snapshot at a particular time and base it upon a reference snapshot with current packages in package chooser.

**Action:**

Change execution so that either it selects a particular time with a CaTimeStamp or specifies a particular reference snapshot with a NULL\_CATIMESTAMP object.

**E03090030 HSDK\_E\_NO\_TAKESNAPSHOT\_CONTEXT ERROR: Could not get Take Snapshot Process from Context.**

**Reason:**

Could not retrieve the Take Snapshot Process from the current context.

**Action:**

Verify that the Take Snapshot Process was properly set.

**E03090031 HSDK\_NO\_REF\_SNAPSHOT ERROR: Could not find reference snapshot in current Project.**

**Reason:**

Tried to execute Snapshot with a referenced snapshot option, but unable to find the view corresponding to this name to base the new snapshot on.

**Action:**

Verify that a snapshot view matching the name specified exists.

**E03090032 HSDK\_E\_NOT\_SNAPSHOT ERROR: Specified reference view is not a Snapshot View.**

**Reason:**

A view matching the specified reference view name was found but this view is not a snapshot view.

**Action:**

Verify that the view specified by the reference view name is a snapshot view and not a working view.

**E03090033 HSDK\_E\_INVALID\_UDP\_TYPE ERROR: Tried to select an invalid type of UDP.**

**Reason:**

HSDK supports only UDPs designated as “Server” or “Client” UDPs. The UDP selected does not carry that proper designation.

**Action:**

The Harvest user who created this UDP should validate it within the Harvest Administrator.

**E03090034 HSDK\_E\_UDP\_ADD\_ARG ERROR: Unexpected error in setting UDP additional arguments.**

**Reason:**

Unexpected error in setting UDP additional arguments.

**Action:**

Check the string that was passed to the additional arguments function and verify that the UDP accepts additional arguments.

**E03090035 HSDK\_E\_UDP\_ADD\_INPUT ERROR: Unexpected error in setting UDP additional input.**

**Reason:**

Unexpected error in setting UDP additional input.

**Action:**

Check the string that was passed to the additional input function and verify that the UDP accepts input.

**E03090036 HSDK\_E\_NO\_UDP\_CONTEXT ERROR: Could not get UDP Process from Context.**

**Reason:**

Could not retrieve the UDP Process from the current context.

**Action:**

Verify that the UDP Process was properly set in the context.

**E03090037 HSDK\_E\_UDP\_INPUTSECURE ERROR: ERROR: Attempted to alter a secure input for a UDP.****Reason:**

Setting the additional input failed because the UDP does not permit input to be provided.

**Action:**

Remove SetAdditionalInput command or have Harvest Administrator modify the UDP to allow input.

**E03090038 HSDK\_E\_SQL\_ZERORESULTS ERROR: SQL Statement returned 0 records.****Reason:**

CaSQL selected 0 records and returned an empty container.

**Action:**

Modify the SQL statement to select valid records.

**E03090039 HSDK\_E\_SQL\_BADSTATEMENT ERROR: Malformed SQL Statement.****Reason:**

SQL statement is invalid and not properly formatted SQL.

**Action:**

Correct any invalid syntax in the SQL statement and reset the CaSQL with SetSQLStatement(LPCTSTR szSQLStr).

**E0309003A HSDK\_E\_NO\_CROSSPROJECTMERGE\_CONTEXT ERROR: Could not get CrossProject Merge Process from Context.****Reason:**

Could not retrieve the CrossProject Merge Process from the current context.

**Action:**

Verify that the CrossProject Merge Process was properly set in the context.

**E0309003B HSDK\_E\_INVALID\_AGENT\_LOGIN\_INFO ERROR: Incomplete or Malformed Agent LogOn information.**

**Reason:**

Unable to connect to remote agent because remote logon information is incomplete.

**Action:**

Verify that the logon information specified in the Context is complete.

**E0309003C HSDK\_E\_INVALID\_AGENT\_LOGIN\_FAILURE ERROR: Remote Agent login failed. .**

**Reason:**

Login to remote agent failed.

**Action:**

Verify that the logon information specified in the Context is correct and complete and that an agent is running on the remote machine.

**E0309003D HSDK\_E\_INVALID\_VERSION\_NUMBER ERROR: Invalid version number specified.**

**Reason:**

The version number specified does not exist or is formatted incorrectly.

**Action:**

Verify version number specified is a string of the correct syntax, that the item name is correct, and that such a version does exist within the current view.

**E0309003E HSDK\_E\_INVALID\_PARENTPATH ERROR: Invalid parent path specified for creating directory item.**

**Reason:**

Parent Path specified does not exist in the current view.

**Action:**

Modify the specified parent directory path to represent a valid directory in the current view.

**E0309003F HSDK\_E\_INVALID\_USERDATA ERROR: Callback User Data is invalid.****Reason:**

The callback handler received invalid user data to send to the callback function.

**Action:**

Verify callback function is set and the callback function user data is specified in the proper format.

**E03090040 HSDK\_E\_INVALID\_CALLBACK\_CONTAINER ERROR: Callback Container is invalid.****Reason:**

The callback handler received an invalid container.

**Action:**

Reset the callback data and verify that the processing calling the callback is succeeding.

**E03090041 HSDK\_E\_INVALID\_CALLBACK\_FUNCTION ERROR: Callback function is invalid.****Reason:**

The function pointer passed to the callback handler is invalid.

**Action:**

Verify the callback function pointer exists and is valid at the time the process is executed.

**EDIT E03090042 HSDK\_E\_INVALID\_CALLBACK\_CHECKOUT ERROR: Checkout Process for callback is invalid.****Reason:**

The callback handler received an invalid pointer to the checkout process.

**Action:**

Verify callback function and userdata.

**E03090043 HSDK\_E\_INVALID\_CALLBACK\_CHECKIN ERROR: Checkin Process for callback is invalid.**

**Reason:**

The callback handler received an invalid pointer to the checkout process.

**Action:**

Verify callback function and userdata.

**E03090044 HSDK\_E\_UNABLE\_CHANGE\_PASSWORD ERROR: Unable to change user password.**

**Reason:**

Unable to update user's password because it does not follow local Password Policy.

**Action:**

Check local Password Policy and validate that the password conforms.

**E03090045 HSDK\_E\_ACCOUNT\_LOCKOUT ERROR: User account locked out.**

**Reason:**

Logging into Harvest Session returned a message that the user is locked out.

**Action:**

Contact Harvest Administrator to unlock user account.

**E03090046 HSDK\_E\_EXPIRED\_PASSWORD ERROR: User's password has expired, please change now.**

**Reason:**

Logging into Harvest Session returned a message that the user has an expired password and cannot login to Harvest.

---

**Action:**

Change User Password with the CaHarvest::ChangePassword(szPassword) function after executing a CaHarvest::Login attempt. Once the password is changed, the Harvest Session and Context will be valid.