



The Clarion Handy Tools Page

CHT CLASSES BY TYPE (BUILD: 21C.02.00)

Document Date: September 23, 2017

Application Configuration

Class: **HNDConfig** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: Application Configuration
Demos: [HNDSETUP.APP](#) [HNDZIPNFTP.APP](#) [HNDZIPNEMAIL.APP](#) [HNDMLSMTP.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDCONFIG.INC
Code: HNDCONFIG.CLW
Template: [EmbedHndConfigFunctions](#)
Description:

*** REVISED JANUARY 2017 ***

This class is used extensively by CHT either directly or indirectly (derived by other classes) to store access configurations for FTP, HTTP and SMTP as well as specific configuration setups for CHT Batch-Bots and CHT Snap-Ins that may require them. There are GET, SET and DISPLAY functions for each of the relevant protocols, Batch-Bots and Snap-Ins.

"GET" programmatically recalls existing settings, "SET" programmatically stores new or changed settings and "DISPLAY" provides an interactive dialog for end-user interaction with settings to provide for initial configuration of an application or to change an existing application configuration.

By default these configurations are stored under a configuration tag HPROP:CHTools. Developers have control over this default behaviour by calling configuration location settings OBJ.SetDefaultConfigLocation('YourConfig') early on in the initialization of any configuration class. Individual product configurations are stored in separately configurable locations under the above setting, via the xRegID parameter in each of the SET, GET and DISPLAY functions.

Default Location Examples:

HKEY_CURRENT_USER\Software\cht\maildefault

and

HKEY_CURRENT_USER\Software\cht\mailssl

Custom Location Examples:

HKEY_CURRENT_USER\Software\YourConfig\yourmaildefault

and

HKEY_CURRENT_USER\Software\YourConfig\yourmailssl

CHT Classes that utilize the HNDConfig class by derivation or composition are: HNDHTTP, HNDINFTFTP, HNDNETFTP, HNDPOP, HNDNETSMTP, HNDSMTP.

[Back To Previous](#)

Application Configuration

Class: **HNDIni** [Click For Methods Listing](#)
Status: Current
Parent: None
Compositions: INIClass(ABC) [HNDAppio](#) [HNDBinFileIO](#)
Type: Application Configuration
Demos: [All CHT Demo Apps](#)
DLL: None
TLB: None
Header: HNDINI.INC
Code: HNDINI.CLW

Template: [INFile](#) [INFileProcedure](#) [IniTableOProcedure](#)

Description:

*** REVISED APRIL 2017 ***

This class extends the INI File/Registry capabilities of your application beyond what is already provided by the Clarion INIClass. It primarily supports dynamic file/table naming (ISAM or SQL) so that the names, and details of tables used by your application are determined at run time rather than being hard coded. This class is added to your application by adding CHT's INIFile template to your Global Extensions.

The latest implementation allows for INI I/O redirection to a dictionary table via easy to-use settings provided by the CHT INIFile template.

[Back To Previous](#)

Browse Building Classes

Class: **HNDBrowse** [Click For Methods Listing](#)

Status: Current

Parent: [HNDBrwFilter](#) [HNDParse](#) [HNDError](#)

Compositions: [HNDSizes](#) [HNDGreenBar](#) [HNDStepClass](#)

Type: Browse Building Classes

Demos: [HND2.APP](#) [HND3.APP](#) [HNDSCHOOL.APP](#) [HNDACCES.APP](#)
[HNDSQL12.APP](#) [More](#)

DLL: None

TLB: None

Header: HNDBROWS.INC

Code: HNDBROWS.CLW

Template: [ExplorerBrowse](#) [LocatorOverRideControl](#)

Description:

*** REVISED MARCH 2016 ***

The HNDBrowse class provides the basis for CHT browse extension templates ExplorerBrowse and LocatorOverRideControl. While these templates provide a developer interface for extending the capabilities of ABC browses, this class (HNDBrowse) provides, in abstract, all of the functionalities delivered by those templates. Browse extension services as widely varied as dynamic sorting, real-language querying, header-click browse re-ordering, greenbar, column swapping and more.

The extremely broad range of browse-extension features delivered by the above-mentioned templates are either implemented in this class or are derived by this class from other, lower-level CHT classes either via derivation (*HNDBrwFilter*, *HNDParse*, *HNDError*) or by composition (*HNDSizes*, *HNDGreenBar*, *HNDStepClass*).

[Back To Previous](#)

Class: **HNDFileLoadBrowse** [Click For Methods Listing](#)

Status: Current

Parent: [HNDBrwFilter](#) [HNDError](#)

Compositions: [HNDGreenBar](#)

Type: Browse Building Classes

Demos: [HNDSCHOOL.APP](#) [HNDPEOPLE.LBX.APP](#)
[HNDCLXHT.APP](#) [HNDCLIENTCLEAN4VIEW.APP](#) [More](#)

DLL: None

TLB: None

Header: HNDFLBRW.INC

Code: HNDFLBRW.CLW

Template: [ListBoxBrowseExtender](#) [HandyMarkerBrowse](#) [HandyWebClientBrowseProcedure](#)
[ClientServerBrowseListBox](#) [More](#)

Description:

** REVISED MARCH 2017 **

SUMMARY OF KEYSTROKES USED FOR IN-QUEUE RECORD MARKING

** MouseLeft Mark record in single record mode.

** ShiftMouseLeft Mark end of range between Selected and row dicked.

** CtrlShiftMouseLeft Continues marking from LastSelected to dicked position.

** CtrlMouseLeft Toggle record marked/unmarked.

** CtrlSpace Same effect as CtrlMouseLeft.

** ShiftUp Mark next record up from selected.

** ShiftDown Mark next record down from selected.

** UpKey Switch to single record mode and move cursor up.
** DownKey Switch to single record mode and move cursor down.
** CtrlUp Move cursor up with no marking or unmarking.
** CtrlDown Move cursor dn with no marking or unmarking.
** ShiftPgDn Mark all from selected to top of current page.
** ShiftPgUp Mark all from selected to end of current page.
** ShiftHome Mark all from selected to first record.
** ShiftEnd Mark all from selected to last record.
** HomeKey Unmark all and select first record.
** CtrlHome Unmark all and select first record.
** EndKey Unmark all and select last record.
** CtrlEnd Unmark all and select first record.
** PgUpKey Unmark all and select last record on page.
** PgDnKey Unmark all and select first record on page.
** CtrlA Select all
** CtrlH "H" how many selected.

[Back To Previous](#)

Class: [HNDMarkerBrowse](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#) [HNDFileLoadBrowse](#) [HNDBrwFilter](#) [HNDParse](#)
Compositions: [HNDUtil](#)
Type: Browse Building Classes
Demos: HND SCHOOL.APP, HND3.APP, HNDMARKERSQLTEST.APP, Many_More
DLL: None
TLB: None
Header: HNDMRKBR.INC
Code: HNDMRKBR.CLW
Template: [HandyMarkerBrowse](#) [ListBoxBrowseExtender](#) [HandyWebClientBrowseProcedure](#)
[ClientServerBrowseListBox](#) [ClientServerBrowse_XCL](#) [More](#)
Description:

*** REVISED JANUARY 2017 ***

The CHT Class, **HNDMarkerBrowse** derives from **HNDFileLoadBrowse**, **HNDBrwFilter**, and **HNDParse** to implement fully queryable, ABC-Browses extended with single and multi *record-marking* capability.

Note that CHT distinguishes between "*record-marking*" and "*record-tagging*", both of which are implemented in various CHT templates and examples. *Record-marking* as delivered by this **HNDMarkerBrowse** class and its hosting **HNDMarkerBrowse** template, marks records in the browse queue using the built-in "Mark" field supplied in that queue by the ABC Browse template when it generates browse code from your design.

Browse-queue record markers as implemented in this **HNDMarkerBrowse** class, are therefore, temporary and are not reflected in the data file being browsed. This style of "record-marking" is not drive or network intensive as there is no "chatter" between the client browse and the data base tables while markers are being placed.

Record-Tagging on the other hand, writes tags into the data file and while these *are* recallable from one browse instance to the next, also requires ongoing "chatter" between the client browse and the data base as record tags are written into the data table(s).

This low-chatter characteristic of CHT file-loaded, record-markable browses, delivered by this **HNDMarkerBrowse** class, also makes this record-marking technique - *derived from its parent class, HNDFileLoadBrowse* - a perfect candidate for *across-the-web Client-Server browses* as delivered by *ClientServerBrowse_XCL* and its set of related templates.

SUMMARY OF KEYSTROKES USED FOR HANDYMARKERBROWSE IN-QUEUE RECORD MARKING

** MouseLeft Mark record in single record mode.
** ShiftMouseLeft Mark end of range between Selected and row dicked.
** CtrlShiftMouseLeft Continues marking from LastSelected to dicked position.
** CtrlMouseLeft Toggle record marked/unmarked.
** CtrlSpace Same effect as CtrlMouseLeft.
** ShiftUp Mark next record up from selected.
** ShiftDown Mark next record down from selected.
** UpKey Switch to single record mode and move cursor up.
** DownKey Switch to single record mode and move cursor down.
** CtrlUp Move cursor up with no marking or unmarking.

** CtrlDown Move cursor dn with no marking or unmarking.
** ShiftPgDn Mark all from selected to top of current page.
** ShiftPgUp Mark all from selected to end of current page.
** ShiftHome Mark all from selected to first record.
** ShiftEnd Mark all from selected to last record.
** HomeKey Unmark all and select first record.
** CtrlHome Unmark all and select first record.
** EndKey Unmark all and select last record.
** CtrlEnd Unmark all and select first record.
** PgUpKey Unmark all and select last record on page.
** PgDnKey Unmark all and select first record on page.
** CtrlA Select all
** CtrlH "H" how many selected.

[Back To Previous](#)

[Buffer, File And Memory Compression](#)

Class: **HNDCompress** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: Buffer, File And Memory Compression
Demos: None
DLL: None
TLB: None
Header: HNDCOMPRESS.INC
Code: HNDCOMPRESS.CLW
Template: [None](#)
Description:

*** REVISED JANUARY 2017 ***

The CHT HNDCompress class, contained in *HNDCOMPRESS.INC* (Header module) and *HNDCOMPRESS.CLW* (Code module) provides string, buffer and memory compression directly.

This class is based on Clarion's relatively recent COMPRESS() and DECOMPRESS() built-in commands and is designed to mimic the function of CHT's HNDsqueeze class to provide an alternative source for a variety of compression services that are more native to Clarion itself.

HNDCompress, as of July 2015 replaces HNDsqueeze (HNDQUEUEZ.INC / HNDQUEUEZ.CLW). The HNDQUEUEZ library is henceforth redundant, although it will continue to be shipped with CHT until we're certain no component or future endeavor will require it. While they both work exactly the same and are interchangeable, the HNDQUEUEZ library requires external DLL HNDZLBEX.DLL to be linked dynamically into the app. The HNDCOMPRESS library, as stated above use Clarion's Native functions COMPRESS() AND DECOMPRESS(). Anytime we can lower the .DLLs to be included burden of an app (particularly installers) we'll choose to do that.

[Back To Previous](#)

[Buffer, File And Memory Compression](#)

Class: **HNDsqueeze** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDBuffer](#)
Type: Buffer, File And Memory Compression
Demos: [HNDZTEST.APP](#) [HNDZDEMO.APP](#)
DLL: hndzlbex.dll
TLB: None
Header: HNDQUEUEZ.INC
Code: HNDQUEUEZ.CLW
Template: [IncludeHNDOSFileFunctions](#) [IncludeHNDiskFunctions](#) [EmbedDiskFunctions](#)
Description:

*** REVISED JANUARY 2017 ***

The CHT *HNDSqueeze* class, contained in *HNDSQUEEZ.INC* (Header module) and *HNDSQUEEZ.CLW* (Code module) provides file and memory compression.

File compression is accomplished via inheritance by CHT class *HNDDOSFILE* which can be attached globally to an application with template *IncludeHNDDOSFileFunctions*. Memory compression is accomplished via inheritance by CHT class *HNDDISK* which can be attached globally to an application with template *IncludeHNDDiskFunctions* or procedure-locally using *EmbedDiskFunctions*.

[Back To Previous](#)

Burn CDROM Disks

Class: **HNDCDROMClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDUtil](#) [HNDError](#)
Compositions: [HNDSrLib](#) [HNDComLibrary](#)
Type: Burn CDROM Disks
Demos: [HNDCDROMTEST.APP](#) [HNDCDROMEXAMPLE.APP](#)
DLL: IMAPI2FS.DLL (WINDOWS_DLL)
TLB: HNDCDROM.TLB
Header: HNDCDROM.INC
Code: HNDCDROM.CLW
Template: [EmbedHNDCDROMClass](#)
Description:

This class is built on MS Windows capabilities deriving from its IMAPI2FS.DLL. It enables your app to programmatically copy files and directories to a CD/DVD Drive containing a variety of CD media types. This functionality is contained in modules HNDCDROM.INC, HNDCDROM.CLW and HNDCDROM.TLB.

Apply HNDCDROMClass to your procedure with CHT template EmbedHNDCDROMClass.

[Back To Previous](#)

CHT Snap-In Clarion Interface

Class: **HNDSnapCall** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#) [HNDUtil](#)
Compositions: [HNDCaptureConsole](#) [HNDBinFileIO](#) [HNDConfig](#)
Type: CHT Snap-In Clarion Interface
Demos: [HNDSNAPTTEST.APP](#) [HNDETESTCHTSNAPGET.APP](#) [HNDETESTCHTSNAPTOOLS.APP](#)
[HNDETESTCHTSNAPZIP.APP](#) [HNDEZIPNEMAILSNAP](#) [HNDEZIPNEMAILSNAPMAPI.APP](#)
DLL: HNDAPCON.DLL or HNDNETCAPCON.DLL
TLB: None
Header: HNDNAPCALL.INC
Code: HNDNAPCALL.CLW
Template: [EmbedSnapCall](#)
Description:

*** REVISED JANUARY 2017 ***

The library, is being continually forward-developed to handle calls to existing, as well as in-development "CHT Snap-Ins" (*C# executables*) and "Batch Bots" (*Clarion executables*) of various kinds.

The base template to attach this class to your Clarion procedure is *EmbedSnapCall*. This CHT class interfaces to the following Snap-In Items:

(1) CHTSNAPSEND.EXE, (2) CHTSNAPEDIT.EXE, (3) CHTSNAPSMTP.EXE,
(4) CHTSNAPMAPI.EXE, (5) CHTSNAPGET.EXE, (6) CHTSNAPZIP.EXE.

This CHT class interfaces to the following Batch-Bot Items:

(1) HNDMAILSMTP.EXE, (2) HNDMAILMAPI.EXE, (3) HNDMAILCDO.EXE,
(4) HNDSLFCMD.EXE, (5) HNDSLFCMDXSA.EXE, (6) HNDBASE64CVT.EXE,
(7) HNDENCRYPT.EXE, (8) HNDFTPNET.EXE, (9) HNDMAILNET.EXE,
(10) HNDMAKEHZO.EXE, (11) HNDZIPNET.EXE.

Future C#-Snap-In items will include:

CHTSNAPFTP.EXE, CHTSNAPSMS.EXE, CHTSNAPPUT.EXE.

See: [Batch Bot Apps](#)

[Back To Previous](#)

COM Utility Library

Class: **HNDComLibrary** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: COM Utility Library
Demos: [HNDCOMZIPPER.APP](#) [HNDNETFTPDEMO.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDCOMLB.INC
Code: HNDCOMLB.CLW
Template: [None](#)
Description:

*** REVISED JULY 2016 ***

HNDComLibrary is a COM-related library included as a function source in all other CHT libraries that interface to COM DLLs or COM OCXs. This library is not directly hooked into Clarion procedures via a template of its own. It is included as a source file by other COM-related libraries and is therefore not intended to be inserted into developers' procedures to provide services of the sort he/she might get from HNDDisk, for instance.

Examples of libraries using HNDComLibrary are: HNDCOMWapiControl, HNDCOMSlider, HNDCOMZipClass and many others. The demo applications cited are again using HNDComLibrary indirectly.

[Back To Previous](#)

Calculators

Class: **HNDCalcControl** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDAppIO](#)
Type: Calculators
Demos: [HNDCALCULATOR.APP](#)
DLL: None
TLB: None
Header: HNDCALC.INC
Code: HNDCALC.CLW
Template: [EmbedCalculatorFunctions](#) [HandyCalcEntryControl ECF](#)
Description:

*** REVISED JULY 2017 ***

HNDCalcControl provides two different calculator formats that may be called via a simple function in your application. A small, basic calculator is started with OBJ.HandyCalc(). A larger, more graphically sophisticated calculator, with an interchangeable skin, and a scrolling tape, is started with OBJ.HandyCalcTape().

[Back To Previous](#)

Capture DOS Console

Class: **HNDCaptureConsole** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: Capture DOS Console
Demos: [HNDCMP.APP](#)
DLL: HNDCAPCON.DLL

TLB: None
Header: HNDCAPCON.INC
Code: HNDCAPCON.CLW
Template: [EmbedHNDCaptureConsole](#)
Description:

*** REVISED JANUARY 2017 ***

A really good example of a "DOS Console Application" with which all Clarion developers are, or at least should be, familiar is CLARIONCL.EXE. From the DOS prompt, hand typed instructions on the "Command-line" or more likely, using a batch file, it is possible to generate, compile and even execute specific templates in the application. "Console" applications of this sort, output execution information to the "DOS Console", the normally black window that constitutes a direct interface to DOS.

With CHT's HNDCaptureConsole class, it is possible to "Capture" the DOS console in such a way that the black, DOS Console Window does not appear, and the execution output that would normally appear in that window is "Captured" and returned to your Clarion application as a string return value.

Another good example of how this class can be used by a standard Clarion application is HNDCMP.APP, better known as "CHT Clarion Project Builder". This application, in fact, runs CLARIONCL.EXE in a "captured" manner, and returns the full output that would normally appear in the Console window and stores it in an application YOURAPPNAMEHERE.LOG file.

As of January 2017 this code module begins to also optionally utilize HNDNETCAPCON.DLL, a .NET DLL that parallels the C++ DLL HNDCAPCON.DLL. At present only the function HNDCaptureConsole.RUNAPI() utilizes this DLL. In the long run, only HNDNETCAPCON.DLL will be employed and HNDCAPCON.DLL will become deprecated. This will have no impact on your use of this code module as the procedure signatures will stay the same as we make this transition. The templates that connect this class to your application will make sure that the correct DLL is copied to your application and the application .SHP file will advise the correct DLL name required by your application.

[Back To Previous](#)

Class: [HNDClarionDDE](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#) [HNDDisk](#)
Compositions: [HNDCaptureConsole](#)
Type: Capture DOS Console
Demos: [HNDCMP.APP](#)
DLL: HNDCapCon.DLL
TLB: None
Header: HNDCLDDE.INC
Code: HNDCLDDE.CLW
Template: [EmbedClarionDDEFunctions](#)
Description:

*** REVISED NOVEMBER 2014 ***

When the Clarion IDE moved from C6.3 to C7 and up, DDE (Dynamic Data Exchange) was dropped as the mechanism of outside control of the Clarion IDE from another application. This class was, at that time, re-oriented from DDE control of Clarion, to command-line control of Clarion.

Thus HNDClarionDDE, despite the "DDE" embedded in its name, incorporates CHT HNDCaptureConsole to interface from HNDCMP.APP, the *CHT Clarion Project Builder*, to CLARIONCL.EXE, in order to implement at the DOS Console level, command and control of various functionalities related to Compiling, Generating and Template Execution, provided by CLARIONCL.EXE.

This class is only used by HNDCMP.APP, as is *EmbedClarionDDEFunctions*, its hosting template. This class also incorporates *HNDDisk* from which it derives various low-level, disk-related capabilities such as writing out text-based .LOG files in which are stored the execution details returned from HNDCaptureConsole.Run().

[Back To Previous](#)

Class: [HNDNetCapConClass](#) [Click For Methods Listing](#)
Status: Current

Parent: [HNDError](#)
Compositions: [HNDStrLib](#) [HNDComLibrary](#)
Type: Capture DOS Console
Demos: [Internal Only at Present](#)
DLL: hndnetcapcon.dll
TLB: hndnetcapcon.tlb
Header: HNDNETCAPCON.INC
Code: HNDNETCAPCON.CLW
Template: [Internal Only at Present](#)
Description:

*** NEW JULY 2016 ***

This class is slated to eventually replace the functionality in the long-time CHT class called *HNDCaptureConsole*, which is used to console-capture the output from DOS-COMMAND-LINE executables like CHTSNAPZIP.EXE and numerous others. The brand new DLL, HNDNETCAPCON.DLL is a C# net dll that does the same thing as the older, C++ HNDCAPCON.DLL, a dll which we expect will eventually become deprecated. We plan this deprecation to take place, not abruptly, but in a gradual way which slowly replaces the use of HNDCaptureConsole class with this HNDNetCapConCls and phases out the HNDCAPCON.DLL in a gradual, manner.

[Back To Previous](#)

[Coding Assistance](#)

Class: [HNDAppHook](#) [Click For Methods Listing](#)
Status: Current
Parent: None
Compositions: None
Type: Coding Assistance
Demos: [HNDSETUP.APP](#)
DLL: None
TLB: None
Header: HNDAPPHK.INC
Code: HNDAPPHK.CLW
Template: [EmbedAppHookFunctions](#)
Description:

*** REVISED MARCH 2016 ***

This class is unusual because, in and of itself, the code module contains no code. All methods in this class are intended for overriding in the application procedure, hence the name "App Hook". See HNDSETUP.APP for an example use of HNDAppHook.

This class hooks into any standard Clarion procedure via template - EmbedAppHookFunctions - and provides named (though generic) embed points that can be made to perform almost any function required by the application.

Normally, hand-embedding into these functions is the most likely use, but conceivably, a template could embed functionality requested by the developer as well.

[Back To Previous](#)

Class: [HNDShellAutoComplete](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: Coding Assistance
Demos: [HNDCLIENTSV.APP](#) [HNDFILSV.APP](#) [More](#)
DLL: SHLWAPI.DLL (WINDOWS_DLL)
TLB: None
Header: HNDAUTOC.INC
Code: HNDAUTOC.CLW
Template: [EmbedShellAutoCompleteFunctions](#)
Description:

*** REVISED NOVEMBER 2014 ***

This single-purpose class assists developers and users with remembering legally-used, and reusable, input values for a form field. When applied to an input control of type

TEXT or ENTRY, it behaves like the URL address field in Microsoft Internet Explorer. Recently entered values that have been entered into similar fields present themselves in a dropdown for easy selection. Available settings are:

- ** SetInputFile() pop up a list of files and/or paths
- ** SetInputUrl() pop up a list of web addresses or URLs
- ** SetInputAny() pop up a list of either of the above

Good example applications to examine and test this with, are HNDCLIENTSV.APP and HNDMTSNG.APP. The "Server Run Directory" field has been set as SetInputAny(). Empty this field and begin to type a path and by the time you type the 3rd character, a dropdown appears to allow selection of previously entered data from similar fields on your PC.

[Back To Previous](#)

[Data Access View To Queue](#)

Class: [HNDDataAdapter](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDBuffer](#)
Type: Data Access View To Queue
Demos: [HNDSRPRO.APP](#) [HNDSRPT.APP](#)
DLL: None
TLB: None
Header: HNDRADAP.INC
Code: HNDRADAP.CLW
Template: [EmbedHNDDataAdapterFunctions](#) [HandySourceInLineProcess](#) [HSP](#)
Description:

*** REVISED NOVEMBER 2014 ***

HNDDataAdapter is a class, that in an abstract manner can read through a CLARION VIEW, with or without a view filter, and insert the data from that view into a queue. The target queue may be a basic queue containing data fields only, or a queue containing also color information fields as well as standard data fields.

The demo application HNDSRPRO.APP illustrates several browses that are based simply on a CLARION QUEUE without a ABC browse class below them. The queue is managed by a *ListboxBrowseExtender* template, and looks for all intents an purposes, like a HandyMarkerBrowse. The queue is filled by a process which uses HNDDataAdapter to traverse one or more (joined) files by means of a Clarion View.

[Back To Previous](#)

[Dates, Times And Calendars](#)

Class: [HNDDates](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: Dates, Times And Calendars
Demos: [HNDCAL.APP](#)
DLL: None
TLB: None
Header: HNDDATES.INC
Code: HNDDATES.CLW
Template: [HandyPopUpCalendarButton](#) [HandyPopUpCalendar](#) [EmbedDateFunctions](#) [EmbedObject](#)
Description:

*** REVISED JANUARY 2017 ***

CHT HNDDates class is a full-function dates and times library that contains an amazing amount of functionality including, for example, two pop-up calendars with customizable, interfaces for North American, European, and Arabian-style calendar layouts.

Numerous HNDDates methods provide time/date operations that can perform date differentiation by time zone, leap year lculations, date and time wrap calculations, Julian dates, UTC dates, business day calculations and much more. This is the go-to library to manage dates and times in any Clarion application. Example application: HNDCAL.APP which applies this class to a Clarion Window using CHT template

EmbedDateFunctions.

[Back To Previous](#)

[Dates, Times And Calendars](#)

Class: [HNDNetDateTimeClass](#) [Click For Methods Listing](#)

Status: Current

Parent: [HNDError](#)

Compositions: [HNDStrLib](#) [HNDComLibrary](#)

Type: Dates, Times And Calendars

Demos: [HNDNETDATETIMEDEMO.APP](#)

DLL: HNDNETDATETIME.DLL

TLB: HNDNETDATETIME.TLB

Header: HNDNETDATETIME.INC

Code: HNDNETDATETIME.CLW

Template: [EmbedNetDateTimeClasses](#)

Description:

*** REVISED JANUARY 2017 ***

The functionality in this class comes up from C# via a CHT COM dll called HNDNETDATETIME.DLL, which used with this class makes the DLL self-registering. If the DLL is not already registered with Windows COM at the time an app utilizing this class+dll it is registered with COM, causing the app to pause briefly and barely noticeably for a second or two and then continue running as usual. Once found registered on the running hardware, the auto-registration code in this class is skipped.

This library provides a set of low-level functions like GetFileDateUTC(), GetDayOfYear(), GetIsDaylightSavingTime(), GetIsLeapYear(), GetTimeZone() and 25 more that assist developer with date and time display and manipulation. See example: HNDNETDATETIMEDEMO.APP which applies this class to a Clarion window using template: EmbedNetDateTimeClasses.

Added WeekOfYearISO() in January 2017 to match the WeekOfYear() ISO implementation in HNDDATES.INC/.CLW

[Back To Previous](#)

[Dot NET - File Compression](#)

Class: [HNDNetZipperClass](#) [Click For Methods Listing](#)

Status: Current

Parent: [HNDError](#)

Compositions: [HNDStrLib](#) [HNDComLibrary](#)

Type: Dot NET - File Compression

Demos: [HNDNETZIPDEMO.APP](#) [HNDZIPNET.APP](#) [HNDZIPEMAILNET.APP](#) [More](#)

DLL: HNDNETZIPPER.DLL

TLB: HNDNETZIPPER.TLB

Header: HNDNETZIPPER.INC

Code: HNDNETZIPPER.CLW

Template: [HandyNetZipperControl](#) [EmbedNetZipFunctions](#)

Description:

*** LAST REVISED MARCH 2016 ***

This class provides a wrapper for a CHT-developed .NET DLL called HNDNETZIPPER.DLL. The .DLL in combination with this wrapper class provide a full range of standard file zipping/unzipping services including bulk and single file operations. The .DLL requires a window with two messaging controls in order to operate. If you require invisible or background operations, hide or minimize the window hosting the HNDNETZIPPER.DLL.

A template called HandyNETZipperControl may be used to add this control to your application window. Demonstration apps are: HNDNETZIPDEMO.APP, HNDZIPNET.APP

[Back To Previous](#)

[Email MAPI](#)

Class: [HNDComMapiControl](#) [Click For Methods Listing](#)

Status: Current
Parent: [HNDError](#)
Compositions: [HNDStrLib](#) [HNDComLibrary](#)
Type: Email MAPI
Demos: [HNDEZMAPI.APP](#)
DLL: MSMAPI32.OCX
TLB: HNDMAPI.TLB
Header: HNDMAPI.INC
Code: HNDMAPI.CLW
Template: [HandyCOMMapiControl](#)
Description:

*** REVISED NOVEMBER 2014 ***

This class serves as a COM wrapper for a Microsoft COM DLL called MSMAPI32.OCX. It provides basic MAPI connectivity for send/receive services using your installed email client (Outlook or Outlook Express). Developers may optionally utilize the email client's native interface or provide a custom interface of their own design.

Sending operations provide for TO, CC and BCC, single or multiple attachments and multiple addresses separated by semi-colons. The only real requirement to get this going is a working, correctly configured email client on the host machine. Example application(s): HNDEZMAPI.APP. Template(s): HandyComMapiControl.

[Back To Previous](#)

Class: **HNDIMail** [Click For Methods Listing](#)
Status: Current
Parent: [HNDIDial](#)
Compositions: [HNDDisk](#) [HNDBuffer](#) [HNDRegistry](#) [HNDEncrypt](#)
Type: Email MAPI
Demos: [HNDMAILSEND.APP](#) [HNDSOURCEMAPIDEMO.APP](#) [HNDOUTLOOK.APP](#)
DLL: None
TLB: None
Header: HNDIMAIL.INC
Code: HNDIMAIL.CLW
Template: [HandyMapiEmail](#) [EmbedMapiEmail](#) [More](#)
Description:

*** REVISED JUNE 2015 ***

This class handles MAPI Email connectivity to the default email client installed on your system. Since MAPI Email is a Microsoft standard, to which not all email clients adhere, MAPI is restricted to the capabilities provided by the host email client(s) installed. However, when MS Windows Live is the default email client, and to a lesser extent MS Outlook, there is no easier way to implement email sending and receiving in your applications than using MAPI connectivity as implemented in this HNDIMail class.

HNDIMail is leveraged by a variety of CHT templates such as InstantOutlookMail, and Jumpstarts such as "MAPI Send With Attachments". It includes attachment compression and encryption, capabilities. The developer can introduce his/her own user interface or leverage the email client's interface to implement email sending and retrieving functionality.

[Back To Previous](#)

Class: **HNDNetMAPIClass** [Click For Methods Listing](#)
Status: New -- Experimental
Parent: [HNDError](#)
Compositions: [HNDStrLib](#) [HNDComLibrary](#)
Type: Email MAPI
Demos: [HNDMAILMAPI.APP](#)
DLL: HNDNETMAPI.DLL
TLB: HNDNETMAPI.TLB
Header: HNDNETMAPI.INC
Code: HNDNETMAPI.CLW
Template: [EmbedNetMAPIClasses](#)
Description:

*** REVISED NOVEMBER 2014 ***

This class is designed to eventually replace the HNDComMapiControl class which uses the Microsoft MSMAPI32.OCX. At a future date, HNDComMapiControl will become deprecated and this class - *HNDNetMapiClass* - will replace it to handle simple MAPI-send applications. This class is based on a DLL called HNDNETMAPI.DLL which CHT has built with C# and which connects to Clarion via COM-interop.

[Back To Previous](#)

Email SMTP

Class: [HNDNetSMTPClass](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDConfig](#) [HNDError](#)
Compositions: [HNDStrLib](#) [HNDComLibrary](#) [IHNDNetSMTP](#)
Type: Email SMTP
Demos: [HNDMAILNET.APP](#) [HNSMTPNETDEMO.APP](#) [HNDZIPNEMAILNET.APP](#) [More](#)
DLL: HNDNETSMTP.DLL
TLB: HNDNETSMTP.TLB
Header: HNDNETSMTP.INC
Code: HNDNETSMTP.CLW
Template: [HNDNetSMTPClass](#)
Description:

*** REVISED JUNE 2015 ***

This class handles SMTP Email sending, standard and SSL. It leverages CHT COM DLL, HNDNETSMTP.DLL. The revised template now pre-configures your setup to ask for SMTP parameters when non are found in the registry. As soon as valid SMTP parameters including SMTPUserName, SMTPPassWord, SMTPServer, SMTPPort, UseSSL, and SMTPReturnAddress are found the template functionality stops nagging for a correct configuration. It's easy to set up multiple SMTP configurations on the same hardware. For an example of that look at HNDZIPNEMAILNET.APP.

HNDNetSMTPClass is leveraged by a variety of CHT templates such as HNDNetSMTPClass, and Jumpstarts such as "SMTP Configure", "SMTP Net Send Basic". It includes attachment compression and encryption, capabilities. The developer can introduce his/her own user interface or leverage the many example applications provided by CHT.

[Back To Previous](#)

Class: [HNDSMTP](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDConfig](#) [HNDError](#)
Compositions: [HNDBuffer](#) [HNDDisk](#) [HNDEncrypt](#) [HNDRegistry](#)
Type: Email SMTP
Demos: [HNDZIPNEMAILSMTP.APP](#) [HNDMAILSMTP.APP](#) [HNSMTPNETDEMO.APP](#) [HNDMAILSMTP.APP](#) [More](#)
DLL: CLARUNEXT.DLL
TLB: None
Header: HNDSMTP.INC
Code: HNDSMTP.CLW
Template: [EmbedSMTPFunctions](#) [Source EmbedSMTPFunctions](#) [More](#)
Description:

*** REVISED MAY 2016 ***

This class handles SMTP Email sending, standard and SSL. It leverages Clarion DLL, CLARUNEXT.DLL which exists in both the run-time set provided by Clarion Professional and Clarion Enterprise. The revised template now pre-configures your setup to ask for SMTP parameters when non are found in the registry. As soon as valid SMTP parameters including SMTPUserName, SMTPPassWord, SMTPServer, SMTPPort, UseSSL, and SMTPReturnAddress are found the template functionality stops nagging for a correct configuration. It's easy to set up multiple SMTP configurations on the same hardware. For an example of that look at HNDZIPNEMAILSMTP.APP.

HNDSMTP is leveraged by a variety of CHT templates such as EmbedSMTPFunctions, and Jumpstarts such as "SMTP Configure", "SMTP Send With Attachments" and more. It includes attachment compression and encryption, capabilities. The developer can introduce his/her own user interface or leverage the many example applications

provided by CHT.

[Back To Previous](#)

Embedding Assist

Class: **HNDSrcProcedure** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: Embedding Assist
Demos: [HANDYBULKMAILBATCHER.APP](#) [HNDMAILSMTP.APP](#)
DLL: None
TLB: None
Header: HNDSRC.INC
Code: HNDSRC.CLW
Template: [HandySourceProcedure](#)
Description:

This class is introduced into your application by a procedure template called *HandySourceProcedure*. Our template and the resulting procedure is a replacement for ABC's native source procedure. The problem, as we see it, with ABC's native source procedure is a lack of embed points into which add-on templates may be added.

While it may seem counter-intuitive to expect template embed points in a "source" procedure, it is perfectly normal and "par-for-the-course" to combine template-embedded code with hand-embedded code in all kinds of situations. So why is this not provided in the ABC native source procedure?

With that thought in mind, this class, *HNDSrcProcedure*, and its procedure template *HandySourceProcedure* provide both template embed points (*i.e. points into which template-provided code may be placed*) as well as hand embed points (*i.e. points into which hand-provided code may be placed*).

Additionally, along the lines of providing a maximum of plug-in functionality from other CHT classes, into the source procedure that results from creating a source procedure the CHT way, we've provided a group of related templates, the names of which begin with "Source_", for example, *Source_EmbedDiskFunctions*.

The following is a comprehensive list of related templates:
Source_EmbedDiskFunctions, *Source_EmbedDateFunctions*,
Source_EmbedDiskExFunctions, *Source_EmbedEncryptionFunctions*,
Source_EmbedHZOFunctions, *Source_EmbedMAPIFunctions*,
Source_EmbedSMTPFunctions, *Source_EmbedUtilityFunctions*,
Source_EmbedZipFunctions, *Source_EmbedQueryBuilderFunctions*

[Back To Previous](#)

Error Handler

Class: **HNDError** [Click For Methods Listing](#)
Status: Current
Parent: None
Compositions: None
Type: Error Handler
Demos: [All](#)
DLL: None
TLB: None
Header: HNDERROR.INC
Code: HNDERROR.CLW
Template: [All](#)
Description:

*** REVISED JANUARY 2017 ***

This class is derived directly or indirectly by all CHT classes. It acts as an error manager/messenger and (optionally) as an application status logging manager. When using CHT classes, critical method calls may be followed by an `OBJ.GetLastErrorCode()` call to obtain an error number. Zero indicates no error. To obtain an error message call

instead OBJ.GetLastError()). A blank message indicates no error. To have the error displayed unconditionally call OBJ.ShowLastError(True). To display an error conditionally based upon the current status condition of property OBJ.Verbose, call OBJ.ShowLastError().

To see status logging in action, study the application called HNDMTSNG.APP in which all server transactions are logged to a text control populated on the server window. The information in that status text control is managed by this HNDError class from inside the server classes which derive HNDError. In deeply nested classes the error manager is normally the base class from which the rest derive. Error numbers and error messages bubble up to the topmost, last derived class, from classes lower down in the heirarchy. This makes error determination easy, since the developer can poll the topmost class instance - the one attached to his/her Clarion procedure - for an error number or message to determine a problem, no matter how deep inside the CHT classes that error may have occurred.

[Back To Previous](#)

Experimental

Class: HNDNetLinkSmtip [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDRegistry](#) [HNDDisk](#)
Type: Experimental
Demos: [Internal Only at Present](#)
DLL: hndnetlinksmtp.dll
TLB: None
Header: HNDNETLINK.INC
Code: HNDNETLINK.CLW
Template: [Internal Only at Present](#)
Description:

*** NEW JULY 2016 ***

This library is under construction. This class is an experimental net-dll-related set of functions which test the feasibility of creating CHT net DLLs which expose a dual interface. 1) A standard function export interface which can be linked to. 2) A COM interface which can be instantiated to connect via Clarion COM. For the time being we won't say more about this, other than to acknowledge that a single experimental dll, called HNDNETLINKSMTP.DLL exists which exports a single function GetCHTForumLogin() successfully. The medium-term objective of this experiment is to further expand this DLL to work both as a standard linkable DLL and as a COM DLL. More information will appear here as the experiment continues.

[Back To Previous](#)

Extended Controls - Animation

Class: HNDAnimate [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#) [HNDDiskEx](#)
Compositions: None
Type: Extended Controls - Animation
Demos: [HNDCPYDM.APP](#)
DLL: None
TLB: None
Header: HNDANIMA.INC
Code: HNDANIMA.CLW
Template: [EmbedObject](#)
Description:

*** REVISED MARCH 2016 ***

This class provides a variety of on-screen animations used most often by developers as an indication of on-going activity or to replace or enhance a progress bar. It encapsulates MS Windows' built-in animation control for easy use on your application windows. See HNDCPYDM.APP for examples.

HNDAnimate is optionally passed into HNDDiskEx to provide animation services on wait

windows for a variety of bulk file processing activities. It may also be used alone on any Clarion window with the .AM of your choice. This class does not have a direct template to implement its functionality. It, instead, is used inside of HNDDISKEX via composition to create and "animation object" that's capable of displaying short animations on a window -- for example: hndfilecopy.avi.

It may be added to a procedure (as can any CHT Class) via the generic class-attachment template called "EmbedObject". But any template that incorporates HNDDiskEx (See HNDDiskEx) will make this class available to your procedure.

[Back To Previous](#)

[Extended Controls - Animation](#)

Class: [HNDComAnimation](#) [Click For Methods Listing](#)

Status: Current

Parent: [HNDError](#)

Compositions: [HNDStrLib](#) [HNDComLibrary](#)

Type: Extended Controls - Animation

Demos: [HNDCPYDM.APP](#)

DLL: MSCOMCT2.OCX

TLB: HNDCMANI.TLB

Header: HNDCMANI.INC

Code: HNDCMANI.CLW

Template: [HandyCOMAnimationControl](#)

Description:

*** REVISED NOVEMBER 2014 ***

CHT HNDComAnimation class provides a control interface for Clarion windows in which to display a variety of .AM files to provide WAIT and PROGRESS displays while the window is busy performing some task.

The .AM file designated to be played on the window (*see \hndapps\avi\ for examples*) is determined by the developer, everything from flying folders for copying files to folders sailing neatly into the Windows Recycle bin.

[Back To Previous](#)

[Extended Controls - Calendars](#)

Class: [HNDDatePicker](#) [Click For Methods Listing](#)

Status: Current

Parent: [HNDError](#)

Compositions: [HNDStrLib](#) [HNDComLibrary](#)

Type: Extended Controls - Calendars

Demos: [HNDDTPK.APP](#)

DLL: MSCOMCT2.OCX

TLB: HNDDTPKR.TLB

Header: HNDDTPKR.INC

Code: HNDDTPKR.CLW

Template: [HandyCOMDatePickerControl](#)

Description:

*** REVISED NOVEMBER 2014 ***

CHT HNDDatePicker Class provides a really functional drop-down calendar control, native to the Windows operating system, and delivered via MSCOMCT2.OCX COM library.

An example application called HNDDTPK.APP displays how to configure the control for a variety of foreground and background settings. A set of handy functions is available to both set and retrieve date values from the drop-down calendar.

[Back To Previous](#)

[Extended Controls - Clocks](#)

Class: [HNDDigitalClockClass](#) [Click For Methods Listing](#)

Status: Current

Parent: [HNDError](#)
Compositions: [HNDDigitalClockControl](#)
Type: Extended Controls - Clocks
Demos: [HNDDIGITALCLOCKDEMO.APP](#)
DLL: HNDDGTL.CLOCK.OCX
TLB: HNDDGTL.TLB
Header: HNDDGTL.INC
Code: HNDDGTL.CLW
Template: [HNDDigitalClockControl](#)
Description:

*** REVISED NOVEMBER 2014 ***

This class activates an OLE/COM control passed in from the window procedure and displays an active, digital clock inside the perimeter of the control. A single Init call gets the clock started. See demo HNDDIGITALCLOCKDEMO.APP. Apply this class to your Clarion Window with a control template called HNDDigitalClockControl.

[Back To Previous](#)

[Extended Controls - Email](#)

Class: [HNDOutlookClass](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDDigitalClockControl](#) [HNDDigitalClockControl](#) [HNDDigitalClockControl](#)
Type: Extended Controls - Email
Demos: [HNDOUTLOOK.APP](#) [HNDOUTLOOKINTERFACEDEMO.APP](#) [HNDOUTLKDEMO.APP](#)
DLL: OUTLCTL.DLL
TLB: HNDOUTTTL.TLB
Header: HNDOUTTTL.INC
Code: HNDOUTTTL.CLW
Template: [None](#)
Description:

*** REVISED NOVEMBER 2014 ***

This class serves as a COM wrapper for a Microsoft COM DLL called OUTLCTL.DLL. CHT's Outlook Control provides a full-feature, user-configurable MS Outlook Email interface that lets users easily control send, receive and view behaviors, and interact with email documents from inside their Clarion application.

There is an implicit assumption, of course, that a working copy of MS Outlook exists on the host system. A demonstration application called HNDOUTLOOK.APP illustrates use of the template and calling HNDOutlookClass Class methods.

[Back To Previous](#)

[Extended Controls - Names And Addresses](#)

Class: [HNDAAddressBox](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDNameBox](#)
Compositions: None
Type: Extended Controls - Names And Addresses
Demos: [HND6.APP](#)
DLL: None
TLB: None
Header: HNDADRBX.INC
Code: HNDADRBX.CLW
Template: [HandyAddressControls](#)
Description:

*** REVISED MARCH 2016 ***

HNDAAddressBox is an address-handling utility that lets your users enter an address into a single, multi-line text field even when the back end data file or table has separate fields for Street, Suite, City, Province/State and Country. Developers can configure this class to direct input into the target data fields of their choice.

When there is any doubt as to the interpretation, or completeness of an input address, a dialog similar to the address dialog in MS Outlook pops up with the components displayed in the target fields, allowing the user to make adjustments if necessary. HNDAddressBox works in two directions so that addresses may be displayed from the data repository in a single, multi-line text field for editing with changes written back to the correct target data table fields.

The list of selectable countries is developer configurable. Thanks to Gene Witherup for his contribution of improvements incorporated to this class. Example application HND6.APP.

[Back To Previous](#)

[Extended Controls - Names And Addresses](#)

Class: [HNDNameBox](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDAppIO](#)
Type: Extended Controls - Names And Addresses
Demos: [HND6.APP](#)
DLL: None
TLB: None
Header: HNDADR BX.INC
Code: HNDADR BX.CLW
Template: [HandyAddressControls](#)
Description:

*** REVISED MARCH 2016 ***

HNDNameBox is a name-handling utility that lets your users enter a name like Mr. Gus M. Creces into a single entry field even when the back end data file or table has separate fields for Title, Salutation, First Name, Initial, and Last Name. Developers can configure this class to direct input into the target data fields of their choice.

When there is any doubt as to the interpretation, or completeness of an input name, a dialog similar to the name dialog in MS Outlook pops up with the components displayed in the target fields, allowing the user to make adjustments if necessary. HNDNameBox works in two directions so that names may be displayed from the data repository in a single field for editing with changes written back to the correct target data table fields.

A variety of name-layout variants are possible using formal or informal name inputs. Title and Salutation dropdowns are configurable for any language and for specialized usage. Example application HND6.APP.

[Back To Previous](#)

[Extended Controls - Sliders And Progress](#)

Class: [HNDComProgressBar](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDStrLib](#) [HNDComLibrary](#)
Type: Extended Controls - Sliders And Progress
Demos: [HNDPROGRESSBAR.APP](#)
DLL: MSCOMCTL.OCX (Microsoft OCX)
TLB: HNDPRGBR.TLB
Header: HNDPRGBR.INC
Code: HNDPRGBR.CLW
Template: [HandyComProgressBarControl](#)
Description:

*** REVISED NOVEMBER 2014 ***

HNDComProgressBar is a Windows COM control progress bar, native to Windows via MSCOMCTL.OCX. Use CHT template HandyComProgressBarControl to drop this control on any Clarion window.

[Back To Previous](#)

[Extended Controls - Sliders And Progress](#)

Class: [HNDComSlider](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDStrLib](#) [HNDComLibrary](#)
Type: Extended Controls - Sliders And Progress
Demos: [HNDSLIDER.APP](#)
DLL: COMCTL32.OCX
TLB: HNDCSLDR.TLB
Header: HNDCSLDR.INC
Code: HNDCSLDR.CLW
Template: [HNDComSliderControl](#)
Description:

*** REVISED MARCH 2016 ***

HNDComSlider is a Windows COM control slider bar, native to Windows via COMCTL32.OCX. Use Control template HNDComSliderControl to drop this control on any Clarion window.

[Back To Previous](#)

[File And Memory Compression](#)

Class: [HNDComZipClass](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDDisk](#)
Compositions: [HNDStrLib](#) [HNDComLibrary](#)
Type: File And Memory Compression
Demos: [HNDCOMZIPPER.APP](#) [HNDZIPNEMAIL.APP](#) [HNDZIPNFTP.APP](#) [More](#)
DLL: HNDZIPR.OCX
TLB: HNDZIPPR.TLB
Header: HNDZIPPR.INC
Code: HNDZIPPR.CLW
Template: [HandyCOMZipperControl](#) [EmbedZipQueryParserFunctions](#) [Source EmbedZipFunctions](#)
Description:

*** REVISED DECEMBER 2016 ***

This class provides a wrapper for a CHT-developed OCX called HNDZIPR.OCX. The OCX in combination with this wrapper class provide a full range of standard file zipping/unzipping services including bulk and single file operations, with optional encryption. The OCX requires a window with two messaging controls in order to operate.

A callback method in this class wrapper posts progress and zip/unzip messages to the controlling window during file zip/unzip operations. If you require invisible or background operations, hide or minimize the window hosting the HNDZIPR.OCX. A template called HandyCOMZipperControl may be used to add this control to your application window.

[Back To Previous](#)

[File And Memory Encryption](#)

Class: [HNDEncrypt](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDBuffer](#) [HNDError](#)
Compositions: [HNDDisk](#) [HNDdosFile](#) [HNDStrLib](#) [HNDComLibrary](#)
Type: File And Memory Encryption
Demos: [HNDXTEST.APP](#) [HNDENCRYPT.APP](#) [HNDFILSV.APP](#)
[HNDFILCL.APP](#) [HNDLIVUP.APP](#) [HNDFFLCLN.APP](#)
DLL: CAPICOM.DLL
TLB: HNDCAPIC.TLB
Header: HNDENCRP.INC
Code: HNDENCRP.CLW
Template: [HEncrypt](#)
Description:

This CHT class is used extensively by several CHT classes, to provide underlying file encryption, and/or memory encryption services to these classes. For example CHT file server apps such as HNDFILSV.APP is able to encrypt files on request from one of its designated file clients: HNDFILCL.APP, HNDLIVUP.APP and HNDFLCLN.APP.

Similarly a CHT data server app such as HNDCLIENTSVLEAN4VIEW.APP, is able to encrypt memory streams of data being sent to its corresponding clients: HNDCLIENTCLEAN4VIEW.APP, HNDCLIENTCLEAN1VIEW.APP.

In combination with the CHT Class HNDCompress, this encryption class is able to package compressed and encrypted files for transport across the web to receiving file clients, as well as compressed and encrypted memory streams across the web to receiving data clients.

[Back To Previous](#)

[File Binary File Access](#)

Class: HNDBinFileIO [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDBuffer](#)
Type: File Binary File Access
Demos: [Most](#)
DLL: None
TLB: None
Header: HNDBINIO.INC
Code: HNDBINIO.CLW
Template: [EmbedHNDBinFileIOFunctions](#)
Description:

*** REVISED OCTOBER 2016 ***

HNDBinFileIO provides binary-level file access, both read and write. Read any file into memory, modify it at byte level and write it back or create binary or text files from scratch.

HNDBinFileIO provides an implementation of memory-mapped files that can be created by one application and read by another, allowing simultaneously-running applications to communicate with one another via an in-memory file that never hits the drive. Provides GUID-creation functionality. HNDBinFileIO is used via composition in these CHT header modules: hndini.inc, hndshell.inc hndspell.inc and hndutil.inc.

[Back To Previous](#)

[File Callback Functions](#)

Class: HNDFileNotify [Click For Methods Listing](#)
Status: Current
Parent: None
Compositions: None
Type: File Callback Functions
Demos: [HNDBACKUPSERVICE A.APP](#) [HNDBACKUPSERVICE B.APP](#)
DLL: None
TLB: None
Header: HNDFILENOTIFY.INC
Code: HNDFILENOTIFY.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

[HZO File Creation](#)

Class: HNDZStore [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)

Compositions: [HNDRegistry](#) FileManager(ABC)
Type: HZO File Creation
Demos: [HNDSETUP.APP](#) [HNDINSTALLMYFILES.APP](#) [HNDSETUPMKR.APP](#)
DLL: HNDZLbex.DLL
TLB: None
Header: HNDZSTOR.INC
Code: HNDZSTOR.CLW
Template: [EmbedHndZStoreFunctions](#)
Description:

*** REVISED NOVEMBER 2014 ***

HNDZStore class is used to insert and extract all types of files for transport and installation purposes into, and out of, a master Clarion .TPS container. These installation container files, are identified by the file extension .HZO. In order to attach HNDZStore to a procedure use *EmbedHndZStoreFunctions*, or you may use simply, *EmbedObject* a generic embed template which can be enrolled to attach any class to a procedure for the purpose of coding and calling into that class' functions.

The best example to study the use of this class is HNDSETUP.APP, also known as "*CHT Installation Containerizer*". Used in conjunction with CHT's file and memory compression class *HNDShrink*, files can be optionally compressed into the target container. And used in conjunction with CHT's encryption class *HNDEncrypt*, files can also be optionally encrypted into the target container via HNDZStore.

HZO installation files created by HNDSETUP.APP, utilizing this HNDZStore class, are the basis for the file installation and transport system used by CHT's Webupdater Toolkit Installer.

[Back To Previous](#)

[INI File And Registry](#)

Class: **HNDAppio** [Click For Methods Listing](#)
Status: Current
Parent: None
Compositions: ICriticalSection(ABC)
Type: INI File And Registry
Demos: [Most CHT Demo Apps](#)
DLL: None
TLB: None
Header: HNDAPPIO.INC
Code: HNDAPPIO.CLW
Template: [EmbedObject](#)
Description:

*** REVISED MARCH 2016 ***

This class is responsible for storing application configuration information to the target location indicated on the Application -> Global Properties -> General -> Non Volatile Settings dialog in the Clarion IDE. Other CHT classes make use of this class to write configuration information (where required) to the target location configured in your application.

HNDAppIO is lower on the pecking order than even the HNDERROR module. Hence it posts no errors. It has no virtuals since it is not meant to be derived. Rather, it is incorporated into other, higher-level modules by composition rather than derivation. Like the HNDError class, it is used ubiquitously, so it is small, simple and thread-safe.

Ten other CHT Classes include HNDAppIO via composition. Consequently, it's functionality is incorporated indirectly into many applications using a variety of CHT Templates without you needing to do anything directly to make use of it. It may be added to a procedure (as can any CHT Class) via the generic class-attachment template called "EmbedObject". CHT header modules that incorporate this class: hndadrbx.inc, hndbrwfl.inc, hndcalc.inc, hndcnt.inc, hnddisk.inc, hndflbrw.inc, hndinetf.inc, hndini.inc, hndscdap.inc, hndutil.inc.

[Back To Previous](#)

[Internet - HTTP Clients](#)

Class: **HNDHttp** [Click For Methods Listing](#)

Status: Current
Parent: [HNDIDial](#)
Compositions: [HNDConfig](#) [HNDDisk](#) [HNDDosFile](#) [HNDEncrypt](#)
[HNDBuffer](#) [HNDAnimate](#) [HndDiskEx](#)
Type: Internet - HTTP Clients
Demos: [HNSLFLC.APP](#) [HNDMTSCL.APP](#) [HNDINSTBA.APP](#) [HNDINSTALLMYFILES.APP](#)
[HNDCLIENTCLEANVIEW.APP](#) [HNDCLIENTCLEANVIEW.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDHTTP.INC
Code: HNDHTTP.CLW
Template: [EmbedHTTPFunctions](#)
Description:

*** REVISED APRIL 2017 ***

This CHT class handles all HTTP-Protocol-based communications from CHT+Clarion Client applications to CHT+Clarion Server applications. There are nearly two-dozen examples (see *partial list above*) of HNDHTTP acting as the HTTP communications system underlying the application. See also a CHT Server document which enumerates CHT servers and clients provided on our website, called [CHTSERVERINVENTORY.HTML](#).

[Back To Previous](#)

[Internet - HTTP Servers](#)

Class: [HNDSubscriptionServer](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#) [HNDBrowserServer](#)
Compositions: [CriticalSection](#)
Type: Internet - HTTP Servers
Demos: [HNSLFSV.APP](#) [HNDMTSNG.APP](#) [HNDCLIENTSV.APP](#) [HNDFILSV.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDSUBSV.INC
Code: HNDSUBSV.CLW
Template: [None](#)
Description:

*** REVISED APRIL 2017 ***

HNDSubscriptionServer derives CHT Server Base Class HNDBrowserServer.

[Back To Previous](#)

[Internet - HTTP Web Clients](#)

Class: [HNDClient](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#) [HNDHttp](#)
Compositions: [HNDAppio](#) [HNDBuffer](#)
Type: Internet - HTTP Web Clients
Demos: [HNDCLIENTCL.APP](#) [HNDCLIENTCLEANVIEW.APP](#) [HNDMTSCL.APP](#)
[HNDFILECLIENTLEAN.APP](#) [HNSLFLC.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDCLNT.INC
Code: HNDCLNT.CLW
Template: [EmbedHTTPClientFunctions](#)
Description:

*** REVISED MARCH 2017 ***

This class is used by all Clarion applications of a "Client" nature that access data, files, or installations across the WEB or WAN. HNDClient derives directly from HNDHTTP which handles low-level web/wan connectivity while HNDCLIENT handles primarily unpacking activities, be that, decompressing, decrypting, insertion and reformatting from data streams into Clarion queues, groups and file data structures for display in Clarion list-box structures or on Clarion forms.

CHT Client Server applications are built on this class, as are the Client Server implementations provided more recently via ListBoxBrowseExtender. The applications listed above as examples are only a sampling of the available applications that apply this CHT HNDCLIENT class in their design.

[Back To Previous](#)

[Internet - HTTP Web Clients](#)

Class: [HNDClientHook](#) [Click For Methods Listing](#)
Status: Current
Parent: None
Compositions: [HNDAppio](#) [HNDClient](#)
Type: Internet - HTTP Web Clients
Demos: [HNDCLIENTCL.APP](#) [HNDCLIENTCLEAN4MEW.APP](#) [HNDMTSCL.APP](#)
[HNDFILECLIENTLEAN.APP](#) [HNDSLFCL.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDCLNT.INC
Code: HNDCLNT.CLW
Template: [EmbedHNDClientHook](#)
Description:

*** REVISED MARCH 2017 ***

This class, as the name suggests, acts as a "hooking mechanism" to to individualize any HNDHTTP/HNDCLIENT class created and connected via another Clarion procedure into a any other Clarion procedure. This has the effect of making a non-global instance of HNDCLIENT available to other procedures to use as if HNDCLIENT was globally instantiated on the app.

Generally, the procedure which instantiates and connects a HNDCLIENT instance is used as an HTTP configuration and connection mechanism, which then is able to pass an already connected instance of HNDCLIENT as a parameter to any other procedure called from there.

CHT's HNDCLIENTHOOK class receives and references this passed in HNDCLIENT instance and makes it fully available and viable on the child procedure without having to re-connect or be re-configured to connect to the data server.

[Back To Previous](#)

[Internet - HTTP Web Servers](#)

Class: [HNDBrowserServer](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDHtml](#) [HNDError](#)
Compositions: [HNDConfig](#) [HNDIPAddress](#) [HNDDisk](#) [HNDBuffer](#) [HNDEncrypt](#) [HNDDates](#) [CriticalSection](#)
Type: Internet - HTTP Web Servers
Demos: [HNDSLFSV.APP](#) [HNDMTSNG.APP](#) [HNDCLIENTSV.APP](#) [HNDFILSV.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDBRWSV.INC
Code: HNDBRWSV.CLW
Template: [EmbedBrowserServer](#)
Description:

*** REVISED SEPTEMBER 2017 ***

HNDBrowserServer, is CHT's base server class. Because this class has evolved tremendously from its first appearance in approximately the year 2000, it is not only "Browser" oriented. Its name is now a total misnomer, since that name is based on its very earliest implementation only.

Consider this class in its presently evolved form, as a CHT's HTTP Server Base Class.

This class is incorporated into every CHT server application by means of a template called *EmbedBrowserServer*, again now a misnomer, but no less practical, as the template that implements CHT's HTTP Server Base Class. All CHT Server applications are founded on this template and this base class.

A child server class named `HNDSUBSCRIPTIONSERVER` derives this server base class, to implement browser-directed server operations, client-directed server operations and various file-directed server operations. The examples listed above are only some of the example applications utilizing this `HNDBrowserServer` class and its hosting template.

[Back To Previous](#)

Internet - JSON

Class: `HNDJSONClass` [Click For Methods Listing](#)

Status: Current

Parent: [HNDBinIO](#) [HNDError](#)

Compositions: None

Type: Internet - JSON

Demos: [HNDCHATSVR.APP](#) [HNDCHATCLNT.APP](#)

DLL: None

TLB: None

Header: `HNDJSON.INC`

Code: `HNDJSON.CLW`

Template: [EmbedHNDSocketEvents](#)

Description:

The purpose of this class is to package outgoing Clarion data structures `GROUPS`, `QUEUES`, `RECORDS`, into standard, delimited JSON strings and to unpackage incoming JSON strings back into Clarion `GROUPS`, `QUEUES` and `RECORDS`.

[Back To Previous](#)

Internet - Sockets

Class: `HNDSocket` [Click For Methods Listing](#)

Status: Current

Parent: [HNDIPAddress](#) [HNDError](#)

Compositions: None

Type: Internet - Sockets

Demos: [HNDCHATSVR.APP](#) [HNDCHATCLNT.APP](#)

DLL: None

TLB: None

Header: `HND SOCKET.INC`

Code: `HND SOCKET.CLW`

Template: [EmbedSocketEvents](#)

Description:

**** REMISED JANUARY 2017 **** This class is a TCP/IP sockets connector and communicator class. It is not standalone and works in conjunction with `CHT` class `HNDSocketEvents`. This class can communicate over raw TCP/IP sockets without requiring HTTP protocol. It does so, using JSON data packages that contain both a data component and a command component. Note that, maximum data package size is limited by equate `HPROP:RcvBufferCmd`. This socket class is initially designed to provide an always-connected socket connection for use with applications such as `HNDCHATSVR.APP` and `HNDCHATCLNT.APP`. Once a client connects, the socket created is not closed until the client logs off or a socket error is encountered at which time the client is disconnected from the server. This behaviour is entirely different than HTTP protocol which does not keep sockets open between the client and the server. Under HTTP, a client makes a request which opens a new, temporary, socket connection with the server and the transaction processed on that socket completes as soon as a response is received by the client from the server.

[Back To Previous](#)

Class: `HNDSocketEvents` [Click For Methods Listing](#)

Status: Current

Parent: [HNDSocket](#) [HNDIPAddress](#) [HNDError](#)

Compositions: [HNDJSONClass](#)

Type: Internet - Sockets

Demos: [HNDCHATSVR.APP](#) [HNDCHATCLNT.APP](#)

DLL: None

TLB: None
Header: HNSOCKET.INC
Code: HNSOCKET.CLW
Template: [EmbedHNSocketEvents](#)
Description:

** REVISED JANUARY 2017 ** This class is a TCP/IP sockets connector and communicator class. It is not standalone and works in conjunction with CHT class HNDIPAddress. See the description of HNSocket to understand how this class works, since HNSocket and HNSocketEvents are part an parcel of one set of CHT socket-connector-communicator functionalities.

[Back To Previous](#)

[Low Level Memory Data Functions](#)

Class: **HNDBuffer** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: Low Level Memory Data Functions
Demos: [HNDMTSNG.APP](#) [HNDMTSSV.APP](#) [HNSLFSV.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDBUFFR.INC
Code: HNDBUFFR.CLW
Template: [EmbedBufferObject](#)
Description:

*** REVISED APRIL 2017 ***

This class provides a means of creating, removing, and even re-sizing large storage memory spaces within your procedures. This eliminates the need to create a pre-sized CSTRING which must accept binary data of unknown length. Inevitably, in situations where the input data size is unknown ahead of time, the CSTRING created is too big and wastes memory, or too small and either fails or GPFs due to memory overwrite. Since this is an object that once out of scope self-destructs, there is also less risk of leaving unused, but still-allocated memory space attached to your application, thus preventing "memory leak".

HNDBuffer also includes a "BufferHash" function which can detect differences between the data held in one buffer and another. A Hash value is kept of any given buffer as it has data inserted or removed from it. At any time, the hash value of one buffer can be compared with that of another buffer to determine if they are the same or different. CHT's server classes use HNDBuffer extensively to create memory storage for large, but infinitely size-variable, data coming and going over the network or the internet.

HNDBuffer is used via composition in the following CHT header modules: hndbinio.inc, hndbrvsv.inc, hnddnt.inc, hnddisk.inc, hndencrp.inc, hndhttp.inc, hndimail.inc, hndjsdap.inc, hndpop.inc, hndradap.inc, hndwprlib.inc, hndscdap.inc, hndsmtp.inc, hndstrlb.inc, hndutil.inc, hndvwdap.inc, hndwhttp.inc and hndwmp.inc.

[Back To Previous](#)

[MS Office Interface](#)

Class: **HNDActiveDocument** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: MS Office Interface
Demos: [HNDWORDDEMO.APP](#) [HNDWORDINTERFACEDEMO.APP](#)
DLL: Microsoft Office
TLB: None
Header: HNDMSWRD.INC
Code: HNDMSWRD.CLW
Template: [EmbedMSWordFunctions](#) [HandyWordInterfaceControls](#)
Description:

*** REVISED MARCH 2016 ***

The HNDActiveDocument class is one component of five, providing an OLE interface connection to Microsoft Office documents. Other MS Office objects in the set are: HNDMSWord, HNDApplication, HNDDocuments, HNDActiveDocument, HNDRefWindow.

This set of five Clarion objects allows Clarion applications utilizing them to connect to MS Word in order to send various formatting insertion and deletion commands (spell checking for example) via the same document interface provided by MS Word itself via its "Visual Basic For Applications", command language. For example applications, see HNDWORDDEMO.APP and HNDWORDINTERFACEDEMO.APP.

[Back To Previous](#)

Class: [HNDApplication](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDDocuments](#)
Type: MS Office Interface
Demos: [HNDWORDDEMO.APP](#) [HNDWORDINTERFACEDEMO.APP](#)
DLL: Microsoft Office
TLB: None
Header: HNDMSWRD.INC
Code: HNDMSWRD.CLW
Template: [EmbedMSWordFunctions](#) [HandyWordInterfaceControls](#)
Description:

*** REVISED MARCH 2016 ***

The HNDApplication class is one component of five, providing an OLE interface connection to Microsoft Office documents. Other MS Office objects in the set are: HNDMSWord, HNDApplication, HNDDocuments, HNDActiveDocument, HNDRefWindow.

This set of five Clarion objects allows Clarion applications utilizing them to connect to MS Word in order to send various formatting insertion and deletion commands (spell checking for example) via the same document interface provided by MS Word itself via its "Visual Basic For Applications", command language. For example applications, see HNDWORDDEMO.APP and HNDWORDINTERFACEDEMO.APP.

[Back To Previous](#)

Class: [HNDDocuments](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDActiveDocument](#)
Type: MS Office Interface
Demos: [HNDWORDDEMO.APP](#) [HNDWORDINTERFACEDEMO.APP](#)
DLL: Microsoft Office
TLB: None
Header: HNDMSWRD.INC
Code: HNDMSWRD.CLW
Template: [EmbedMSWordFunctions](#) [HandyWordInterfaceControls](#)
Description:

*** REVISED MARCH 2016 ***

The HNDDocuments class is one component of five, providing an OLE interface connection to Microsoft Office documents. Other MS Office objects in the set are: HNDMSWord, HNDApplication, HNDDocuments, HNDActiveDocument, HNDRefWindow.

This set of five Clarion objects allows Clarion applications utilizing them to connect to MS Word in order to send various formatting insertion and deletion commands (spell checking for example) via the same document interface provided by MS Word itself via its "Visual Basic For Applications", command language. For example applications, see HNDWORDDEMO.APP and HNDWORDINTERFACEDEMO.APP.

[Back To Previous](#)

Class: [HNDMSWord](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDApplication](#)

Type: MS Office Interface
Demos: [HNDWORDDEMO.APP](#) [HNDWORDINTERFACEDEMO.APP](#)
DLL: Microsoft Office
TLB: None
Header: HNDMSWRD.INC
Code: HNDMSWRD.CLW
Template: [EmbedMSWordFunctions](#) [HandyWordInterfaceControls](#)
Description:

*** REVISED MARCH 2016 ***

The HNDMSWord class is one component of five, providing an OLE interface connection to Microsoft Office documents. Other MS Office objects in the set are: HNDMSWord, HNDApplication, HNDDocuments, HNDActiveDocument, HNDRefWindow.

This set of five Clarion objects allows Clarion applications utilizing them to connect to MS Word in order to send various formatting insertion and deletion commands (spell checking for example) via the same document interface provided by MS Word itself via its "Visual Basic For Applications", command language. For example applications, see HNDWORDDEMO.APP and HNDWORDINTERFACEDEMO.APP.

[Back To Previous](#)

Class: [HNDOfficeClass](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDUtil](#) [HNDStrLib](#) [HNDComLibrary](#)
Type: MS Office Interface
Demos: [HNDWORDDEM.APP](#) [HNDWORDINTERFACEDEMO.APP](#)
DLL: Microsoft Office
TLB: HNDOUTTL.TLB
Header: HNDOFFICE.INC
Code: HNDOFFICE.CLW
Template: [EmbedHNDWordClassFunctions](#) [HandyWordInterfaceControls](#)
Description:

This class provides a direct, OLE interface into the Microsoft Office products suite. It is the parent, generalized MS Office interface for more MS-product-specific classes such as HNDOfficeWord and HNDOfficeOutlook.

[Back To Previous](#)

Class: [HNDRefWindow](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDMSWord](#)
Type: MS Office Interface
Demos: [HNDWORDDEMO.APP](#) [HNDWORDINTERFACEDEMO.APP](#)
DLL: Microsoft Office
TLB: None
Header: HNDMSWRD.INC
Code: HNDMSWRD.CLW
Template: [EmbedMSWordFunctions](#) [HandyWordInterfaceControls](#)
Description:

*** REVISED MARCH 2016 ***

The HNDRefWindow class is one component of five, providing an OLE interface connection to Microsoft Office documents. Other MS Office objects in the set are: HNDMSWord, HNDApplication, HNDDocuments, HNDActiveDocument, HNDRefWindow.

This set of five Clarion objects allows Clarion applications utilizing them to connect to MS Word in order to send various formatting insertion and deletion commands (spell checking for example) via the same document interface provided by MS Word itself via its "Visual Basic For Applications", command language. For example applications, see HNDWORDDEMO.APP and HNDWORDINTERFACEDEMO.APP.

[Back To Previous](#)

Class: [HNDOfficeWord](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#) [HNDOfficeClass](#)
Compositions: None
Type: MS Office Interface (Word)
Demos: [HNDWORDDEM.APP](#) [HNDWORDINTERFACEDEMO.APP](#)
DLL: Microsoft Office
TLB: HNDWORD.TLB,HNDOUTTL.TLB
Header: HNDWORD.INC
Code: HNDWORD.CLW
Template: [EmbedHNDWordClassFunctions](#) [HandyWordInterfaceControls](#)
Description:

This class provides a direct, OLE interface to Microsoft Office, particularly MS WORD. Functions included are: List recent documents, Close current document, Get document name, Close all documents, Minimize MS Word, Restore MS Word, Maximize MS Word, Spell check document, Grammar check document.

[Back To Previous](#)

Querying, Locating And Filtering

Class: [HNDParse](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: Querying, Locating And Filtering
Demos: [HND2.APP](#) [HND2.APP](#) [HNDACCES.APP](#) [HNDSQL1.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDPARSE.INC
Code: HNDPARSE.CLW
Template: [None](#)
Description:

This class is the Query Interpreter used throughout CHT to build real-language queries into everything from SQL data view searches to file compression utilities and email filtering. CHT Query Language is based on the principle that queries are easier to construct when the query language being used resembles the user's own native tongue.

At least half of CHT demo applications include CHT Query Language in some way or another to conduct a search or to construct a filter so finding an example of its use is straightforward enough. When querying data tables CHT query language is cross-interpreted into the brand of SQL used by your data table, and into standard Clarion when your data table is an ISAM type like TopSpeed.

Check out the PDF document called <http://www.cwhandy.com/june2006.pdf> for a comprehensive essay on the use of CHT Query Language to provide searching and filtering in your applications.

[Back To Previous](#)

Record Filtering Classes

Class: [HNDBrwFilter](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDParse](#) [HNDError](#)
Compositions: [HNDLocator](#) [HNDUtil](#)
Type: Record Filtering Classes
Demos: [HND2.APP](#) [HND3.APP](#) [HNDACCES.APP](#) [HNDSQL2.APP](#) [HND SCHOOL.APP](#) [More](#)
DLL: None
TLB: None
Header: HND BRWFL.INC
Code: HND BRWFL.CLW
Template: [QueryParsingFilterControl](#) [PopFavoriteQueries](#) [EXPB](#) [ExplorerBrowse](#) [More](#)
Description:

*** REVISED MARCH 2016 ***

This class (HNDBrwFilter) is used by all CHT browse extensions to provide filtering capability using CHT Query Language derived from the CHT HNDParse class. The advanced features of this class come built-in to the CHT browse-extension templates ExplorerBrowse, HandyMarkerBrowse, LocatorOverRideControl, and others.

Thus, these features may be used via embedding inside your browse procedure without applying any other CHT query control or search control templates. However, a wide variety of such query and search control templates is available to write search and query code for you, without embedding, using the functionalities provided by HNDBrwFilter and it's parent HNDParse.

Example query templates are:
QueryParsingFilterControl, PopFavoriteQueries_HMB,
HandyMarkerBrowseLocatorControlEx and nearly a dozen more.

[Back To Previous](#)

SMS Messages

Class: [HNDSMS](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDConfig](#) [HNDError](#)
Compositions: [HNDBuffer](#) [HNDDisk](#) [HNDEncrypt](#)
Type: SMS Messages
Demos: [HNDSMSMESSAGE.APP](#)
DLL: CLARUNEXT.DLL
TLB: None
Header: HNDSMS.INC
Code: HNDSMS.CLW
Template: [EmbedSMSFunctions](#)
Description:

*** REVISED AUGUST 2015 ***

This class handles SMS message sending. It leverages Clarion DLL, CLARUNEXT.DLL which exists in both the run-time set provided by Clarion Professional and Clarion Enterprise. This new CHT Class (as of July 2015) provides an SMS API with which to send SMS message either via an SMS Gateway Server or an SMTP Email Server. SMS messages sent via an SMTP Email Server will be transferred to the target packaged as an email. In either case the SMS address is the recipient's cell number plus his/her SMS GateWay Domain.

Format the receiving address as #####@sms.gatewaydomain.com.
For an example application look at: HNDSMSMESSAGES.APP.

[Back To Previous](#)

Sound And Video

Class: [HNDAvi](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDSlider](#) [HNDAanimate](#)
Type: Sound And Video
Demos: [HNDQMCI.APP](#)
DLL: None
TLB: None
Header: HNDAVI.INC
Code: HNDAVI.CLW
Template: [EmbedAVFunctions](#)
Description:

*** REVISED MARCH 2016 ***

The CHT HNDAvi class implements MS Windows MCI, audio and video playback giving Clarion developers relatively painless programmatic access to these built-into-Windows sound and video functionalities.

Since the capabilities of this class have evolved tremendously over the years, the class

name has become a misnomer. AVI is only one file format of many, that can be played. Demo application HNDQMC1.APP illustrates playing file formats MP3, WAV, AVI, MIDI, MPG and WMV. A template called EmbedAVIFunctions may be used to attach the HNDAMI class to your procedure.

[Back To Previous](#)

[Split Bar Functionality](#)

Class: **HNDSplitter** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: Split Bar Functionality
Demos: [HNDSPPLITTEST.APP](#) [HNDCLXHT.APP](#) [HNDTPXHT.APP](#)
DLL: None
TLB: None
Header: HNDSPPLITTER.INC
Code: HNDSPPLITTER.CLW
Template: [EmbedHNDSplitter](#) [HandySplitBarVertical](#) [HandySplitBarHorizontal](#)
Description:

*** REVISED NOVEMBER 2014 ***

This class handles a split-bar control, either vertical or horizontal, which divides up the window and moves or resizes controls up/down or left/right, respectively, when the split-bar control is dragged.

A very good example of how this control can be used to maximize window real-estate without using tabs is HNDCLXT.APP.

The parent template to attach this functionality to a window is *EmbedHNDSplitter*. Once that template is populated, the control templates, *HandySplitBarVertical* or *HandySplitBarHorizontal* can be dropped on the window.

[Back To Previous](#)

[Window Callback Functions](#)

Class: **HNDCallBack** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDRGBColor](#)
Type: Window Callback Functions
Demos: [HND8.APP](#) [Most CHT Demos](#)
DLL: None
TLB: None
Header: HNDALBK.INC
Code: HNDALBK.CLW
Template: [EmbedWindowFunctions](#) [EmbedHNDCallBackFunctions](#)
Description:

*** REVISED JUNE 2016 ***

This is a low-level, window call-back functions class derived by a higher level class called *HNDWindow*. As such, *HNDCallBack* is never attached to your window procedure all by itself. It comes in as a derived component of the CHT *HNDWindow* Class, which is populated on Window procedures by *EmbedWindowFunctions*.

One example, of what the code in this class is capable of doing, is finding other windows, even those of other running programs, and sending messages to them via the function OBJ.SendAppEvent().

Since most CHT Demos incorporate template *EmbedWindowFunctions*, on their Window procedures, based in the ABC Window template, almost any CHT demo will illustrate some aspect of the *HNDCallBack* class.

[Back To Previous](#)

Class: [HNDDisk](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDUtil](#) [HNDError](#)
Compositions: [HNDDosFile](#) [HNDBuffer](#) [HNDEncrypt](#) [HNDZStore](#) [HNDAanimate](#)
Type: Windows API
Demos: [HND5.APP](#) [HNSETUP.APP](#) [HNSCHDM.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDDISK.INC
Code: HNDDISK.CLW
Template: [EmbedDiskFunctions](#) [More](#)
Description:

*** REVISED JUNE 2017 ***

This class contains an expansive collection of utility functions that provide services as widely varied as file compression, file encryption, disk file searching with querying, file search and replace, shell operations and more. Because HNDDisk derives HNDUtil, it inherits all of its abilities as well. HNDDisk is used either directly or indirectly by numerous other CHT classes, drawing on its lower level utility functionality. For direct use in your procedures, HNDDisk can be attached to any procedure using the CHT template called EmbedDiskFunctions. HNDDisk is also derived directly by the HNDDiskEx class.

This higher level class derives and utilizes HNDDisk and HNDUtil methods and properties as if they were its own. The method names in HNDDisk are reasonably self explanatory, for example: ShrinkFile(), GrowFile(), RunShell() FileCopy(), DirMove() and so on.

Separate details are provided for each method in the HNDDisk class. By scanning the method (procedure) names to determine likely candidates to provide the functionality you're looking for, and then reading the method's description, which often includes the names of demonstration applications, you should be able to gain a full understanding of how to use any method in this class. Note that some methods, meant for internal use by the class itself, will be so marked and may be more scantily documented.

[Back To Previous](#)

Class: [HNDDiskEx](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDDisk](#) [HNDUtil](#) [HNDError](#)
Compositions: None
Type: Windows API
Demos: [HNSETUP.APP](#) [HNDCPYDM.APP](#) [HNDZTEST.APP](#) [HNDCPYDM.APP](#) [More](#)
DLL: None
TLB: None
Header: HNDDSKEX.INC
Code: HNDDSKEX.CLW
Template: [EmbedDiskFunctions](#)
Description:

*** REVISED NOVEMBER 2014 ***

This class contains a group of utility functions that provide bulk file processing services including: compression, encryption, containerization, copying, moving and erasing. For example: BulkContainerizeFiles(), BulkFileMove().

Use HNDDiskEx with installer applications, or server/client applications that need to package files in a secure, compact manner for web transfer.

[Back To Previous](#)

Class: [HNDUtil](#) [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: [HNDBuffer](#)
Type: Windows API
Demos: [Most](#)
DLL: None
TLB: None

Header: HNDUTIL.INC
Code: HNDUTIL.CLW
Template: [None](#)
Description:

*** REVISED MARCH 2017 ***

This class contains a broad collection of utility functions that provide services as widely varied as string manipulation, operating system detection, shell folder detection, path manipulation, file statistics, data hashing and more. This class can be attached to your application procedures directly using the CHT template called EmbedUtilityFunctions. HNDUtil is also derived directly by the HNDDisk class and indirectly by the HNDDiskEx class.

These higher level classes derive and utilize HNDUtil methods and properties as if they were their own. The method names in HNDUtil are reasonably self explanatory, for example: GetComputerName(), LongToHex(), RemoveBetween(), NoSpace() and so on.
--RECENT CHANGES AND ADDITIONS--

MARCH 2017 PrintShell now presents a preview of the to-be-printed file from which the print menu may be exercised.

[Back To Previous](#)

[XXX Docs Yet ToDo XXX](#)

Class: **HNDDockingWindow** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: [HNDSRCT.APP](#)
DLL: None
TLB: None
Header: HNDDOCWN.INC
Code: HNDDOCWN.CLW
Template: [EmbedDockingWindow](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDDosFile** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDDOSF.INC
Code: HNDDOSF.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDDynaEntry** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None

Header: HNDYEN.INC
Code: HNDYEN.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDEventLoggerClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDEVEER.INC
Code: HNDEVEER.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDExcelClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDEXCEL.INC
Code: HNDEXCEL.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDExport** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDEXPOR.INC
Code: HNDEXPOR.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDFile** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None

TLB: None
Header: HNDFILE.INC
Code: HNDFILE.CLW
Template: [None](#)
Description:

Class description is in preparation.
[Back To Previous](#)

Class: **HNDFlashClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDFLASH.INC
Code: HNDFLASH.CLW
Template: [None](#)
Description:

Class description is in preparation.
[Back To Previous](#)

Class: **HNDGreenbar** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDGNBAR.INC
Code: HNDGNBAR.CLW
Template: [None](#)
Description:

Class description is in preparation.
[Back To Previous](#)

Class: **HNDHexToDecimal** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDHEXDC.INC
Code: HNDHEXDC.CLW
Template: [None](#)
Description:

Class description is in preparation.
[Back To Previous](#)

Class: **HNDIDial** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos:

DLL: None
TLB: None
Header: HNDIDIAL.INC
Code: HNDIDIAL.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDIPAddress** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDIPADR.INC
Code: HNDIPADR.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDIScavage** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDISCAV.INC
Code: HNDISCAV.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDInetFtp** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDINETF.INC
Code: HNDINETF.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDInternetExplorer** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None

Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDSHDW.INC
Code: HNDSHDW.CLW
Template: [None](#)
Description: Class description is in preparation.
[Back To Previous](#)

Class: **HNDJDOAdapter** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDJSDAP.INC
Code: HNDJSDAP.CLW
Template: [None](#)
Description: Class description is in preparation.
[Back To Previous](#)

Class: **HNDLocator** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDLOCTR.INC
Code: HNDLOCTR.CLW
Template: [None](#)
Description: Class description is in preparation.
[Back To Previous](#)

Class: **HNDMailDefaults** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDMLDEF.INC
Code: HNDMLDEF.CLW
Template: [None](#)
Description: Class description is in preparation.
[Back To Previous](#)

Class: **HNDMailSlotClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)

Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDMAIOT.INC
Code: HNDMAIOT.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDMenu** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDMENU.INC
Code: HNDMENU.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDMsCDO** [Click For Methods Listing](#)
Status: Current
Parent: [HNDConfig](#)
Compositions: [HNDError](#) [HNDDisk](#) [HNDDosFile](#) [HNDDStrLib](#) [HNDComLibrary](#) [HNDDates](#) [HNDEncrypt](#) [HNDRegistry](#)
Type: XXX Docs Yet ToDo XXX
Demos: [HNDCDODM.APP](#) [HNDMAILCDO.APP](#) [HNDZIPMAILCDO.APP](#)
DLL: CDO.DLL, MAPI32.DLL, CDOHTML.DLL
TLB: HNDMSADO.TLB,HNDMSADO.TLB
Header: HNDMSCDO.INC
Code: HNDMSCDO.CLW
Template: [EmbedCDOMailFunctions](#)
Description:

*** REVISED DECEMBER 2015 ***

HNDMsCDO provides SMTP mail from MS Windows CDO (Collaboration Data Objects) mail capability.

Collaboration Data Objects are a high-level set of COM objects that allow you to easily access the e-mail system embedded in the Microsoft Windows product line. CDO objects are generally used by client applications. For the most part, CDO objects are used by clients wanting to access the e-mail system. Service providers must be implemented using the native MAPI library.

Not all Microsoft Windows product line products install CDO capability by default. When you install Exchange 5.5 or later, you must take special steps in order to install these libraries. Choose Custom Setup. From this setup, select IIS/Active Server/Active Server Pages Option. This setup process causes the earlier DLLs to be installed on your system.

[Back To Previous](#)

Class: **HNDMsWebCapClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None

DLL: None
TLB: None
Header: HNDMSWAP.INC
Code: HNDMSWAP.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDMscalClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDMSCAL.INC
Code: HNDMSCAL.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDNetDriveDataClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDNETDRIVEDATA.INC
Code: HNDNETDRIVEDATA.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDNetFTPClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDNETFTP.INC
Code: HNDNETFTP.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDNetWlanAPIClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX

Demos: None
DLL: None
TLB: None
Header: HNDNETWLANAPI.INC
Code: HNDNETWLANAPI.CLW
Template: [None](#)
Description:

Class description is in preparation.
[Back To Previous](#)

Class: **HNDOSControl** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDOSCTL.INC
Code: HNDOSCTL.CLW
Template: [None](#)
Description:

Class description is in preparation.
[Back To Previous](#)

Class: **HNDOfficeOutlook** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDOUTLOOKMS.INC
Code: HNDOUTLOOKMS.CLW
Template: [None](#)
Description:

Class description is in preparation.
[Back To Previous](#)

Class: **HNDProgress** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDPROGR.INC
Code: HNDPROGR.CLW
Template: [None](#)
Description:

Class description is in preparation.
[Back To Previous](#)

Class: **HNDQueryBuilder** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)

Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDQBINF.INC
Code: HNDQBINF.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDRGBColor** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDRGBCO.INC
Code: HNDRGBCO.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDRegistry** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDREG.INC
Code: HNDREG.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDReportEngine** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDRWPRLIB.INC
Code: HNDRWPRLIB.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDRuler** [Click For Methods Listing](#)
Status: Current

Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDRULER.INC
Code: HNDRULER.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDScriptHostClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDWSHOM.INC
Code: HNDWSHOM.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDScripiter** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDSCDAP.INC
Code: HNDSCDAP.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDService** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDSERVC.INC
Code: HNDSERVC.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class:

HNDSShell [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNSHELL.INC
Code: HNSHELL.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

HNDSShellLinkClass [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNSHLNK.INC
Code: HNSHLNK.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

HNDSizes [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNSIZES.INC
Code: HNSIZES.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

HNDSlider [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNSLIDE.INC
Code: HNSLIDE.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDSpell** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDSPELL.INC
Code: HNDSPELL.CLW
Template: [None](#)

Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDStepClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDSTEPS.INC
Code: HNDSTEPS.CLW
Template: [None](#)

Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDStrLib** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDSTRLB.INC
Code: HNDSTRLB.CLW
Template: [None](#)

Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HNDString** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDSTR.INC
Code: HNDSTR.CLW
Template: [None](#)

Description:
Class description is in preparation.

[Back To Previous](#)

Class: **HNDStringClass** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDSTRCLASS.INC
Code: HNDSTRCLASS.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDText** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDHTML.INC
Code: HNDHTML.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDTiles** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDTILES.INC
Code: HNDTILES.CLW
Template: [None](#)
Description:

Class description is in preparation. This library is under construction.

[Back To Previous](#)

Class: **HNDViewAdapter** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDWVDAP.INC
Code: HNDWVDAP.CLW
Template: [None](#)
Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDViewManager** [Click For Methods Listing](#)

Status: Current

Parent: [HNDError](#)

Compositions: None

Type: XXX Docs Yet ToDo XXX

Demos: None

DLL: None

TLB: None

Header: HNDWWMGR.INC

Code: HNDWWMGR.CLW

Template: [None](#)

Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDWinHTTP** [Click For Methods Listing](#)

Status: Current

Parent: [HNDError](#)

Compositions: None

Type: XXX Docs Yet ToDo XXX

Demos: None

DLL: None

TLB: None

Header: HNDWHTTP.INC

Code: HNDWHTTP.CLW

Template: [None](#)

Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDWindow** [Click For Methods Listing](#)

Status: Current

Parent: [HNDError](#)

Compositions: None

Type: XXX Docs Yet ToDo XXX

Demos: None

DLL: None

TLB: None

Header: HNDWINDO.INC

Code: HNDWINDO.CLW

Template: [None](#)

Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HNDWindowsMediaPlayer** [Click For Methods Listing](#)

Status: Current

Parent: [HNDError](#)

Compositions: None

Type: XXX Docs Yet ToDo XXX

Demos: None

DLL: None

TLB: None

Header: HNDWMP.INC

Code: HNDWMP.CLW

Template: [None](#)

Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HndBody** [Click For Methods Listing](#)

Status: Current

Parent: [HndTable](#)

Compositions: None

Type: XXX Docs Yet ToDo XXX

Demos: None

DLL: None

TLB: None

Header: HNDHTML.INC

Code: HNDHTML.CLW

Template: [None](#)

Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HndEntry** [Click For Methods Listing](#)

Status: Current

Parent: [HNDInput](#)

Compositions: None

Type: XXX Docs Yet ToDo XXX

Demos: None

DLL: None

TLB: None

Header: HNDHTML.INC

Code: HNDHTML.CLW

Template: [None](#)

Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HndFooter** [Click For Methods Listing](#)

Status: Current

Parent: [HndTable](#)

Compositions: None

Type: XXX Docs Yet ToDo XXX

Demos: None

DLL: None

TLB: None

Header: HNDHTML.INC

Code: HNDHTML.CLW

Template: [None](#)

Description:

Class description is in preparation.

[Back To Previous](#)

Class: **HndForm** [Click For Methods Listing](#)

Status: Current

Parent: None

Compositions: None

Type: XXX Docs Yet ToDo XXX

Demos: None

DLL: None

TLB: None

Header: HNDHTML.INC

Code: HNDHTML.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HndGlobal** [Click For Methods Listing](#)
Status: Current
Parent: None
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDHTML.INC
Code: HNDHTML.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HndHeader** [Click For Methods Listing](#)
Status: Current
Parent: [HndTable](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDHTML.INC
Code: HNDHTML.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HndHtml** [Click For Methods Listing](#)
Status: Current
Parent: [HndDosFile](#)
Compositions: [Util](#) [Page](#) [PageQ](#)
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDHTML.INC
Code: HNDHTML.CLW
Template: [None](#)
Description:
Class description is in preparation.
[Back To Previous](#)

Class: **HndInput** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB:

Header: None
Code: HNDHTML.INC
Template: [None](#)
Description: Class description is in preparation.
[Back To Previous](#)

Class: **HndPage** [Click For Methods Listing](#)
Status: Current
Parent: [HNDError](#)
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDHTML.INC
Code: HNDHTML.CLW
Template: [None](#)
Description: Class description is in preparation.
[Back To Previous](#)

Class: **HndTable** [Click For Methods Listing](#)
Status: Current
Parent: None
Compositions: None
Type: XXX Docs Yet ToDo XXX
Demos: None
DLL: None
TLB: None
Header: HNDHTML.INC
Code: HNDHTML.CLW
Template: [None](#)
Description: Class description is in preparation.
[Back To Previous](#)
