

CHT Applications

[HOME](#) [CLASSES](#) [TPLS](#) [APPS](#)

TEMPLATES, CLASSES, UTILITIES, APPS, PROJECTS

CHT APPLICATIONS FULL SET (BUILD: 23C.03.00) AUGUST 24, 2019

CATEGORY: BATCH BOT

*** HNDENCRYPT.APP ***

NAME: HNDENCRYPT.APP
TITLE: BATCH-BOT ENCRYPTION [V10.19.08]
DATE: 8/24/2019
CATEGORY: BATCH BOT
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDENCRYPT.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

HNDENCRYPT.EXE command-line parameters are as follows:

```
** HELP
--help          [OPT] Pops up a splash help screen.
--video         [OPT] Display a video or a document from the web which
                further explains use of the application.
--update       [OPT] Self-updates to the latest available version from CHT
                servers. Utilizes CHTSNAPGET.EXE if / when invoked.

** ENCRYPT/DECRYPT
--input         [REQ] Path and name of input file.
                If the path is omitted, the .EXE path is assumed.
--output       [REQ] Output path only - local to this exe if omitted.
--mode         [OPT] Encode or decode - defaults to encode if omitted.
--key          [OPT] Encrypting password - read from registry if omitted.
```

The output file name is formatted from the input file with a .hzy extension. For example: test.pdf becomes testpdf.hzy.

The encrypted file is both compressed and encrypted by the operation of this CHT Batch Bot application. This operation is performed in a single pass whether encoding or decoding.

*** HNDFTPNET.APP ***

NAME: HNDFTPNET.APP
TITLE: BATCH-BOT FTP TOOL [V10.19.08]
DATE: 8/24/2019

CATEGORY: BATCH BOT

CLASSES: [List App Classes](#)

TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDFTPNET.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

HNDFTPNET.EXE is dependent on a CHT DLL called HNDFTPNET.DLL being present and visible on the system path, for its FTP capabilities.

The application communicates to this DLL via COM, requiring the HNDFTPNET.DLL to be registered on the host system. This DLL will auto-register on the host system when first called and found not to be registered.

CONFIGURING:

--config0
[CFG] Set up FTP configuration information #0

--config1
[CFG] Set up FTP configuration information #1

--config2
[CFG] Set up FTP configuration information #2

--set0
[CFG] Select and use configuration #0

--set1
[CFG] Select and use configuration #1

--set2
[CFG] Select and use configuration #2

COMMANDS:

--send
[OPT] Provides file name of the local file to be uploaded.
This file need not exist if used with the --mask tag to indicate that a pre-zip is performed using a mask.

--remote
[REQ] Names the remote directory with which to interact.
This is a required parameter which is formed as follows:
/ftprootdir/subdir1/subdir2/

--local
[OPT] Names a local directory when the --mask tag is used to describe one or more files for pre-zipping.
Defaults to the application directory if omitted.

--pwd
[OPT] Password used when pre-zipping with --mask.

--mask
[OPT] DOS-style file mask (eg: *.app) describes files to be zipped. Requires CHTSNAPZIP.EXE to be available.
--send parameter must be .zip file name

--fetch
[OPT] File name of a remote file to be downloaded.
Use instead of --send or --erase when downloading.

--erase
[OPT] Provides file to be erased from a remote directory.
Use instead of --fetch and --send when deleting.

--list
[OPT] Used by itself, this provides a file listing of the remote directory stipulated after the tag.
E.G. --list /web/subdir/

LOGGING:

All successful transactions or errors are logged into "hndftpnet.log" located in the same directory as the hndftpnet.exe.

HELP:

--help
[OPT] Used by itself, this pops up a splash help screen
Creates hndftpnethelp.txt which contains the contents of this splash screen.

NAME: HNDMAILCDO.APP
TITLE: BATCH-BOT CDO MAIL [V10.19.08]
DATE: 8/24/2019
CATEGORY: BATCH BOT
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The email template used to build this application is EMBEDCDOMAILFUNCTIONS. The CHT CDO Class is called HNDMSCDO located in files HNDMSCDO.INC and HNDMSCDO.CLW.

HNDMAILCDO.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

HNDMAILCDO.EXE runs from the DOS prompt or can be called from another application using Clarion RUN. It accepts several command line parameters that describe the email to be sent. These parameters are as follows:

SENDING:

```
--to          [REQ] mymail@address.com (multiples add semi-colon between)
--cc          [OPT] mycc@address.com (multiples add semi-colon between)
--bcc         [OPT] mybcc@address.com (multiples add semi-colon between)
--subj        [REQ] This is the subject line.
--body        [OPT] This is a short email body.
              Optional when the --body parameter is used.
--file        [OPT] emailbody.txt/.html (Alternate to --body for body text)
--attach      [OPT] somezip.zip (Name of attachment requires full path)
--mail        [OPT] mailfile.txt (INI file with tags To=, Subj=, Body=, Attach= )
```

CONFIGURING:

```
--config0    [CFG] Set up mail configuration information #0
--config1    [CFG] Set up mail configuration information #1
--config2    [CFG] Set up mail configuration information #2
--set0       [CFG] Select and use configuration #0
--set1       [CFG] Select and use configuration #1
--set2       [CFG] Select and use configuration #2
```

LOGGING:

All transactions such as config changes, emails or errors are logged into "hndmailcdo.log" located in the same directory as the hndmailcdo.exe.

HELP:

```
--help Pops up a splash help screen and help file hndmailcdohelp.txt
--silent [OPT] Suppresses popup logs and help during mult-recipient sends
```

NAME: HNDMAILMAPI.APP
TITLE: BATCH-BOT MAPI MAIL [V10.19.08]
DATE: 8/24/2019
CATEGORY: BATCH BOT
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This "CHT Batch Bot" application provides MAPI send capability provided by a CHT DLL called HNDNETMAPI.DLL written with C#.

The CHT template illustrated in this application is called SOURCE_EMBED_MAPI_FUNCTIONS. The class underlying this template is called HNDIMAIL located in files HNDIMAIL.INC and HNDIMAIL.CLW.

HNDMAILMAPI.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

When you have a properly configured MAPI email client like Outlook or Windows Live Mail configured, this application is able to connect with it to send email. This application is stand-alone and requires no other CHT DLLs.

HNDMAILMAPI.EXE runs from the DOS prompt or can be called from another application using Clarion RUN, from a batch file or even from a .VBS script. It is capable of sending standard email of the type(s) configured in your default email client.

It accepts several command line parameters that describe the email to be sent. These parameters are as follows:

SENDING:

--to [REQ] mail@address.com (or multiples + semi-colon separators)
--subj [REQ] This is the subject line.
--body [OPT] This is a short email body.
Optional when the --file parameter is used.
--file [OPT] emailbody.txt (Alternate to --body for lengthy body text)
Optional when the --body parameter is used.
--attach [OPT] somezip.zip (Name of attachment requires full path)
--mail [OPT] mailfile.txt - INI file with tags To=, Subj=, Body=, Attach=

LOGGING:

All transactions such as config changes, emails or errors are logged into "hndmailmapilog" located in the same directory as the hndmailmapie.exe.

HELP:

--help Pops up a help text file hndmailmapihelp.txt

*** HNDMAILNET.APP ***

NAME: HNDMAILNET.APP
TITLE: BATCH-BOT NET MAIL [V10.19.08]
DATE: 8/24/2019
CATEGORY: BATCH BOT
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDMAILNET.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

The CHT email template used in this application is EMBEDNETSMTPCLASSES. The CHT class underlying this template is HNDNETSMTPCLASS located in files HNDNETSMTP.INC and HNDNETSMTP.CLW.

HNDMAILNET.EXE runs from the DOS prompt or can be called from another application using Clarion RUN, from a batch file or even from a .VBS script. It is capable of sending standard SMTP email as well as SSL SMTP email as found with GMAIL, HOTMAIL and other similar services.

It accepts several command line parameters that describe the email to be sent. These parameters are as follows:

**** SENDING:**

--to [REQ] mail@address.com
--cc [OPT] mail@address.com
--bcc

[OPT] mail@address.com
--subj [REQ] This is the subject line.
--body [OPT] This is a short email body. Optional when --file is used.
For lengthy body text use --body instead.
--file [OPT] emailbody.txt (Alternate to --body for lengthy body text)
Optional when --body is used.
--attach [OPT] somezip.zip (Name of attachment requires full path)
--mail [OPT] INI-style file with tags To=, Subj=, Body=, Attach=
--silent [OPT] Suppress popup document between multiple emails

CONFIGURING:

--config0 [CFG] Set up mail configuration information #0
--config1 [CFG] Set up mail configuration information #1
--config2 [CFG] Set up mail configuration information #2
--set0 [CFG] Select and use configuration #0
--set1 [CFG] Select and use configuration #1
--set2 [CFG] Select and use configuration #2

LOGGING:

All transactions such as config changes, emails or errors are logged into "hndmailnet.log" located in the same directory as the hndmailnet.exe.

HELP:

--help [OPT] Pops up a help document
--silent [OPT] Suppress pop up log and help file during multi-recipient sends.

*** HNDMAILSMTP.APP ***

NAME: HNDMAILSMTP.APP
TITLE: BATCH-BOT SMTP MAIL [V10.19.08]
DATE: 8/24/2019
CATEGORY: BATCH BOT
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Further it is dependent on correct configuration of a valid SMTP email setup with UserName, Password, Email Server and Email Return Address. Our template and SMTP classes provide a configuration function with user-controllable registry location settings.

HNDMAILSMTP.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

The email template illustrated is EMBEDSMTPFUNCTIONS. The CHT class underlying this template is HNDSMTP located in files HNDSMTP.INC and HNDSMTP.CLW.

HNDMAILSMTP.EXE runs from the DOS prompt or can be called from another application using Clarion RUN. It accepts several command line parameters that describe the email to be sent. These parameters are as follows:

SENDING:

--to [REQ] mail@address.com;mail@address2.com;
--cc [OPT] mailcc@address.com;mailcc@address2.com;
--bcc [OPT] mailbcc@address.com;mailbcc@address2.com;
--from [OPT] address of sender, must be same domain
--subj [REQ] This is the subject line.
--body

[OPT] This is a short email body.
Optional if --file parameter is used.

--file [OPT] emailbody.txt (Alternate to --body for lengthy body text)
Optional if --body parameter is used.

--attach [OPT] somezip.zip (Name of attachment requires full path)

--mail [OPT] mailfile.txt (INI file with tags To=, Subj=, Body=, Attach=)

CONFIGURING:

--config0 [CFG] Set up mail configuration information #0

--config1 [CFG] Set up mail configuration information #1

--config2 [CFG] Set up mail configuration information #2

--set0 [CFG] Select and use configuration #0

--set1 [CFG] Select and use configuration #1

--set2 [CFG] Select and use configuration #2

LOGGING:

All transactions such as config changes, emails or errors are logged into "hndmailsmtp.log" located in the same directory as the hndmailsmtp.exe.

HELP:

--help [OPT] Pops up a help splash screen and hndmailsmtphelp.txt

--silent [OPT] Suppresses log and help file popups following send operations

*** HNDMAKEHZO.APP ***

NAME: HNDMAKEHZO.APP
TITLE: BATCH-BOT HZO MAKER [V10.19.08]
DATE: 8/24/2019
CATEGORY: BATCH BOT
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDMAKEHZO.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

HNDMAKEHZO.EXE command-line parameters are as follows:

--help [OPT] Pops up this splash help screen

--video [OPT] Displays a help video or help document from the web

--update [OPT] Self-update from the web if changes available
Utilizes CHTSNAPGET.EXE if / when invoked.

** Build A Container

--input [REQ] Path of input directory

--output [REQ] Path and name of output .HZO file

** Extract a Container

--extract [REQ] Name of container to extract

--target [REQ] Name of target extraction directory

** Used with Build or Extract

--pass [OPT] Encrypting password

--query [OPT] CHT Query to describe files

NAME: HNDSEARCH.APP
TITLE: BATCH-BOT SEARCH TEXT [V10.19.08]
DATE: 8/24/2019
CATEGORY: BATCH BOT
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDSEARCH.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

The application accepts several command line parameters that describe search values, replace values and more. These parameters are as follows:

- help
 [OPT] Pops up application splash screen.
 When used, all other parameters are ignored.
- video
 [OPT] Displays help video or help document from the web.
- mask
 [REQ] Required parameter eg: c:\temp*.app
 Describes single or multiple files to be searched.
 Path required, unless file(s) in same directory as this exe.
- search
 [REQ] Required parameter eg: stop(
 This describes what you're searching for.
- replace
 [OPT] Optional parameter if only a search
 This replaces the value under --search
- update
 [OPT] Update HNDSEARCH.EXE from the web.
 When used, all other parameters are ignored.
 Utilizes CHTSNAPGET.EXE if / when invoked.

NAME: HNDSLFCMD.APP
TITLE: BATCH-BOT HTTP TOOL [V10.19.08]
DATE: 8/24/2019
CATEGORY: BATCH BOT
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This "CHT Batch Bot" application is a command-line HTTP tool which can send to, fetch and erase files from any standard CHT HTTP site managed by a Type 4 CHT Server such as HNDSLFSV.APP.

The "CHT Batch Bot" application can be called from one of your Clarion applications, from a Windows batch file or from a Windows .VBS script to provide reliable file upload, download and delete services.

HNDSLFCMD.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

CONFIGURATION COMMANDS:

- config0 to --config9
 [CFG] Set up HTTP configuration information #0 to #9

SET COMMANDS:

"Set" commands may be combined with --send, --fetch or --erase.

- set0 to --set9
 [CFG] Select and use configuration #0 to #9
- set
 [CFG] Manually inserted url and port, ignore config settings.
 E.G. --set http://www.yoursite.org:88/
- userpass
 [CFG] Manually inserted username and password

E.G. --userpass username:userpass

ACTION COMMANDS

--send

[REQ] Provides file name of the local file to be uploaded.
Use with --local to name the local pickup directory.
Use with --remote to name the remote target directory.
Multiple files may be listed, separated by commas.

--fetch

[OPT] File name of a remote file to be downloaded.
Use instead of --send or --erase for downloading files.
Use with --local to name the local target directory.
Use with --remote to name the remote pickup directory.
Multiple files may be listed, separated by commas.

--erase

[OPT] Provides file to be erased from a remote directory.
Use instead of --fetch and --send when deleting.
Use with --remote to name the remote directory.
Multiple files may be listed, separated by commas.

SOURCE/TARGET COMMANDS:

--remote

[OPT] Remote directory with which to interact.
Remote directory below server /root/ directory.
Example usage: subdir1/subdir2/
Defaults to the server /root/ directory if omitted.

--local

[OPT] Names a local directory to indicate --send pickup directory or to indicate --fetch target directory.
Defaults to the application directory if omitted.

HELP COMMANDS:

--help

[OPT] Pops up a splash help screen.
Creates hndslfcmhelp.txt which contains the contents of this splash screen.

LOGGING:

All successful transactions and/or errors are logged into "hndslfcmd.log" located in the same directory as the hndslfcmd.exe.

*** HNDZIPNET.APP ***

NAME: HNDZIPNET.APP
TITLE: BATCH-BOT ZIP FILES [V10.19.08]
DATE: 8/24/2019
CATEGORY: BATCH BOT
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDZIPNET.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

HNDZIPNET.EXE is dependent on one DLL being present and visible on the system path, namely: HNDNETZIPPER.DLL.

The application communicates to this DLL via COM requiring the HNDNETZIPPER.DLL to be registered on the host system. The DLL will auto-register on the host system when first called and found not to be registered. Auto-registration can only be performed from an "admin" account.

** ZIPPING/UNZIPPING/LISTING ***

--zip

[REQ] Provides file name of the target zip file to be created.
Note that the ESC key cancels zip operations in all cases.
Use --path to name the source files location.

--unzip

[REQ] File name of the source zip file to be processed.
Use --path to name the unzip location.

--list [OPT] Lists the entire contents of the zip to a file called "hndzipnet.txt" and displays it in CHT Snap Edit.

--path [OPT] Names the path where source files are located. Application path is used when this tag is omitted.

--subs [OPT] --subs denotes include subdirectories

--win [OPT] --win 1 denotes show progress window

--mask [OPT] Standard dos-style file mask to describe files to be zipped,, for example *.app

--qry [OPT] CHT-Style file query to describe files to be zipped, for example: NAME ENDSWITH APP OR NAME * .DCT

--silent [OPT] --silent requires no parameter use this for "silent" running so that Snap Edit does not pop up with results

HELP:
--help [OPT] Pops up this splash help screen. Creates a help file called "hndzipnet.txt" Displays "hndzipnet.txt" in CHT Snap Editor window.

LOGGING:
All successful transactions or errors are logged into "hndzipnet.txt" located in the same directory as the hndzipnet.exe.

CATEGORY: BROWSES EXPB

*** HND2.APP ***

NAME: HND2.APP
TITLE: BROWSES EXPB (ISAM) [V10.19.08]
DATE: 8/24/2019
CATEGORY: BROWSES EXPB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates primarily, the use of a Clarion Handy Tools Template called ExplorerBrowse (EXPB); hence the name of the application.

EXPB is an extension template that extends a standard ABC browse with some 25+ features including header-click browse column sorting, built-in locators, greenbar support, and much more.

EXPB itself can also be extended by adding a locator template called ExplorerBrowseLocatorControl and a query control called QueryParsingFilterControl (QPFC). Examples of both are provided. Ancillary templates called HandyQueryProcessLimiter and HandyQueryReportLimiter allow you to pass queries directly from the browse to a report or process via a window button template called HandyQueryProcessButton. See the procedure called Demobrowse4 for examples of this powerful feature.

The major CHT classes involved are: HNDBrowse, HNDBrwFilter and HNDParse. Subscribing developers should take some time to study the header files for these three classes to become familiar with available properties and methods in order to gain the maximum amount of utility from them.

*** HNDEXPLORERSQLJOINTEST.APP ***

NAME: HNDEXPLORERSQLJOINTEST.APP
TITLE: BROWSES EXPB (SQL JOINS) [V10.18.12]
DATE: 8/24/2019

CATEGORY: BROWSES EXPB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application is designed to illustrate cascading joins on a CHT Explorerbrowse. It shows how to configure up to 4 browses (1 parent and 3 children) which must each reset when the parent is reset but which then must also reset the each of the child browses below them in a top to bottom cascade.

TOPMOST (PARENT) resets all child browses
LEVEL 1 CHILD resets LEVEL 2 and LEVEL 3 CHILD
LEVEL 2 CHILD Resets LEVEL 3 CHILD

This application may be combined with HNDSQLTRACE to see the SQL being triggered as browses are reset through navigation, query, and order change. CHT Explorerbrowse optimizes and reduces the counter-productive SQL "Chatter" that normally issues from ABC Browse Class during SQL resets.

*** HINDEXPLORERSQLTEST.APP ***

NAME: HINDEXPLORERSQLTEST.APP
TITLE: BROWSES EXPB (SQL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES EXPB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates CHT ExplorerBrowse used in an SQL setting.

It illustrates these concepts:

- *** A CHT ExplorerBrowse Template
- *** A CHT QueryParsingFilterControl Template
- *** A CHT ExplorerBrowseLocatorControlEx Template
- *** A CHT ExplorerBrowseQueryBuilder Template
- *** A CHT ExplorerBrowseOneColumnQueryControl Template

This application can be used in conjunction with HNDSQLTRACE.EXE to explore at the SQL Select level what happens a browse is opened, reordered, queried and so on.

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

*** HNDISAMPOPFAVEXPB.APP ***

NAME: HNDISAMPOPFAVEXPB.APP
TITLE: BROWSES EXPB (ISAM) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES EXPB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates primarily, the use of a Clarion Handy Tools Template called ExplorerBrowse. This is an extension template that extends a standard ABC browse with some 25+ features including header-click browse column sorting, built-in locators, greenbar support, and much more.

ExplorerBrowse itself can also be extended by adding a variety of extension templates. This application adds PopFavoriteQueries_EXPB to provide a way to inject prebuilt queries into the browse to limit the records being displayed.

PopFavoriteQueries_EXPB provides a query control, a clear button, a favorites query selection control, a query builder button, a query save button and a query delete button.

The application also illustrates a procedure built with a CHT Jumpstart that allows you to manage and modify or insert pre-built queries.

*** HNDSQL0.APP ***

NAME: HNDSQL0.APP
TITLE: BROWSES EXPB (SQL) [V10.19.02]
DATE: 8/24/2019
CATEGORY: BROWSES EXPB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates CHT Explorerbrowse used in an SQL setting.

It illustrates these concepts:

*** A CHT ExplorerBrowse implementation returning SQL in the foreground.

*** A hand built SQL query control that accepts an SQL-Language filter.

*** A hand built CHT query control that accepts a CHT Query-Language filter.

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

*** HNDSQL1.APP ***

NAME: HNDSQL1.APP
TITLE: BROWSES EXPB (SQL) [V10.19.02]
DATE: 8/24/2019
CATEGORY: BROWSES EXPB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates CHT Explorerbrowse used in an SQL setting.

It illustrates these concepts:

*** A CHT Query Builder implementation returning SQL in the foreground.

*** A hand built SQL query control using an ABC reset field and a standard entry control

*** Hiding/unhiding columns programmatically

*** Hidden columns are hidden also in CHT Query Builder

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

*** HNDSQL3.APP ***

NAME: HNDSQL3.APP
TITLE: BROWSES EXPB (SQL) [V10.18.12.0225]
DATE: 8/24/2019
CATEGORY: BROWSES EXPB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates CHT ExplorerBrowse used in an SQL setting.

It illustrates these concepts:

*** A CHT ExplorerBrowse Template

*** A CHT ExplorerBrowseQueryBuilder Template

*** A CHT PopFavoriteQueries_EXPB Template

*** A CHT ExplorerBrowseLocatorControlEx Template

*** A CHT ExplorerBrowseOneColumnQueryControl Template

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

*** HNDSQL7.APP ***

NAME: HNDSQL7.APP
TITLE: BROWSES EXPB (SQL) [V10.19.02]
DATE: 8/24/2019
CATEGORY: BROWSES EXPB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates CHT Explorerbrowse used in an SQL setting with locator template: ExplorerBrowseOneColumnQueryControl.

It illustrates these concepts:

- *** CONTAINS query on a single field [configurable]
- *** Display back-end SQL query to the window.
- *** Execute Locator with tab or with GO button.
- *** Insert fixed range query with a check box.

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

*** HNDSQLPOPFAVEXPB.APP ***

NAME: HNDSQLPOPFAVEXPB.APP
TITLE: BROWSES EXPB (SQL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES EXPB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates CHT Explorerbrowse used in an SQL setting in conjunction with query template PopFavoriteQueries_EXPB.

It illustrates these concepts:

- *** A Explorerbrowse implementation running on SQL Data Tables.
- *** CHT Query template PopFavoriteQueries_EXPB.

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

**CATEGORY:
BROWSES HMB**

*** HND3.APP ***

NAME: HND3.APP
TITLE: BROWSES HMB (ISAM) [V10.19.06]
DATE: 8/24/2019
CATEGORY: BROWSES HMB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates two record marking templates that assist in marking records on ABC browses and hand-coded list boxes.

The browse-marking templates involved are: HandyMarkerBrowse (HMB) and ListBoxBrowseExtender (LBX). HandyMarkerBrowse extends a normal ABC browse template while ListBoxBrowseExtender extends a standard list box. They are separate browse-producing templates with different feature sets and some overlapping features.

Other marking-related templates ancillary to these two allow you pass marked records to processes for

export to HTML, TEXT, FILE, EXCEL and print reports limited to only the records marked in your browse. These templates are: HandyMarkerBrowesProcessLimiter, HandyMarkerBrowseReportLimiter, HandyMarkerBrowseProcessButton and HandyMarkerBrowseReportButton.

A ancillary template called HandyMarkerBrowseSetsControl allows you to store and recall sets of marked records by name.

NOTE: One procedure in this application illustrates ListBoxBrowseExtender (LBX) which is an ABC-Browse-Free implementation from CHT. It's the future of Clarion-based browses.

*** HNDISAMPOPFVHMB.APP ***

NAME: HNDISAMPOPFVHMB.APP
TITLE: BROWSES HMB (ISAM) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES HMB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates primarily, the use of a Clarion Handy Tools Template called HandyMarkerBrowse (HMB) This is an extension template that extends a standard ABC browse with some 25+ features including header-click browse column sorting, built-in locators, greenbar support, and much more.

HMB itself can also be extended by adding a variety of extension templates. This application adds PopFavoriteQueries_HMB to provide a way to inject prebuilt queries into the browse to limit the records being displayed.

PopFavoriteQueries_HMB provides a query control, a clear button, a favorites query selection control, a query builder button, a query save button and a query delete button.

*** HNDMARKERSQLJOINTEST.APP ***

NAME: HNDMARKERSQLJOINTEST.APP
TITLE: BROWSES HMB (SQL JOINS) [V10.18.01]
DATE: 8/24/2019
CATEGORY: BROWSES HMB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This app was developed as a test application to study ABC Browse Class reset behaviour by tracing with HandySQLTrace. It was used in revisions to CHT Browse Classes and templates made May 2012.

The purpose of those revisions was to eliminate double and triple SQL requests send from the ABC Browse classes whenever the browse was reset by an order change or a filter change. The ABC BrowseClass function called ResetQueue() is a monument to inefficiency and bad design in the context of an SQL application.

Since CHT's HandyMarkerBrowse is an extension of the ABC browse class, all resets and queue refills are handled by the ABC Browse Class. Our efforts with this CHT browse extension revision was to as much as possible mitigate the inefficient behaviours built into the ABC Browse Class.

This application in conjunction with HNDSQLTRACE.EXE can be used to illustrate just how much CHT's latest Browse Extension templates can improve the speed and efficiency of your SQL browses.

*** HNDMARKERSQLTEST.APP ***

NAME: HNDMARKERSQLTEST.APP
TITLE: BROWSES HMB (SQL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES HMB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application illustrates HandyMarkerBrowse in conjunction with three query control templates: HandyMarkerBrowseLocatorControlEx, HandyMarkerBrowseQueryBuilderControl and QueryParsingFilterControl.

This app was developed as a test application to study ABC Browse Class reset behaviour by tracing with HandySQLTrace. It was used in various revisions to CHT Browse Classes and templates over the years.

The purpose of those revisions was to eliminate double and triple SQL requests send from the ABC Browse classes whenever the browse was reset by an order change or a filter change. The ABC BrowseClass function called ResetQueue() is a monument to inefficiency and bad design in the context of an SQL application.

Since CHT's HandyMarkerBrowse is an extension of the ABC browse class, all resets and queue refills are handled by the ABC Browse Class. Our efforts with CHT browse extension revisions was to as much as possible mitigate the inefficient behaviours built into the ABC Browse Class.

This application in conjunction with HNDSQLTRACE.EXE can be used to illustrate just how much CHT's latest Browse Extension templates can improve the speed and efficiency of your SQL browses.

*** HNDPEOPLEHMB.APP ***

NAME: HNDPEOPLEHMB.APP
TITLE: BROWSES HMB (ISAM) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES HMB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

These are as follows:

Hot Key Alerts:

- ** MouseLeft - Mark a single record, unmarks any others.
- ** ShiftMouseLeft - Mark range between selected and click.
- ** CtrlShiftMouseLeft - Mark from selected to click position.
- ** CtrlMouseLeft - Toggle record marked/unmarked.
- ** CtrlMouseLeft - On header, pops up column hide menu.
- ** 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 up.
- ** DownKey - Switch to single record mode and move down.
- ** CtrlUp - Move cursor up with no marking or unmarking.
- ** CtrlDown - Move cursor down with no marking.
- ** 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.

*** HND SCHOOL.APP ***

NAME: HND SCHOOL.APP
TITLE: BROWSES HMB (SQL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES HMB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This is a CHT demo actively being re-built with CHT templates on ABC demo application SCHOOL.APP,

hence the name HNDSCCHOOL.APP.

Its purpose is to study the use of the SQLite driver with CHT browses, and query templates set to UseSQL=True. We chose to "CHT-ify" this application specifically because it uses several SQLite data tables located in a repository file called SCHOOL.SQLite.

A second purpose is, of course, to provide some visual and functional contrast between a raw, wizard-produced, really UGLY and semi-functional, ABC application and ABC applications extended, enhanced and functionalized by a variety of CHT templates.

To analyse the allocation of templates in this application, open the application in the Clarion IDE and navigate to Global Extensions, select AACHTControlPanel and click on the menu item "Templates By Proc". This sequence of actions triggers an HTML document to display which enumerates and names all templates used in the application by procedure.

We chose to apply, as always, our "minimalist" screen design which avoids unnecessary use of tab controls, and shuns MDI child windows. Control buttons are for the most part kept to the toolbar and when placed on a window are always above the edit fields or list box.

NOTE 1: It appears that the SQLite driver DLL does not provide tracing either by file or by callback, so that CHT SQLTrace will not provide SQL tracing information from this app.

NOTE 2: This application applies, for the most part CHT's browse extension template HandyMarkerBrowse (HMB), hence comes the name of the application. However, one procedure utilizes ListBoxBrowseExtender (LBX) which illustrates that an underlying ABC browse is no longer necessary to build fantastic Clarion browses. LBX browses are the bright future of data browsing with Clarion, ABC browses are the clunky past.

*** HNDSQL2.APP ***

NAME: HNDSQL2.APP
TITLE: BROWSES HMB (SQL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES HMB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates CHT HandyMarkerBrowse used in an SQL setting.

It illustrates these concepts:

- *** A CHT Query Builder implementation returning SQL in the foreground.
- *** An hand-built SQL query control using an ABC Reset Field and a standard entry control.
- *** Hiding/unhiding columns programmatically
- *** Hidden columns are hidden also in CHT Query Builder

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

*** HNDSQL4.APP ***

NAME: HNDSQL4.APP
TITLE: BROWSES HMB (SQL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES HMB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates CHT HandyMarkerBrowse used in an SQL setting.

It illustrates these concepts:

- *** A CHT HandyMarkerBrowse Template
- *** A CHT HandyMarkerBrowseQueryBuilder Template
- *** A CHT PopFavoriteQueries_HMB Template

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

NAME: HNDSQL8.APP
TITLE: BROWSES HMB (SQL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: BROWSES HMB
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates CHT HandyMarkerBrowse (HMB) used in an SQL setting with a new locator template called HandyMarkerBrowseLocatorControlEx.

It illustrates these concepts:

- *** Filtering locator filters on the current sort column.
- *** Locator drill-down "breadcrumb" trail until clear button.
- *** Display either CHT query language or SQL.
- *** Execute Locator with tab or with GO button.
- *** Report Example passing browse filter to report

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

**CATEGORY:
BROWSES LBX**

NAME: HNDLBXDM.APP
TITLE: BROWSES LBX (ISAM) [V10.18.08]
DATE: 8/24/2019
CATEGORY: BROWSES LBX
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates primarily a record-marking template, that assists in marking records using ordinary Clarion list boxes.

That template is called ListBoxBrowseExtender or "LBX" for short.

Some Features Illustrated In This Application:

- * ListBoxBrowseExtender Has Green Bar.
- * LBX has multiple optional Query Control templates (_LBX)
- * Optionally use ORDER BY In Query Control For Mult-Column Sort.
- * ListBoxBrowseExtender Honours Retain-Record.
- * ListBoxBrowseExtender Generates Data Queue
- * ListBoxBrowseExtender connects update forms
- * ListBoxBrowseExtender has queue auto-fill on startup
- * LBX has optional pre-filter to limit auto-fill on startup
- * LBX has selectable fill-queue methods for local or remote data

Four example procedures illustate some powerful LBX features;

* PROCEDURE: ListBoxBrowseExtenderInQueueQuery()

This procedure illustrates a query template called InQueueQueryControl_LBX. This control filters the records already in the list box queue, into a secondary queue called "the shadow queue" and displays the result in the same browse. Clearing this filter resets to the starting or "Main" queue and displays the original record set. All of this except the original fill-queue happens without ANY further file access being needed.

* PROCEDURE: ListBoxBrowseExtenderSortOrderFieldFilter()

This procedure illustrates a template called SortOrderFieldFilter_LBX. This control requires the user to enter only a string of characters to match something contained in the sort-order field. As above, matching data are inserted into the shadow queue and are displayed in the same browse. As above, also, clearing this filter resets to and displays the "Main" queue record set.

* PROCEDURE: ListBoxBrowseExtenderUniversalQueryControl()

This procedure illustrates a template called UniversalQueryControl_LBX. This control when empty fills the queue with all table/view records unless a pre-filter is added to the template config. Click the "Queue Data" button to switch to the "Shadow Queue" and enter a query. These records are filtered from the Main queue into the Shadow queue. Click back on the "Queue Data" button and the browse switches back to the Main queue with all original records intact.

* PROCEDURE: ListBoxBrowseExtenderPopFavoriteQueries()

This procedure populates a template called PopFavoriteQueries_LBX. This control opens a table called PreQueries and scans it for queries designated for the view name defined on the template. Queries may be optionally added, or deleted at run time and can be selected from a popup menu. The PreQueries table is defined in HND0.DCT from which it can be imported into your own dictionary. Prequeries may be ISAM or SQL as required by your application.

*** HNDLBXSQL.APP ***

NAME: HNDLBXSQL.APP
TITLE: BROWSES LBX (SQL) [V10.19.08]
DATE: 8/24/2019
CATEGORY: BROWSES LBX
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It illustrates these concepts:

- *** Filling the browse queue directly from a single file
- *** Filling the browse queue from multiple files using a view
- *** Filling the browse queue manually by embedding
- *** Filling the browse queue using class method SQLSelect
- *** Filling the browse queue from a remote CHT Server Type 2
- *** Filling the browse queue from a remote CHT Server Type 3
- *** Incorporating an Update Form with UpdateButtons_LBX.
- *** A hand-code created/managed query control.

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

*** HNDLBXSQLEX.APP ***

NAME: HNDLBXSQLEX.APP
TITLE: BROWSES LBX (SQL) [V10.19.08]
DATE: 8/24/2019
CATEGORY: BROWSES LBX
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It illustrates these concepts:

- *** An LBX browse with UniversalQueryControl_LBX
- *** An LBX browse with InQueueQueryControl_LBX
- *** Connecting an update form with UpdateButtons_LBX

*** PROCEDURE: LBXWithUniversalQueryControl()

The browse queue is loaded without pre-filter initially from the data table. The query control provides for entry of CHT queries which will re-fill the "Main" queue with a new subset of data directly from the data table while "Main Queue Data" is displayed. Click over to the "Shadow Queue Data" and CHT queries in the query control will display subsets of the "Main" data queue without data table access being exercised. Clearing the Main Queue Data query control reloads the Main queue from the data table.

*** PROCEDURE: LBXWithInQueueQueryTemplate()

The browse queue is loaded without pre-filter initially from the data table. The query control provides for entry of a CHT query string. Records matching the query string are inserted into the "Shadow" queue which is then displayed. "Main" queue records remain undisturbed. Clearing the query empties the shadow queue and re-displays the Main queue. No file access is required after the initial data load from the data tables. A reset from file button, reloads the main queue from the data tables.

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

NAME: HNDPEOPLE_LBX.APP
TITLE: BROWSES LBX (ISAM) [V10.19.08]
DATE: 8/24/2019
CATEGORY: BROWSES LBX
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

LBX "FillFrom" tabs used:
 FillFromFile() -- Local Data Tables
 FillFromView() -- Local Data Tables
 FillFromCHTServer() -- Remote Data Tables

Upcoming versions of this application will add additional browses based on LBX "FillFrom" tabs:
 FillFromSQLSelect -- Local Data Tables
 FillByEmbedding -- Local or Remote Data Tables
 FillFromUCRServer -- Remote Data Tables

Hot Key Alerts:

- ** MouseLeft - Mark a single record, unmarks any others.
- ** ShiftMouseLeft - Mark range between selected and click.
- ** CtrlShiftMouseLeft - Mark from selected to click position.
- ** CtrlMouseLeft - Toggle record marked/unmarked.
- ** CtrlMouseLeft - On header, pops up column hide menu.
- ** 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 up.
- ** DownKey - Switch to single record mode and move down.
- ** CtrlUp - Move cursor up with no marking or unmarking.
- ** CtrlDown - Move cursor down with no marking.
- ** 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.

**CATEGORY:
BROWSES MIXED**

NAME: HNDACCES.APP
TITLE: BROWSES MIXED (SQL) [V10.19.07]
DATE: 8/24/2019
CATEGORY: BROWSES MIXED
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates the added speed and convenience of the NATIVE SQL filtering capability built into the CHT query classes.

The CHT template QueryParsingFilterControl (QPFC) has the ability to generate optimised NATIVE SQL directly from your real-language CHT queries.

The SQL is then sent to your data base using View{{PROP:SQLFilter}} rather than via ABC.SetFilter(). We've also added an optional QUERY BUILDER (OBJ.QBInterface()) to the ? button of the

QueryParsingFilterControl (QPFC) template.

The CHT-generated SQL appears in the WHERE clause component of the SELECT string sent to your data base. The chief advantage of this is speed. Clarion's file drivers do not always translate clarion filter strings into effective SQL. In some cases, depending how complex your query, ABC.Setfilter() causes the driver to literally read large portions of the data base, forcing the driver to manage filtering rather than letting the data base handle it.

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

*** HNDRLDN.APP ***

NAME: HNDRLDN.APP
TITLE: BROWSES MIXED (SQL) [V10.19.02]
DATE: 8/24/2019
CATEGORY: BROWSES MIXED
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

In this application, we're illustrating some new SQL "drill-down" capabilities build into the locators available for ExplorerBrowse, LocatorOverrideControl and HandyMarkerBrowse. When any of these locators is configured to act as a filtering locator you have the option of asking the template to remember the last locator query and to re-apply it with your next locator query. The more locates performed before the locator clear button is clicked, the deeper you drill down into the current data set.

A new feature on the HandyMarkerBrowseProcessButton and HandyMarkerBrowseReportButton templates lets you opt to have the marked record set translated into an SQL IN(x,y,z) request to be sent to a report or process rather than a queue of marked records. Default behavior is as before, to send a queue.

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. You do not need a copy of the Microsoft Access to test this, and a copy of NORTHWIND.MDB is included in the install. The application will find the data base and register it with ODBC if not already registered.

*** HNDMIXEDBROWSESQJJOINTTEST.APP ***

NAME: HNDMIXEDBROWSESQJJOINTTEST.APP
TITLE: BROWSES MIXED (SQL) [V10.19.02]
DATE: 8/24/2019
CATEGORY: BROWSES MIXED
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application illustrates a mixed browse extension, SQL Join set of browses consisting of HandyMarkerBrowses (Parent, Child1 and Child3) and ExplorerBrowse (Child2).

This app was developed as a test application to study ABC Browse Class reset behaviour by tracing with HandySQLTrace. It was used in revisions to CHT Browse Classes and templates made late May 2012.

The purpose of those revisions was to eliminate double and triple SQL requests sent from the ABC Browse classes whenever the browse was reset by an order change or a filter change.

It is the nature of templates of one type to generate code together in a given embed point (say priority 9500) in the order that the templates are populated on the window. This rule of thumb does not hold when populating two templates with different types into the same embed priority point. One template type will generate as a group, then the other, regardless of the order in which they're populated.

When creating a JOIN SET with mixed browse extensions such as in this application, CHT had to devise a way to get the templates to populate in a consistent order that matched the developer's design for browse filling order. A JOINed set of browses must always fill in the order PARENT, CHILD1, CHILD2, CHILD3.

If the templates generate into the same embed point in the wrong order (due to template types having been mixed together) the firing order of browses can become jumbled and events can collide such that browses fill in the wrong order and SQL "Chatter" results.

Our latest implementation of HandyMarkerBrowse and ExplorerBrowse extensions has overcome this native template-generation-order behaviour (where order matters) by generating into named, hidden, embed points based on each template's position in the JOIN SET heirarchy, regardless of the template type, allowing mixed template types in a given JOIN SET.

This application in conjunction with HNDSQLTRACE.EXE can be used to illustrate just how much CHT's latest Browse Extension templates can improve the speed and efficiency of your SQL browses.

*** HNDPOPFAVORITEQUERIESEXAMPLE.APP ***

NAME: HNDPOPFAVORITEQUERIESEXAMPLE.APP
TITLE: BROWSES MIXED (ISAM) [V10.19.02]
DATE: 8/24/2019
CATEGORY: BROWSES MIXED
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates two CHT browse extension templates, ExplorerBrowse (EXPB) and HandyMarkerBrowse (HMB) used in conjunction with recently introduced query templates that improve on the query management and recall aspects of CHT Query Language queries on your browses.

PopFavoriteQueries_EXPB is paired with Explorerbrowse (hence the _EXPB extension) and PopFavoriteQueries_HMB is paired with HandyMarkerBrowse (hence the _HMB extension).

Our short form for these PopFavoriteQueries templates is PFQ which is the short form or acronym used in the name of this application.

These query templates, while configured identically and operating identically, place template-specific embeds intended for the target CHT browse extension that they are paired with. This makes the overall template code less complex and easier to maintain and evolve than a multi-target template like QueryParsingFilterControl.

All PopFavoriteQueries (PFQ) templates are CHT's response to a request at the 2012 CHT Developers' conference for improved query management and re-use in QueryParsingFilterControl.

There is a PopFavoriteQueries_QPFC template that is paired with QueryParsingFilterControl to add features to it, but in most cases it is easier and more efficient to remove QPFC from your browse and replace it with one of these, depending on which CHT browse extension you are using.

CATEGORY: COSMETICS

*** HNDMNSPL.APP ***

NAME: HNDMNSPL.APP
TITLE: COSMETICS (SPLASH WIZ) [V10.18.12]
DATE: 8/24/2019
CATEGORY: COSMETICS
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Procedures in this application are available as JUMP START procedures via the CHTJumpStartProceduresUtility Template under the topic Main And Splash.

The application illustrates a variety of splash screens available as instant CHT Jumpstart wizards aimed at helping you improve the cosmetic appearance of your splash screens.

Import one of these "Jumpstarts" instantly, then quickly customize to your own requirements with text and images of your own choosing.

CATEGORY:

CALCULATORS

*** HNDCALCULATOR.APP ***

NAME: HNDCALCULATOR.APP
TITLE: CHT Calculator Demo [V10.18.01.0287]
DATE: 8/24/2019
CATEGORY: Calculators
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Extension template EmbedCalculatorFunctions attaches a CHT class called HNDCalcControl to your application procedure, making its functions available to the procedure.

Control template HandyCalcEntryControl_ECF drops an entry control and a button on your procedure window. These controls work in tandem. A numerical value in the entry control is passed to the calculator - accessed via the button control - and the calculator passes a calculated result back into the control. A thread option on the template provides for passing data back to the entry control without closing the calculator, which can be left open on top of the application window.

Until the close of 4th quarter 2008, this template and the HNDCalcControl class should be considered beta and feature incomplete. If you have feature suggestions, for the class or the templates, please do feel free to make them on the CHT forum using the "Suggestions" setting.

Any questions, feel free to ask them on the CHT forum.
Gus M. Creces
The Clarion Handy Tools Page

CATEGORY: DATES AND CALENDARS

*** HNDCAL.APP ***

NAME: HNDCAL.APP
TITLE: HNDDates Class Demonstration [V10.18.07.0317]
DATE: 8/24/2019
CATEGORY: Dates And Calendars
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

As usual, subscribers with a SOURCE code subscription receive the code module as well as the header. DLL version subscribers receive only the header module and the class code is in one of the CHT DLLs.

The application illustrates the syntax used to call many of the functions available in the HNDDates class. There are also two separate implementations of the calendar showing how you can modify the look of the CHT calendar by calling methods like OBJ.SetCalendarIcon() and embedding new calendar image properties in that method.

*** HNDCMCAL.APP ***

NAME: HNDCMCAL.APP
TITLE: COM Calendar Demo [V10.18.08.0209]
DATE: 8/24/2019
CATEGORY: Dates And Calendars
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

A CHT control template called HandyCOMCalendarControl helps you drop this control onto any application window and instantiate a Clarion-style interface to the OCX ICalendar control.

Any questions, feel free to ask: news.cwhandy.com

NAME: HNDDTPK.APP
TITLE: Demonstrate HNDDatePicker Class [V10.18.12.0213]
DATE: 8/24/2019
CATEGORY: Dates And Calendars
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

A CHT control template called HandyCOMDatePickerControl helps you drop this control onto any application window and instantiate a Clarion-style interface to the OCX IDatePicker control.

Any questions, feel free to ask: news.cwhandy.com

*** HNDNETDATETIMEDEMO.APP ***

NAME: HNDNETDATETIMEDEMO.APP
TITLE: CHT .NET DATE TIME CLASSES [V10.18.04.0262]
DATE: 8/24/2019
CATEGORY: Dates And Calendars
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It illustrates how the EmbedNetDateTimeClasses template can give your procedure the ability to obtain detailed system and file date time information, for example time zone information, UTC date time information, and more.

HNDNETDATETIME.DLL is a viable .NET assembly that also works with Clarion.NET.

Any questions feel free to ask.

**CATEGORY:
EMAIL SMTP**

*** HNDJUMPSTARTSMTPMAILTESTS.APP ***

NAME: HNDJUMPSTARTSMTPMAILTESTS.APP
TITLE: SMTP Jumpstarts Example [V10.18.12]
DATE: 8/24/2019
CATEGORY: EMAIL SMTP
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates the use of several CHT SMTP Jump Start templates as listed below:

- ** SMTP Configure
- ** SMTP Send Without Attachments
- ** SMTP Send With Attachments
- ** SMTP Send With Compression
- ** SMTP Send HTML Advanced
- ** SMTP Send TEXT Advanced

These are available to be imported into your application and create procedures which can be called from your own procedures or menus.

From your application pull down the "Application Menu", then the "Utility Templates" menu. From here, select the "AACHTJumpStartProceduresUtility" and move to the 8th tab entitled "Email Template". The template names listed above are visible from that tab.

CATEGORY:

EMAIL EXAMPLE

*** HNDBULKSMTPMAIL.APP ***

NAME: HNDBULKSMTPMAIL.APP
TITLE: Bulk SMTP Email [V10.19.01.0118]
DATE: 8/24/2019
CATEGORY: Email Example
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application also includes a set of separate attachment encryption/decryption procedures. While attachment encryption is invoked on the fly as you send mail, the decryption procedure can be used to unencrypt attachments received from this application.

The Clarion Handy Tools offer a very broad spectrum of email support in the form of templates, OOP classes, examples and Jump Start procedures.

Any questions, feel free to ask: news.cwhandy.com

*** HNDCDODM.APP ***

NAME: HNDCDODM.APP
TITLE: CDO Send Mail With Interface [V10.18.09.0233]
DATE: 8/24/2019
CATEGORY: Email Example
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The email template illustrated is EMBEDCDOMAILFUNCTIONS.

Procedure CDOSendMailWithInterface() attaches the HNDMSCDO class via a CHT template called EmbedCDOMailFunctions. The resulting procedure can be attached to any application to create a robust, run-ready, custom, email procedure complete with user interface using our Jump Start Template interface, "CDO Email Send".

CDO Email provides for adding attachments to your emails where the attachment may not be located locally, such as files situated on password protected (or not) ftp servers, http servers, https servers.

Any questions, feel free to ask: news.cwhandy.com

*** HNDMAILME.APP ***

NAME: HNDMAILME.APP
TITLE: HandyMailMeControl Templates Demo [V10.17.11.0190]
DATE: 8/24/2019
CATEGORY: Email Example
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

- * HandyMailMeControl - window hot link to email address
- * HandyMailMeControl_EWF - window hot link to email address, advanced

In about 30 seconds you can build procedures that, like the examples provided, launch your default email client and prepare it for sending email, and even auto send the mail without user intervention.

While HandyMailMeControl adds a HNDWindow class instance to your procedure unconditionally, the _EWF template is a child of EmbedWindowFunctions (hence, _EWF in the name) and only becomes visible when that template is already present on your application. It shares the HNDWindow class instance populated by EmbedWindowFunctions.

*** HNDMLADSM.APP ***

NAME: HNDMLADSM.APP
TITLE: Advanced Send-SMTP-Mail Demo [V10.18.12.0358]
DATE: 8/24/2019
CATEGORY: Email Example
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

These are ADVANCED send procedures using higher level HNDSMTP features such as HTML email and attachment encryption and compression.

All procedures in this application are available as JUMP START procedures via the CHTJumpStartProceduresUtility Template. Import any of them into your application as a model from which to work forward.

NOTE: This application was revised in JUNE 2015 to take advantage of changes to the HNDSMTP class to incorporate CLARUNEXT.DLL to handle SMTP sends. This DLL, as of Build 19B.02.00 replaces the HNDMSM.DLL used prior to that point.

Any questions, feel free to ask: news.cwhandy.com

*** HNDMLSMTP.APP ***

NAME: HNDMLSMTP.APP
TITLE: Basic Send-SMTP-Mail Demo [V10.19.07.0508]
DATE: 8/24/2019
CATEGORY: Email Example
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

These are BASIC send procedures using a minimum of HNDSMTP features, designed specifically for easy understanding. More complex mail procedures are available in other demonstration applications, for example, HNDMLADSM.APP.

All procedures in this application are available as JUMP START procedures via the CHTJumpStartProceduresUtility Template. Import any of them into your application as a model from which to work forward.

NOTE: This application was revised in JUNE 2015 to take advantage of changes to the HNDSMTP class to incorporate CLARUNEXT.DLL to handle SMTP sends. This DLL, as of Build 19B.02.00 replaces the HNDMSM.DLL used prior to that point.

Any questions, feel free to ask: news.cwhandy.ca

*** HNDSMTPNETDEMO.APP ***

NAME: HNDSMTPNETDEMO.APP
TITLE: CHT .NET SMTP [V10.18.09.0312]
DATE: 8/24/2019
CATEGORY: Email Example
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The email template in use is called EMBEDNETSMTPCLASSES. The CHT Class used by this template is the HNDNETSMTPCLASS located in files HNDNETSMTP.INC and HNDNETSMTP.CLW.

This application and the HNDNetSMTP class, while doing some practical work, serve the primary purpose of allowing developers to explore the implications of using actual .NET DLLs in their Clarion applications. The operating system environment on which this application runs, must have at least Microsoft Framework 2 installed.

The technique of using "COM Interop" to DLLs created with a .NET compiler such as C#.NET, VB.NET or Clarion.NET provide a wealth of opportunity to expand the capabilities of CHT and consequently the

capabilities of Win32 Clarions. We'd appreciate your feedback in testing this simple application in your development environments.

This application now includes procedure(s) using the template CHTBestPracticeEmbedClass as an example for developers.

*** HNDSOURCEMAPIDEMO.APP ***

NAME: HNDSOURCEMAPIDEMO.APP
TITLE: MAPI / ZIP / ABC Source Procedure [V10.18.12.0322]
DATE: 8/24/2019
CATEGORY: Email Example
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Source_EmbedMAPIFunctions template
Source_EmbedZipFunctions template

The SMTP template provides an embeddable CHT class called HNDsmtp located in HNDIMAIL.INC and HNDIMAIL.CLW.

The ZIP template provides an embeddable CHT Class called HNDComZipClass located in HNDZIPPR.INC and HNDZIPPR.CLW.

If you want to learn to hand code with these two classes, this application is a perfect learning platform. The procedures provided may also be imported directly into your apps to provide pre-built functionality.

*** HNDSOURCESMTPDEMO.APP ***

NAME: HNDSOURCESMTPDEMO.APP
TITLE: SMTP / ZIP / ABC Source [V10.18.0011]
DATE: 8/24/2019
CATEGORY: Email Example
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Source_EmbedSMTPFunctions template
Source_EmbedZipFunctions template

The SMTP template provides an embeddable CHT class called HNDsmtp located in HNDSMTP.INC and HNDSMTP.CLW.

The ZIP template provides an embeddable CHT Class called HNDComZipClass located in HNDZIPPR.INC and HNDZIPPR.CLW.

If you want to learn to hand code with these two classes, this application is a perfect learning platform. The procedures provided may also be imported directly into your apps to provide pre-built functionality.

**CATEGORY:
EXTENDED CONTROLS - CLOCKS**

*** HNDDIGITALCLOCKDEMO.APP ***

NAME: HNDDIGITALCLOCKDEMO.APP
TITLE: CHT Digital Clock Demo [V10.18.12.0214]
DATE: 8/24/2019
CATEGORY: Extended Controls - Clocks
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

To do achieve this, the template invokes a CHT COM wrapper class called HNDDigitalClockClass which loads and calls into a CHT COM DLL called HNDDGTLCLOCK.OCX.

No coding is required. Standard Clarion sizing and hiding properties apply to the digital clock container.

Any questions, feel free to ask: news.cwhandy.com

CATEGORY: EXTENDED CONTROLS - EVENT LOGGING

*** HNDDEVENTLOGGERDEMO.APP ***

NAME: HNDDEVENTLOGGERDEMO.APP
TITLE: HandyComEventLoggerControl [V10.18.11]
DATE: 8/24/2019
CATEGORY: Extended Controls - Event Logging
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

CHT's Event Logger Control provides an easy-to-use COM interface inside your application procedures for writing Events, Errors and Warnings to the SYSTEM APPLICATION LOG where they can be found and read using EVENTVWR.MSC.

See HNDEVEER.INC, HNDEVEER.CLW and HNDEVEER.TLB to study the header, code module and method interfaces respectively.

Any questions, feel free to ask: news.cwhandy.com

CATEGORY: EXTENDED CONTROLS - FIELD AUTOCOMPLETE

*** HNDDDROPDOWN.APP ***

NAME: HNDDDROPDOWN.APP
TITLE: Autocomplete Dropdown Example [V10.18.08.0216]
DATE: 8/24/2019
CATEGORY: Extended Controls - Field AutoComplete
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

In this case, we're autocompleting a "State Name" field from a file called hndstatesnew.tps. By simply typing 3 or 4 letters into this field a dropdown appears which displays the closest matches for the typed value. By continuing to type beyond 4 characters a matching value is brought into your autocomplete control.

Alternatively, with the down arrow key, you can scroll down into the dropdown control to select one of the values that appear there.

Adding to the autocomplete file: If you type a unique value and press CtrlEnter, the entered value is added to the autocomplete source file.

Deleting from the autocomplete file: If you bring an existing value from the autocomplete file into the autocomplete entry control and press CtrlDelete you are able to delete a value from the autocomplete source file.

The Clarion Handy Tools Page

CATEGORY: EXTENDED CONTROLS - MAIL SLOTS

*** HNDMAILSLOTCL.APP ***

NAME: HNDMAILSLOTCL.APP
TITLE: CHT Mail Slot Client [V10.18.12.0180]

DATE: 8/24/2019
CATEGORY: Extended Controls - Mail Slots
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application is half of a two-application set and acts as client that sends messages - up to 212 characters long - to a server application called HNDMAILSLOTSV.APP.

An MS Windows "Mailslot" provides for inter-application/inter-process messaging. Applications sharing a network domain can talk to one another and give or receive commands over a Mailslot.

The template incorporates a CHT COM MailSlot Class Wrapper called HNDMailSlotClass into your procedure. See HNDMAIOT.INC, HNDMAIOT.CLW and HNDMAIOT.TLB to study the header, code module and method interfaces respectively.

*** HNDMAILSLOTSV.APP ***

NAME: HNDMAILSLOTSV.APP
TITLE: CHT MailSlot Server [V10.18.07.0182]
DATE: 8/24/2019
CATEGORY: Extended Controls - Mail Slots
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application is half of a two-application set and acts as a server that picks up messages - up to 212 characters long - posted from the client application called HNDMAILSLOTCL.APP. An MS Windows "Mailslot" provides for inter-application/inter-process messaging. Applications sharing a network domain can talk to one another and give or receive commands over a Mailslot.

The template incorporates a CHT COM MailSlot Class Wrapper called HNDMailSlotClass into your procedure. See HNDMAIOT.INC, HNDMAIOT.CLW and HNDMAIOT.TLB to study the header, code module and method interfaces respectively.

**CATEGORY:
EXTENDED CONTROLS - NAMES AND ADDRESSES**

*** HND6.APP ***

NAME: HND6.APP
TITLE: CHT Address Controls Demo [V10.18.12.0230]
DATE: 8/24/2019
CATEGORY: Extended Controls - Names And Addresses
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The HandyAddressControl Template provides three free-form address capture controls similar to those found in Microsoft Outlook. This set of controls allows users to type their name and address in a more-or-less free form style while still allowing the developer to capture address components in separate data base fields.

The HandyRulerControl Template provides horizontal and vertical rulers combined with an image control to provide scaling for drawings and images.

The HandySliderControl Template provides a standard MS Windows slider control that can be used in a variety of settings from browses to image scaling.

As of the O7B Build some extra code has been added to the underlying classes to allow the address control to remain completely empty if so required.

CATEGORY:

EXTENDED CONTROLS - SCRIPTING

*** HNDWINDOWSSCRIPTHOSTDEMO.APP ***

NAME: HNDWINDOWSSCRIPTHOSTDEMO.APP
TITLE: CHT Windows Script Host Demo [V10.18.11.0185]
DATE: 8/24/2019
CATEGORY: Extended Controls - Scripting
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Windows Script Host provides a wide variety of low-level operating system configuration functions normally enabled in WScript, JScript and VBScript. Microsoft describes Windows Script Host as follows: "Windows Script Host (WSH), a feature of the Microsoft Windows 2000/XP family of operating systems, is a powerful multi-language scripting environment ideal for automating system administration tasks.

Scripts running in the WSH environment can leverage the power of WSH objects and other COM-based technologies that support Automation, such as Windows Management Instrumentation (WMI) and Active Directory Service Interfaces (ADSI), to manage the Windows subsystems that are central to many system administration tasks."

This template and its underlying wrapper class, HNDScriptHostClass based in modules HNDWSHOM.INC /HNDWSHOM.CLW /HNDWSHOM.TLB provide you the equivalent of running windows administration scripts from inside your Clarion application, using Clarion syntax rather than writing WScript, JScript or VBScript.

Any questions, feel free to ask: news.cwhandy.com

CATEGORY: EXTENDED CONTROLS - SHELL

*** HNDWINDOWSSHELLDEMO.APP ***

NAME: HNDWINDOWSSHELLDEMO.APP
TITLE: CHT Windows Shell Demo [V10.18.09.0179]
DATE: 8/24/2019
CATEGORY: Extended Controls - Shell
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Windows Script Host provides a wide variety of low-level operating system configuration functions normally enabled in WScript, JScript and VBScript. Microsoft describes Windows Script Host as follows: "Windows Script Host (WSH), a feature of the Microsoft Windows 2000/XP family of operating systems, is a powerful multi-language scripting environment ideal for automating system administration tasks. Scripts running in the WSH environment can leverage the power of WSH objects and other COM-based technologies that support Automation, such as Windows Management Instrumentation (WMI) and Active Directory Service Interfaces (ADSI), to manage the Windows subsystems that are central to many system administration tasks."

This template and its underlying wrapper class, HNDScriptHostClass based in modules HNDWSHOM.INC /HNDWSHOM.CLW /HNDWSHOM.TLB provide you the equivalent of running windows administration scripts from inside your Clarion application, using Clarion syntax rather than writing WScript, JScript or VBScript.

Any questions, feel free to ask: news.cwhandy.com

CATEGORY: EXTENDED CONTROLS - SLIDERS AND PROGRESS

*** HNDPROGRESSBAR.APP ***

NAME: HNDPROGRESSBAR.APP

TITLE: COM Progress Bar Template Demo [V10.19.04.0180]
DATE: 8/24/2019
CATEGORY: Extended Controls - Sliders And Progress
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

With a progress bar, users can follow the progress of a lengthy operation. A progress bar may show either an approximate percentage of completion (determinate) or indicate that an operation is ongoing (indeterminate). Microsoft usability studies have shown that users are aware of response times of over one second. Consequently, you should consider operations that take two seconds or longer to complete to be lengthy and in need of progress feedback.

See HNDPRGBR.INC, HNDPRGBR.CLW and HNDPRGBR.TLB to study the header, code module and method interfaces respectively.

Any questions, feel free to ask: news.cwhandy.com

*** HNDSLIDER.APP ***

NAME: HNDSLIDER.APP
TITLE: Demonstrate HNDComSlider Class [V10.18.09.0174]
DATE: 8/24/2019
CATEGORY: Extended Controls - Sliders And Progress
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

A CHT control template called HandyComSliderControl helps you drop this control onto any application window and instantiate the Clarion-style interface to the COM ISlider control.

Any questions, feel free to ask: news.cwhandy.com

**CATEGORY:
EXTENDED CONTROLS - SPLITTERS**

*** HNDSPITTERTEST.APP ***

NAME: HNDSPITTERTEST.APP
TITLE: CHT Splitter Example [V10.18.09.0143]
DATE: 8/24/2019
CATEGORY: Extended Controls - Splitters
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The EmbedHNDSplitter extension template is applied to the procedure first. This is the parent of the two related control templates. Then HandySplitBarVertical and HandySplitBarHorizontal can be applied as required by the developer.

This example has applied the ABC resizer to allow window and control resize. The EmbedHNDSplitter template allows for optional interaction with ABC resizer. The default setting is to disable this ABC resizer interaction. When ABC resizer is enabled, it's up to the developer to apply that template to the procedure and configure it to his liking.

**CATEGORY:
EXTENDED CONTROLS - WEB LINKS**

*** HNDVISITME.APP ***

NAME: HNDVISITME.APP
TITLE: VisitMe Control Templates Demo [V10.18.09.0178]
DATE: 8/24/2019

CATEGORY: Extended Controls - Web Links

CLASSES: [List App Classes](#)

TEMPLATES: [List App Templates](#)

DESCRIPTION:

- * HandyVisitMeControl - window hotlink to website
- * HandyVisitMeControl_EWF - window hot link to website, advanced

In about 30 seconds you can build procedures that, like the examples provided, launch your default browser and visit the URL of your choice.

While HandyVisitMeControl adds a HNDWindow class instance to your procedure unconditionally, the _EWF template is a child of EmbedWindowFunctions (hence, _EWF in the name) and only becomes visible when that template is already present on your application. It shares the HNDWindow class instance populated by EmbedWindowFunctions.

CATEGORY: FILE AND MEMORY COMPRESSION

*** HNDTESTSEMISILENTHZOCOMPRESSION.APP ***

NAME: HNDTESTSEMISILENTHZOCOMPRESSION.APP
TITLE: JumpStartSemiSilentHZOCompression [V10.18.09.0186]
DATE: 8/24/2019
CATEGORY: File And Memory Compression
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The JumpStartSemiSilentHZOCompression(). function takes the following parameters:

- xSrc - A root source directory for files to be compressed
 - xTgt - A target directory and file name of an hzo container
 - xQry - A CHT File Query to describe the files to be compressed
 - xPwd - A password to be used in data-encrypting the files
 - xSubs - True or false to indicate search subdirectories of the root source
 - xWinHndl - Window handle of parent window
 - *xProgress - Reference to progress variable local to parent window
- The function returns the number of files compressed if successful and zero if it fails.

Any questions, feel free to ask: news.cwhandy.com

*** HNDTESTSEMISILENTHZODECOMPRESSION.APP ***

NAME: HNDTESTSEMISILENTHZODECOMPRESSION.APP
TITLE: JumpStartSemiSilentHZODeCompression [V10.18.09.0184]
DATE: 8/24/2019
CATEGORY: File And Memory Compression
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The JumpStartSemiSilentHZODeCompression(). function takes the following parameters:

- xSrcFile - Name of the .HZO container to be processed
- xTgt - A target directory into which the .HZO is decontainerized
- xQry - A CHT File Query to describe the files to be decompressed
- xPwd - A password to be used in data-decrypting the files
- xSubs - True or false to include subdirectories of the root source
- xWinHndl - Window handle of parent window
- xProgress - Reference to progress variable local to parent window

The function returns the number of files decompressed if successful and zero if it fails.

Any questions, feel free to ask: news.cwhandy.com

*** HNDTESTSILENTHZOCOMPRESSION.APP ***

NAME: HNDTESTSILENTHZOCOMPRESSION.APP
TITLE: JumpStartSilentHZOCompression [V10.18.09.0215]
DATE: 8/24/2019
CATEGORY: File And Memory Compression
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The JumpStartSilentHZOCompression(). function takes the following parameters:

xSrc - A root source directory for files to be compressed
xTgt - A target directory and file name of an hzo container
xQry - A CHT File Query to describe the files to be compressed
xPwd - A password to be used in data-encrypting the files
xSubs - True or false to indicate search subdirectories of the root source

The function returns the number of files compressed if successful and zero if it fails.

Any questions, feel free to ask: news.cwhandy.com

*** HNDTESTSILENTHZODECOMPRESSION.APP ***

NAME: HNDTESTSILENTHZODECOMPRESSION.APP
TITLE: JumpStartSilentHZODECompression [V10.18.09.0206]
DATE: 8/24/2019
CATEGORY: File And Memory Compression
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The JumpStartSilentHZODECompression(). function takes the following parameters:

xSrcFile - The .HZO file to be decontainerized
xTgt - A target directory into which the .HZO files are decompressed
xQuery - A CHT File Query to describe the files to be decompressed
xPwd - A password to be used in data-decrypting the files
xSubs - True or false to include subdirectories of the root source

The function returns the number of files decompressed if successful and zero if it fails.

Any questions, feel free to ask: news.cwhandy.com

*** HNDZDEMO.APP ***

NAME: HNDZDEMO.APP
TITLE: BulkShrinkFiles/BulkGrowFiles [V10.19.08]
DATE: 8/24/2019
CATEGORY: File And Memory Compression
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application uses the generic template EmbedDiskFunctions to attach a class called HNDDISK to your application procedures. The application illustrates the use of two HNDDISK functions:

OBJ.BulkShrinkFiles() -> OBJ.ShrinkFile()
OBJ.BulkGrowFiles() -> OBJ.GrowFile()

The HNDDISK Class provides several different Compression and Decompression methods including memory and file compression and decompression as well as ZIP compression and decompression.

The SHRINK and GROW functions provide CHT proprietary file compression and decompression capabilities that are pure Clarion code using internal Clarion functions created for that purpose.

During SHRINK, the target file is read into a memory buffer, the memory buffer is compressed in-memory (by Clarion's COMPRESS function) and then written to disk as binary bytes.

During GROW, the target file is read into a memory buffer, the memory buffer is decompressed in-memory (by Clarion's DECOMPRESS function) and then written to disk as binary bytes.

The SHRINK version of a file is given the extension .HNZ, and the original file name and extension ae

retained inside a file header embedded in the .HNZ file. Hence, the GrowFile() function is able to re-establish a fully expanded version of the original file including the file extension.

SHRINK and GROW are also optionally encryption capable. When encryption is enabled, the compressed file is given the extension .HZY.

CHT proprietary compression and encryption provided by these functions are extremely useful for secure file transport via email, FTP and HTTP since the decompression phase can only be performed by another CHT application that is made aware of the encryption password or by a CHT Clarion procedure built into your CHT email application, CHT FTP application or CHT Web Server.

See also HNDCPYDM.APP for more examples including file encryption.

*** HNDZTEST.APP ***

NAME: HNDZTEST.APP
TITLE: File And Memory Compression Demo [V10.18.09.0174]
DATE: 8/24/2019
CATEGORY: File And Memory Compression
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates the low-level file reading, writing capabilities of the HNDDosFile class.

- * Class attached to procedure using EmbedObject template.
- * Read a file into a CSTRING memory buffer as is or compressed in a single pass.
- * Write a compressed CSTRING memory buffer to a file as is (still compressed) or uncompressed.
- * Compress a CSTRING memory buffer in-memory.
- * Uncompress a CSTRING memory buffer in-memory.
- * Compression capabilities of this class come from it's parent class HNDShrink.
- * File reading-writing capabilities come from Clarion's DOS file driver.
- * More...

CATEGORY: FILE AND MEMORY ENCRYPTION

*** HNDXTEST.APP ***

NAME: HNDXTEST.APP
TITLE: Encryption Test Bench Demo [V10.18.09.0168]
DATE: 8/24/2019
CATEGORY: File And Memory Encryption
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates the low-level file and memory encryption, capabilities of the HNDEncrypt class.

- * Class attached to procedure using EmbedObject template
- * 128 Bit encryption inherits abilities from Windows Crypto API
- * Files encrypted or decrypted in a single pass
- * Optional file compression/decompression
- * Encrypt a STRING or CSTRING memory buffer in-memory
- * Decrypt a STRING or CSTRING memory buffer in-memory
- * Convert Encrypted Data to Base64 for web transfer
- * Optional encryption key generation
- * Encrypt passwords with embedded key using SetPass(), GetPass()
- * Encrypt buffers with embedded key using AutoEncrypt(),AutoDecrypt()
- * More...

Any questions, feel free to ask: news.cwhandy.com

CATEGORY: FILE BLOB FUNCTIONS

*** HNDFBLOB.APP ***

NAME: HNDFBLOB.APP
TITLE: File To Blob Example [V10.18.08.0243]
DATE: 8/24/2019
CATEGORY: File BLOB Functions
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Please note, that the code in the EntryPointMain() procedure has very little to do with file to blob or blob to file. This code looks up a file name to be sent to a blob and manages the data base that stores the blob record. Your blob data base may look nothing like the file ZStore that we are using here.

The salient point is that when calling the functions JumpStartFileToBlob() and its inverse, JumpStartBlobToFile(), you are responsible for creating, fetching and saving the data base records in the blob-containing data table. The functions provided, will insert the files to, and extract files independently from, your blob record. Data base I/O is up to you. However the I/O required is standard ABC stuff which you should already know how to do.

We have demonstrated here a couple of HNDDisk class functions to help us with our popup menu, when someone selects a file to be restored or deleted, for example. These have no direct relationship with the two blob functions being demonstrated.

Any questions, feel free to ask: news.cwhandy.ca

**CATEGORY:
FILE MOVING, COPYING**

*** HNDCPYDM.APP ***

NAME: HNDCPYDM.APP
TITLE: Copy, Move, Rename, Compress, Encrypt [V10.18.10.0230]
DATE: 8/24/2019
CATEGORY: File Moving, Copying
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

- * JumpStartBulkFileCompressExpandWithProgress() - Bulk file compression and expansion with optional encryption and decryption
- * JumpStartBulkFileContainerizeWithProgress() - Bulk file compression to a container (.HZO) file with optional encryption.
- * JumpStartBulkFileDeContainerizeWithProgress() - Bulk file decompression from container (.HZO) file
- * JumpStartBulkFileCopyWithProgress () - Bulk file copying to a target directory
- * JumpStartBulkFileEncryptDecryptWithProgress() - Bulk file encryption/decryption with optional compression/decompression
- * JumpStartBulkFileEraseWithProgress() - Bulk file removal
- * JumpStartBulkFileMoveWithProgress() - Bulk file moving
- * JumpStartBulkFileRenameWithProgress() - Bulk file renaming

A query allows you to describe the files to be acted on. Or files to be acted on may be dropped from MS Explorer.

Any questions, feel free to ask: news.cwhandy.ca

*** HNDFSYNC.APP ***

NAME: HNDFSYNC.APP
TITLE: CHT File Synchronizer [V10.18.12]
DATE: 8/24/2019
CATEGORY: File Moving, Copying
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This "CHT Utility Application" is a folder synchronizer capable of keeping the files in one or more folders on

your system or network in synch. In this implementation, file masks may be specified individually for each synch directory record.

Use it to periodically back up your CHT files to the standard accessory directory or to back up before a WEBUPDATER session. A pre-built sync file called hndfsync.tps has been provided for this purpose. Macros in this sync file can resolve your \hndapps\, \libsrc\ and \template\ locations based on your Clarion + CHT installation specifics.

This application illustrates the kinds of things that can be done with several CHT templates used in combination.

The major templates used are:

- * EmbedDiskFunctions - to attach the HNDDisk class
- * ExplorerBrowse - to attach the HNDBrowse class
- * EmbedWindowFunctions - to attach the HNDWindow class

Any questions, feel free to ask on the CHT forum: [http://news.cwhandy.ca/LOG\\$](http://news.cwhandy.ca/LOG$)

CATEGORY: FILE SEARCH AND REPLACE

*** HNDSCHDM.APP ***

NAME: HNDSCHDM.APP
TITLE: CHT Search And Replace Demo [V10.18.08.0230]
DATE: 8/24/2019
CATEGORY: File Search And Replace
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

- OBJ.BulkSearchReplaceText()
- OBJ.BulkFileRename()

BulkSearchReplaceText() -
Searches through text and program code files of all types in the target directory to replace words, commands, phrases.

BulkFileRename() -
Adds a prefix and removes spaces and extended characters from filenames in the target directory.

CATEGORY: GENPACK

*** HNDAPPSXML2HTMLGENPACK.APP ***

NAME: HNDAPPSXML2HTMLGENPACK.APP
TITLE: XML2HTML GENPAK [V10.18.12]
DATE: 8/24/2019
CATEGORY: GENPACK
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDAPPSXML2HTMLGENPACK.APP is a convenient, multi-app solution for multi-app recompile when required.

This app has only a splash screen for self-documentation purposes and serves only as a repository with which to group and multi-compile a set of related CHT apps, without using CHT Project Builder.

It contains the following CHT XML2HTML applications:

HNDXML2HTMLGEN.APP
HNDDOCUMENTBUILDER.APP
HNDSLFSVCUS.APP
HNDSLFSVXMLDATASVR.APP

HNDBULKMAILBATCHER.APP
HNDBULKNETMAILPROMO.APP
HNDTPLDOCGEN.APP
HNDAPPSDOCGEN.APP

*** HNDAPPSZIPNPOSTGENPACK.APP ***

NAME: HNDAPPSZIPNPOSTGENPACK.APP
TITLE: ZIP-N-POST GENPAK [V10.18.12]
DATE: 8/24/2019
CATEGORY: GENPACK
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This app has only a splash screen for self-documentation purposes and serves as a repository with which to group and multi-compile a set of related CHT apps, without using CHT Project Builder.

It contains the following 15, CHT ZIP N'POST applications:

- 1 HNDZIPNEMAILSNAP.APP
- 2 HNDZIPNEMAILSNAPMAPI.APP
- 3 HNDZIPNPOSTTOCHTSERVER.APP
- 4 HNDZIPNPOSTTOCHTSERVEROCX.APP
- 5 HNDZIPNPOSTTOCHTSETUP.APP
- 6 HNDZIPNPOSTTOCHTXSA.APP
- 7 HNDZIPNEMAILBBOT.APP
- 8 HNDZIPNEMAILBBOTCDO.APP
- 9 HNDZIPNEMAILBBOTMAPI.APP
- 10 HNDZIPNEMAILDCDO.APP
- 11 HNDZIPNEMAILMAPI.APP
- 12 HNDZIPNEMAILNET.APP
- 13 HNDZIPNEMAILSMTP.APP
- 14 HNDZIPNFTP.APP
- 15 HNDZIPNFTPNET.APP

*** HNDLBXGENPACK.APP ***

NAME: HNDLBXGENPACK.APP
TITLE: BROWSES LBX GENPAK [V10.19.08]
DATE: 8/24/2019
CATEGORY: GENPACK
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This app has only a splash screen for self-documentation purposes and serves as a repository with which to group and multi-compile a set of related CHT apps, without using CHT Project Builder.

It contains the following 4 CHT LBX-related applications:

- hndlboxdm.app
- hndpeople_lbx.app
- hndlboxsql.app
- hndlboxsqllex.app

*** HNDSEVERGENPACK.APP ***

NAME: HNDSEVERGENPACK.APP
TITLE: HTTP SERVER GENPAK [V10.18.12]
DATE: 8/24/2019
CATEGORY: GENPACK
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This app has only a splash screen for self-documentation purposes and serves as a repository with which to group and multi-compile a set of related CHT apps, without using CHT Project Builder.

It contains the following 22 CHT Server Applications:

hndclientsv.app
hndclientsvlean.app
hndclientsvlean1view.app
hndclientsvlean4view.app
hndfilsv.app
hndfilsvspecial.app
hndlrvsv.app
hndmtssv.app
hndmtsnng.app
hndnewslinksvsvr.app
hndslfsv.app
hndslfsvxmldatasvr.app
hndslfsvxmlfilesvr.app
hndslfsvxsa.app
hndslfsvxsa3part.app
hndsvlv.app
hndeventserver.app
hndfileserverlean.app
hndinstallationserver.app
hndteamdocserver.app
hndslfsvxsa2part_100_82.app
hndslfsvxsa2part_100_88.app

*** HNDUTILITYAPPS.APP ***

NAME: HNDUTILITYAPPS.APP
TITLE: BATCH-BOT GENPAK [V10.18.09]
DATE: 8/24/2019
CATEGORY: GENPACK
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This app has only a splash screen for self-documentation purposes and serves only as a repository with which to multi-compile a set of CHT apps, without using CHT Project Builder.

It contains the following CHT Utility applications:

HNDENCRYPT.APP
HNDMAILCDO.APP
HNDMAILMAPI.APP
HNDMAILSMTP.APP
HNDMAKEHZO.APP
HNDSLFCMD.APP
HNDXML2HTMLGEN.APP
and, of course. HNDUTILITYAPPS.APP

*** HNDXSAKIT1.APP ***

NAME: HNDXSAKIT1.APP
TITLE: XSA-KIT-1 GENPAK [V10.18.08]
DATE: 8/24/2019
CATEGORY: GENPACK
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It serves only as a convenience to pull these four projects together for easy examination, compilation and customization where necessary.

The projects contained in CHT-XSA-KIT-01 are:
CHT-XSA-KIT-01.01 - HNDSLFCMDXSA.CWPROJ
CHT-XSA-KIT-01.02 - HNDSLFSVXSA.CWPROJ
CHT-XSA-KIT-01.03 - HNDSETUPXSA.CWPROJ
CHT-XSA-KIT-01.04 - HNDINSTALLXSA.CWPROJ

Because this is an .APP it is distributed only to subscribers to the full CHT Toolkit, as it uses CHT templates in its design.

CATEGORY: HTTP CLIENT TYPE 1

*** HNDBRWCL.APP ***

NAME: HNDBRWCL.APP
TITLE: HTTP CLIENT (FORUM) [V10.19.08]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 1
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The URL being accessed is <http://news.cwhandy.ca/>.

The above URL and an HTTPS version may also be accessed directly in your favorite browser. Note that our SSL certificate for HTTPS access is self-signed and must be installed on your computer prior to accessing [HTTPS://news.cwhandy.ca](https://news.cwhandy.ca). The correct certificate is installed by our Webupdater installer for current CHT Subscribers.

The user interface looks and interacts exactly as it would were you to use IE directly but without some of the navigational clutter that normally appears on the IE window.

This client application serves to illustrate how to log in and then make a variety of data requests and data updates from a Clarion client application using the IE browser class called HNDInternetExplorer and a template called HandyComInternetExplorer.

This application now includes procedure(s) using the template CHTBestPracticeEmbedClass as an example for developers.

*** HNDFORUMVIEWERLITE.APP ***

NAME: HNDFORUMVIEWERLITE.APP
TITLE: HTTP CLIENT (FORUM) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 1
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application communicates with the CHT's Forum Login Page's DOM (Document Object Model) using methods provided in CHT class, HNDInternetExplorer.

CHT Forum Login information is assumed to already be in your system's registry, because your system has been configured for The Clarion Handy Tools by our WEBUPDATER tool.

Further, a set of popup selections containing pre-configured query macros is available for the forum postings (messages) page. These macros are again inserted directly into the correct web variables on the page followed by a page refresh request. This has the same effect as if the user typed these values directly into the posting page query control.

*** HNDSRCT.APP ***

NAME: HNDSRCT.APP
TITLE: CHT Web Scriptor [V10.19.03]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 1
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Use it to create HTML/Javascript page and form scripts, Javascript Subroutines, HTML/Text email scripts and Cascading Style Sheets.

Once signed onto your TYPE 1 CHT Server Project Directory this scripter will interact with your locally running server to give you real-time page previewing using live data from your server.

The "page-follow" component of this application interacts with your live, local test "browser server" to show you the "JDO" data packages sent from the server to your web page.

These JDO packages appear in a tab on the scripter interface to eliminate guess work as to what the server back end views are sending (by way of data, javascript and html) to your web pages on a page-by-page basis.

CATEGORY: HTTP CLIENT TYPE 3

*** HNDCLIENTCL.APP ***

NAME: HNDCLIENTCL.APP
TITLE: Data Client TYPE 3 (Forum) [V10.19.08]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This client application serves to illustrate how to log in and then make a variety of data requests and data updates from a Clarion client application (using the HNDCLNT class) to a CHT Server configured for Client Server Mode.

NOTE: This client uses a two-piece login (Serial number, Last Name) and does not require the email address. The server (HNDCLIENTCL.APP) omits the Email Address Key on the server login configuration dialog to make this possible.

This applicaiton uses templates HandyWebClientBrowseProcedure and HandyWebClientUpdateFormProcedure to build the web browse and web update form, respectively.

*** HNDCLIENTCLEAN1VIEW.APP ***

NAME: HNDCLIENTCLEAN1VIEW.APP
TITLE: Data Client TYPE 3 (Views) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The server, HNDCLIENTSVLEAN4VIEW.APP, is an example Client Server with multiple back-end data views built on several .TPS based demo files called "People", "Folks", "Product" and "NGMember" used in other CHT demo applications.

While this 1-view application accesses only one of that server's four views, other applications planned in this series, will successively access 2 views, 3 views and finally 4 views. This series of 4 client applications and the server (HNDCLIENTSVLEAN4VIEW.APP) will be used to create a training video series and a set of support documents to teach development of CHT Client Server applications.

This client's browse uses exclusively PopFavoriteQueries_CSBL to accompany ClientServerBrowseListBox.

ClientServerBrowseListBox populates the QueryParsingFilterControl Template template natively, but to illustrate how it can be done, we've manually pulled the QPFC off these browses and replaced them with the PopFavoriteQueries_CSBL template to handle all queries.

Use this Client and it's accompanying server as a study example when you begin building your own CHT Web Client Server applications.

This application is entirely template-built and jumpstart-built and contains almost no hand-embedded code.

NAME: HNDCLIENTCLEAN4VIEW.APP
TITLE: Data Client TYPE 3 (Views) [V10.19.03]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The server, HNDCLIENTSVLEAN4VIEW.APP, is an example Client Server with four back-end data views built on a several .TPS based demo files called "People", "Folks", "Product" and "NGMember" used in other CHT demo applications.

This client's browses use exclusively PopFavoriteQueries_CSBL to accompany ClientServerBrowseListBox

ClientServerBrowseListBox populates the QueryParsingFilterControl Template template natively, but to illustrate how it can be done, we've manually pulled the QPFC off these browses and replaced them with the PopFavoriteQueries_CSBL template to handle all queries.

Use this Client and it's accompanying server as a study example when you begin building your own CHT Web Client Server applications.

This application is entirely template-built and jumpstart-built and contains almost no hand-embedded code.

*** HNDCLIENTSVLEAN.APP ***

NAME: HNDCLIENTSVLEAN.APP
TITLE: CLIENT SERVER TYPE 3 [V10.19.01]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application is designed to act as a starting point for all CHT "Client Servers" intended to provide interactive, remote data services to a Clarion client application. The developer can begin adding server-side views to this without having to design any of the server "plumbing" such as configuration or server client maintenance.

This special "lean" version has had all non-essential componentry removed and focusses entirely on CHT templates and classes required to build a fast, efficient CHT Client Data Server.

This server application has a matching demonstration client called HNDCLIENTCLEAN.APP. Since the server has no views for the client to connect to, this client contains only basic connection capability. The application dictionary is HNDCLIENTSVLEAN.DCT.

This server is able to provide both compressed and encrypted data streams consisting of browse data, form data, report and process data based on interactive queries placed from the client side targeted at pre-determined data views (single or multi-table) that the developer adds into the server.

This application is entirely template-built and jumpstart-built and contains almost no hand-embedded code.

*** HNDFILCL.APP ***

NAME: HNDFILCL.APP
TITLE: File Client TYPE 3 (Basic) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This "Basic" version has had all non-essential componentry removed and focusses entirely on CHT templates and classes required to build a fast, efficient CHT secure file client. It uses a 3-piece login.

This client application has two matching demonstration server called HNDFILSV.APP and HNDFILSVLEAN.APP which are enabled to provide secure upload and download services.

This client is able to provide both compressed and encrypted file movement services without requiring either pre-compression or pre-encryption of the upload or download files.

This application is entirely template-built and jumpstart-built and contains almost no hand-embedded code.

*** HNDFILECLIENTLEAN.APP ***

NAME: HNDFILECLIENTLEAN.APP
TITLE: File Client TYPE 3 (Lean) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This server utilizes a three-part login consisting of LAST NAME, SERIAL NUMBER and EMAIL ADDRESS that must match the same three values of a user in the server's "subscriber" data table.

This special "lean" version has had all non-essential componentry removed and focusses entirely on CHT templates and classes required to build a fast, efficient CHT secure file client.

This client application has a matching demonstration client called HNDFILESERVERLEAN.APP which is enabled to provide secure upload and download services. The server uses a dictionary called HNDFILERSERVERLEAN.DCT.

This client is able to provide both compressed and encrypted file movement services without requiring either pre-compression or pre-encryption of the upload or download files.

This application is entirely template-built and jumpstart-built and contains almost no hand-embedded code.

*** HNDFLCLN.APP ***

NAME: HNDFLCLN.APP
TITLE: File Client TYPE 3 (Advanced) [V10.19.07]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

A combination of HTTP Client and Server are intended to take the place of FTP file transfer, providing much more secure and speedy file transfer services in both directions.

The application acts as secure file transfer client in combination with TYPE 3, authenticating servers called HNDFILSV.APP, HNDSVLV.APP and HNDFILSVLEAN.APP which act as secure file transfer servers.

It is an "Advanced" version client which includes a sophisticated login-configuration dialog, and separate upload and download procedures that include local and remote file lookups.

This client is able to provide both compressed and encrypted file movement services without requiring either pre-compression or pre-encryption of the upload or download files.

This application is entirely template-built and jumpstart-built and contains almost no hand-embedded code.

*** HNDMTSCL.APP ***

NAME: HNDMTSCL.APP
TITLE: Data Client TYPE 3 (Forum) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 3

CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This client application serves to illustrate how to log in and then make a variety of data requests and data updates from a Clarion client application (using the HNDCLNT class) to a CHT Server configured for Client Server Mode. This application and accompanying server provide the theoretical backdrop for the CHT Client Server course.

NOTE: This client uses a two-piece login (Serial number, Last Name) and does not require the email address. The server (HNDCLIENTCL.APP) omits the Email Address Key on the server login configuration dialog to make this possible.

This applicaiton uses templates HandyWebClientBrowseProcedure and HandyWebClientUpdateFormProcedure to build the web browse and web update form, respectively.

**CATEGORY:
HTTP CLIENT TYPE 4**

*** HNDINSTALLMYFILES.APP ***

NAME: HNDINSTALLMYFILES.APP
TITLE: CLIENT TYPE 3 (INSTALL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application first connects to HNDFILSVSPECIAL.APP to download an installation files list approved at the server end. Next it downloads the listed files and places them in the designated directories below the installer's configured target directory. Files are only downloaded when the CRC value provided for any file is different than the CRC of the file already installed.

The server auto-encrypts and auto-compresses files at the request of this client and files are auto-decrypted and auto-decompressed on arrival. Replaced files are copied to the backup directory before replacement is made. When the download file is an .HZO container, this client extracts it to the download directory, while honoring the sub-paths indicated by the .HZO contents.

A configuration procedure in this application is provided in order to change the installation server as well as the local target and backup directories. The installer always recalls it's last installation log from the config menu.

*** HNDINSTBA.APP ***

NAME: HNDINSTBA.APP
TITLE: CLIENT TYPE 4 (INSTALL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This very basic installation example illustrates a basic installation JumpStart called JumpStartGetRemoteInstallationFileBasic() that pulls together several CHT classes, namely HNDHTTP, HNDZSTORE, HNDISKEX and HNDBUFFER, to provide developers an easy-to-implement means of building across-the-web self updating into their products.

This application downloads and installs three standard CHT .HZO container files built with another CHT demo application called HNDSETUP.APP. All of these installation containers contain compressed material (mostly images) and one of them, the third, is both compressed and encrypted. While running the app, the user can adjust the target installation and backup locations.

NOTE:
The jumpstart called JumpStartGetRemoteInstallationFileBasic() mentioned above is presently available

from the CHT Jumpstart interface on the "Installation Functions" tab along with two others that auto enable website security and serial number authentication against a standard CHT web server.

*** HNDSETUPMKR.APP ***

NAME: HNDSETUPMKR.APP
TITLE: CLIENT TYPE 4 (INSTALL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The HNDSLFSVXSA.APP server is configured as an XSA-2 configurable server.

First, use HNDSETUP.APP to build an .HZO container with the files to be installed. Set up a server based on HNDSLFSV.APP to deliver your e-commerce web pages as well as your installation containers. An example container comes with this application for preliminary, local testing. That container is called HNDXSAINSTALLEEXAMPLE.HZO.

Once you've tested this app as-is for installaation from a local copy of HNDSLFSVXSA.APP, then use a renamed copy of this application and complete the template prompts in your version of this app to build your install following its example.

Once the template dialogs have been satisfactorily completed, compile the application (your renamed copy of this applcation) to build the installer/updater for the HZO container configured in the template.

Finally, use, HNDSETUP.APP to upload your installation containers to your website using the HTTP upload dialog built into that application.

*** HNDSETUPMKR_XSA3_HC.APP ***

NAME: HNDSETUPMKR_XSA3_HC.APP
TITLE: CLIENT TYPE 4 (INSTALL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

First, use HNDSETUP.APP to build an .HZO container with the files to be installed. Set up a server based on HNDSLFSVXSA3PART.APP to deliver your e-commerce web pages as well as your installation containers.

Then use a renamed copy of this application and complete the template prompts in your version of this app to build your install following its example.

Once the template dialogs have been satisfactorily completed, compile the application (your renamed copy of this applcation) to build the installer/updater for the HZO container configured in the template.

Since the HNDSLFSVXSA3PART.APP server uses a "Customer" file where installation privileges are established and controlled, a customer maintenance application called HNDSLFSVCUS.APP has been added to the toolkit. This server uses a "Customer" table located in an SQLITE container file named HNDSLFSV.SQLITE and the maintenance app is fix-coded to access and maintain that same custome table.

In this example installer, the "Authentication Model" switch has been set to position 4 which indicates a hard-coded authentication string consisting of 3 discrete pieces separated by commas. Hence the name of this example: HNDSETUPMKR_XSA3_HC.APP (with HC for hard-coded).

Finally, use, HNDSETUP.APP to upload your installation containers to your website using the HTTP upload dialog built into that application.

*** HNDSETUPMKR_XSA3_LOOKUP.APP ***

NAME: HNDSETUPMKR_XSA3_LOOKUP.APP
TITLE: CLIENT TYPE 4 (INSTALL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 4
CLASSES: [List App Classes](#)

TEMPLATES: [List App Templates](#)

DESCRIPTION:

First, use HNDSETUP.APP to build an .HZO container with the files to be installed. Set up a server based on HNDSLFSVXSA3PART.APP to deliver your e-commerce web pages as well as your installation containers.

Then use a renamed copy of this application and complete the template prompts in your version of this app to build your install following its example.

Once the template dialogs have been satisfactorily completed, compile the application (your renamed copy of this application) to build the installer/updater for the HZO container configured in the template.

Since the HNDSLFSVXSA3PART.APP server uses a "Customer" file where installation privileges are established and controlled, a customer maintenance application called HNDSLFSVCUS.APP has been added to the toolkit. This server uses a "Customer" table located in an SQLITE container file named HNDSLFSV.SQLITE and the maintenance app is fix-coded to access and maintain that same custome table.

In this example installer, the "Authentication Model" switch has been set to position 5 which indicates a dynamic, user-configurable authentication string consisting of 3 discrete pieces separated by commas.

Hence the name of this example: HNDSETUPMKR_XSA3_LOOKUP.APP (with LOOKUP to indicate looked up in the registry configuration).

When this installer first starts, unconfigured, it pops up a dialog asking for the three-pieces of the login, in this case, emailaddress, serialnumber, lastname. This string must match entries in the server's customer table so the exact components of the login string can be determined by the developer. If entered incorrectly the installer will re-display the login configuration dialog untill properly entered.

If the customer;s account login is changed or the account is expired, the installer will ask for the credentials to be re-entered. Otherwise, if credential information is satisfied, the install proceeds without pausing or asking for any configuraiton.

Finally, use, HNDSETUP.APP to upload your installation containers to your website using the HTTP upload dialog built into that application.

*** HNDSETUPMKR_XSA_HC.APP ***

NAME: HNDSETUPMKR_XSA_HC.APP
TITLE: CLIENT TYPE 4 (INSTALL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This installer has the XSA username and password hard coded (hence "HC" in the name) into the application from the template where it is entered.

A hard-coded username/password of this sort in your installer, assumes you don't need download control over specific individuals. For instance, your installer is provided to a company's IT department to update your product installation on their network. You can still turn these installers off or force them to be updated periodically by using the server config, to change the required username/password.

Other, more individually controllable, variations of this installer are possible, of course. And, in addition to this one, we have several examples in the CHT Toolkit. Search in /HNDAPPS/ for HNDSETUPMKR_XSA*.APP.

First, use HNDSETUP.APP to build an .HZO container with the files to be installed. Set up a server based on HNDSLFSVXSA.APP to deliver your e-commerce web pages as well as your installation containers.

Then use a renamed copy of this application and complete the template prompts in your version of this app to build your install following its example.

Once the template dialogs have been satisfactorily completed, compile the application (your renamed copy of this application) to build the installer/updater for the HZO container configured in the template.

Finally, use, HNDSETUP.APP to upload your installation containers to your website using the HTTP upload dialog built into that application.

*** HNDSETUPMKR_XSA_LOOKUP.APP ***

NAME: HNDSETUPMKR_XSA_LOOKUP.APP
TITLE: CLIENT TYPE 4 (INSTALL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

First, use HNDSETUP.APP to build an .HZO container with the files to be installed. Set up a server based on HNDSLFSVXSA.APP to deliver your e-commerce web pages as well as your installation containers. Configure a global username/password combination to control installation access by your installers.

Then use a renamed copy of this application and complete the template prompts in your version of this app to build your install following its example.

Once the template dialogs have been satisfactorily completed, compile the application (your renamed copy of this application) to build the installer/updater for the HZO container configured in the template.

Finally, use, HNDSETUP.APP to upload your installation containers to your website using the HTTP upload dialog built into that application.

RE: SECURITY

The username/password required by this application for access to your server, is not hard coded as in HNDSETUPMKR_XSA_HC.APP. The first time this application is run, it pops up a dialog which requires the user to enter the correct username/password combination that you have provided to them. Thereafter this dialog does not re-appear unless it fails to successfully contact your server (because perhaps you changed the required server credentials).

At that point, new access information can be provided to the user-base, or not, depending on the update privileges to which they are entitled by your update policy.

*** HNDSLFCL.APP ***

NAME: HNDSLFCL.APP
TITLE: HTTP CLIENT - TYPE 4 (FILES) [V10.19.08]
DATE: 8/24/2019
CATEGORY: HTTP CLIENT TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application illustrates how to make secure file upload and download requests to and from a CHT server root directory or any directory below server root.

A log-in dialog has been added, under an unlock image button on the tool bar. This produces an XSA, 2-part authentication dialog for server-level authentication prior to transfer access being allowed. The USER NAME and PASS CODE #, entered must match the same two values established on the server.

HNDSLFCL.APP illustrates two secure file transfer functions: HTTPGetFileSecure() and HTTPPutFileSecure(). The functions provide a progress window with a cancel button.

The functions are hosted in CHT Class HNDCLIENT and are made available in the application by a template called EmbedHTTPClientFunctions.

This application requires no .DCT (Clarion Dictionary).

See also C# client CHTSNAPGET.EXE for command-line and batch-script downloading from HNDSLFSV.APP.

CATEGORY:
HTTP SERVER TYPE 1

*** HNDEVENTSERVER.APP ***

NAME: HNDEVENTSERVER.APP
TITLE: BROWSER SERVER TYPE 1 [V10.19.02]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 1

CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The fundamental principles and practices of this server's design are pure ABC. If you understand Clarion, and ABC Object Oriented Principles, you can duplicate a server of this type using The Clarion Handy Tools. It may be used as the basis for building forward any other type of "Browser" server design or simply as a model on which to base a new design.

Web scripts constituting the "user interface" for this "Browser" server are built, as usual, using CHT Scripter. These scripts are fully previewable and are live-data-interactive during script modification and testing.

CHT Servers of this type do not rely on any generic web server software like Microsoft Internet Information Server. Our servers tie directly into the TCP/IP libraries built into the Windows NT/XP/Vista operating systems. Beyond that, all Internet listening, sending activities, data packaging, compression, encryption and so forth are written in Clarion OOP source code available as part of CHT's standard installation.

Principle server templates in use are: EmbedBrowserServer and BrowserServerInitializationControls, accompanied by a group of page-builder templates that are able to generate XHTML or pure Javascript Data Objects. Included with the CHT Tool Kit is a multi-function editor/scripter (HNDSCRPT.APP) which is used to design the client-side/browser side pages and forms, including CSS (Cascading Style Sheets) and JS (Javascript) Subroutines.

NOTE: This is a TYPE 1 browser server medical events server -- from which to learn and with which to build forward. Some back end views portions of the server are already installed and working.

*** HNDINSTALLATIONSERVER.APP ***

NAME: HNDINSTALLATIONSERVER.APP
TITLE: BROWSER SERVER TYPE 1 [V10.19.02]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 1
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

First, the server provides a web browse and form that let you choose one of three CHT versions and the associated installer(s).

Second, the server provides authentication for the downloaded installers, connecting current subscribers and rejecting expired subscribers.

When a current subscriber connects via the updater, a list of current installation HZO's is provided from a standard .TPS or SQL file that you control. The installer can use this list to download those files for which CRCs are different from local containers it may have present.

*** HNDLRNSV.APP ***

NAME: HNDLRNSV.APP
TITLE: BROWSER SERVER TYPE 1 [V10.19.02]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 1
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application is entirely template-built and contains almost no hand-embedded code.

NOTE: This is a starting-point server from which to build forward. It proceeds to the point of login but no further. The "messages" back-end portion of the operation is ready to be grafted in by the developer.

*** HNDMTSNG.APP ***

NAME: HNDMTSNG.APP
TITLE: BROWSER SERVER TYPE 1 [V10.19.08]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 1
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

We've had this server - or an earlier variation of it - in active service since 2002. It has evolved since then, as have the templates and classes on which it is based. But the fundamental principles and practices in its design have remained pure ABC. If you understand Clarion, and ABC Object Oriented Principles, you can duplicate a server of this type using The Clarion Handy Tools.

CHT Servers of this type do not rely on any generic web server software like Microsoft Internet Information Server. It ties directly into the TCP/IP libraries built into the Windows operating system. Beyond that, all Internet listening, sending activities, data packaging, compression, encryption and so forth are written in Clarion OOP source code available as part of the fully-integrated CHT tool kit.

Principle server templates in use are: EmbedBrowserServer and BrowserServerInitializationControls, accompanied by a group of page-builder templates that are able to generate XHTML or pure Javascript Data Objects. Included with the CHT Tool Kit is a multi-function editor/scripter (HNDSCRPT.APP) which is used to design the client-side/browser side pages and forms, including CSS (Cascading Style Sheets) and JS (Javascript) Subroutines.

The dictionary file used by this application is HNDMTSNG.DCT.

NOTE: This is a browser server version of CHT's support forum -- from which to learn and with which to build forward. The back end views portions of the server are already installed and working.

CATEGORY: HTTP SERVER TYPE 2

*** HNDNEWSLINKSLISTSVR.APP ***

NAME: HNDNEWSLINKSLISTSVR.APP
TITLE: HTTP TYPE 4 UCR SERVER [V10.19.02]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 2
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

In this particular example, application functionality has been limited to acting as a dynamic data, page server and providing several example UCR\$ (User Custom Request) procedure calls that may be called from any HTML web browser, desktop, tablet or phone.

This server was branched from HNDSLFSV.APP to demonstrate building dynamic-data web pages based on CHT XML2HTMLGEN techniques. Take note that it has had its default base page changed from INDEX.HTML to TEMPLATESBROWSE.UCR. This is still a TYPE 4 server, however, because these UCR\$ requests do not require prior authentication against a server password or against a user file, page transactions with this server do not require prior login, and do not operate on a SESSION Identifier.

This server utilizes a function called AutoGenHTMLFromXML_VER02 exported from HNDXML2HTMLGEN.APP to convert standard CHT XML template files into HTML. This is the same functionality that triggers an HTML file to be generated from CHTSNAPEDIT.EXE when the "Preview" menu key is clicked.

Calling any one of this server's specialized UCR\$ functions, triggers the return of a dynamic data page based on one of CHT's .XML page templates.

Data rows inserted into a temporary copy of "000templatesbrowse.xml" file before processing to an HTML file, are read from a TPS table called "TPLHEADER".

*** HNDSLFSVXMLDATASVR.APP ***

NAME: HNDSLFSVXMLDATASVR.APP

TITLE: HTTP SERVER TYPE 2 [V10.19.02]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 2
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

In this particular example, application functionality has been limited to acting as a dynamic data, page server and providing several example UCR\$ (User Custom Request) procedure calls that may be called from an HTML web link.

This server was branched from HNDSLFSV.APP in Sept 2017 to demonstrate building dynamic-data web pages based on CHT XML2HTMLGEN techniques. Take note that it has had it's default base page changed from INDEX.HTML to UCRBROWSES.HTML in order to simplify testing and navigating to the UCR\$ functions test page.

It utilizes a function called AutoGenHTMLFromXML_VER02 exported from HNDXML2HTMLGEN.APP to convert standard CHT XML template files into HTML. This is the same functionality that triggers an HTML file to be generated from CHTSNAPEDIT.EXE when the F5 key is pushed.

Calling any one of this server's specialized UCR\$ functions, triggers the return of a dynamic data page based on one of CHT's .XML page templates.

One example, UCR\$=NEWSLINKSLISTCOLUMNLABELS.UCR& calls a function in this server called NewsLinksListColumnLabels() which processes an XML template file called 000newslinkslist.xml.

*** HNDSLFSVXMLFILESVR.APP ***

NAME: HNDSLFSVXMLFILESVR.APP
TITLE: HTTP SERVER TYPE 2 [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 2
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

In this particular example, application functionality has been limited to acting as a dynamic data, page server and providing several example UCR\$ (User Custom Request) procedure calls that may be called from an HTML web link.

This server was branched from HNDSLFSV.APP in October 2017 to demonstrate building dynamic-data web pages based on CHT XML2HTMLGEN techniques. Take note that it has had it's default base page changed from INDEX.HTML to UCRFILESERVER.HTML in order to simplify testing and navigating to the UCR\$ functions test page.

It utilizes a function called AutoGenHTMLFromXML_VER02 exported from HNDXML2HTMLGEN.APP to convert standard CHT XML template files into HTML. This is the same functionality that triggers an HTML file to be generated from CHTSNAPEDIT.EXE when the F5 key is pushed.

Calling any one of this server's specialized UCR\$ functions, triggers the return of a dynamic data page based on one of CHT's .XML page templates.

NOTE: At time of writing, NOV 1, 2017 this server is under construction with expected completion before the end of 2017, at which time this "under-construction" notation will be removed.

*** HNDTYPE4SERVERPEOPLE.APP ***

NAME: HNDTYPE4SERVERPEOPLE.APP
TITLE: TYPE 4 UCR SERVER - PEOPLE [V10.19.08]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 2
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

In this particular example, application functionality has been limited to acting as a dynamic data, page

server and providing several example UCR\$ (User Custom Request) procedure calls that may be called from any HTML web browser, desktop, tablet or phone.

This server was branched from HNDSLFSV.APP to demonstrate building dynamic-data web pages based on CHT XML2HTMLGEN techniques. Take note that it has had its default base page changed from INDEX.HTML to PEOPLEBROWSE.UCR. This is still a TYPE 4 server, however, because these UCR\$ requests do not require prior authentication against a server password or against a user file. Page transactions with this server do not require prior login, and do not operate on a SESSION identifier.

This server utilizes a function called AutoGenHTMLFromXML_VER02 exported from HNDXML2HTMLGEN.APP to convert standard CHT XML template files into HTML. This is the same functionality that triggers an HTML file to be generated from CHTSNAPEDIT.EXE when the "Preview" menu key is clicked.

Calling any one of this server's specialized UCR\$ functions, triggers the return of a dynamic data page based on one of CHT's .XML page templates.

Data rows inserted into a temporary copy of "000peoplebrowse.xml" file before processing to an HTML file, are read from a TPS file called "HNDO.TPS" which contains, a table called "PEOPLE".

*** HNDTYPE4UCRSERVERTEMPLATES.APP ***

NAME: HNDTYPE4UCRSERVERTEMPLATES.APP
TITLE: HTTP TYPE 4 UCR SERVER [V10.19.08]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 2
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

In this particular example, application functionality has been limited to acting as a dynamic data, page server and providing several example UCR\$ (User Custom Request) procedure calls that may be called from any HTML web browser, desktop, tablet or phone.

This server was branched from HNDSLFSV.APP to demonstrate building dynamic-data web pages based on CHT XML2HTMLGEN techniques. Take note that it has had its default base page changed from INDEX.HTML to TEMPLATESBROWSE.UCR. This is still a TYPE 4 server, however, because these UCR\$ requests do not require prior authentication against a server password or against a user file, page transactions with this server do not require prior login, and do not operate on a SESSION identifier.

This server utilizes a function called AutoGenHTMLFromXML_VER02 exported from HNDXML2HTMLGEN.APP to convert standard CHT XML template files into HTML. This is the same functionality that triggers an HTML file to be generated from CHTSNAPEDIT.EXE when the "Preview" menu key is clicked.

Calling any one of this server's specialized UCR\$ functions, triggers the return of a dynamic data page based on one of CHT's .XML page templates.

Data rows inserted into a temporary copy of "000templatesbrowse.xml" file before processing to an HTML file, are read from a TPS table called "TPLHEADER".

CATEGORY: HTTP SERVER TYPE 3

*** HNDCLIENTSV.APP ***

NAME: HNDCLIENTSV.APP
TITLE: CLIENT SERVER TYPE 3 [V10.19.08]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This server application requires a matching demonstration client called HNDCLIENTCL.APP and a dictionary called HNDCLIENTSV.DCT.

The server provides both compressed and encrypted data streams consisting of browse data, form data, report and process data based on interactive queries placed from the client side targeted at pre-determined data views (single or multi-table) provided in the server.

NOTE: This server is configured for a two-piece login. It omits the Email Address Key on the BrowserServerInitialization controls template. In a two-piece login only the Login ID and Last Name (in this case) are used. This is a client server for reading and posting remote messages -- from which to both learn and with which to build forward. The back end views portions of the server are already installed and working.

*** HNDCLIENTSVLEAN1VIEW.APP ***

NAME: HNDCLIENTSVLEAN1VIEW.APP
TITLE: CLIENT SERVER TYPE 3 [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The developer can add multiple server-side views to a server of this sort (HNDCLIENTSV.APP) without having to design any of the server "plumbing" such as configuration or server client maintenance.

This server application has a matching demonstration client called HNDCLIENTCLEAN1VIEW.APP, which illustrates connecting to this server and browsing its VIEWS across the web.

This server is able to provide both compressed and encrypted data streams consisting of browse data, form data, report and process data based on interactive queries placed from the client side targeted at pre-determined data views (single or multi-table) that the developer adds into the server.

This application is entirely template-built and jumpstart-built and contains almost no hand-embedded code.

*** HNDCLIENTSVLEAN4VIEW.APP ***

NAME: HNDCLIENTSVLEAN4VIEW.APP
TITLE: CLIENT SERVER TYPE 3 [V10.19.07]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The developer can add multiple server-side views to a server of this sort (HNDCLIENTSV.APP) without having to design any of the server "plumbing" such as configuration or server client maintenance.

This server application has a matching demonstration client called HNDCLIENTCLEAN1VIEW.APP, which illustrates connecting to this server and browsing its VIEWS across the web.

This server is able to provide both compressed and encrypted data streams consisting of browse data, form data, report and process data based on interactive queries placed from the client side targeted at pre-determined data views (single or multi-table) that the developer adds into the server.

This application is entirely template-built and jumpstart-built and contains almost no hand-embedded code.

*** HNDFILESERVERLEAN.APP ***

NAME: HNDFILESERVERLEAN.APP
TITLE: FILE SERVER TYPE 3 [V10.19.02]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This special "lean" version has had all non-essential componentry removed and focusses entirely on CHT templates and classes required to build a fast, efficient CHT Secure File Server.

The server has a configuration dialog to handle naming and locating the server's configuration and members files. It provides full control over which directories are visible to the client and which files are available for download by the client. A user browse and edit form are also provided for "subscriber" maintenance.

This server application has a matching demonstration client called HNDFILECLIENTLEAN.APP which includes upload and download procedure examples. This server application uses dictionary called HNDFILESERVERLEAN.DCT.

This server is able to provide both compressed and encrypted file movement services without requiring either pre-compression or pre-encryption of the upload or download files. The client determines whether compression and encryption are used and this server is ready to comply to requests of this sort.

This application is entirely template-built and contains almost no hand-embedded code.

NOTE: This is a starting-point server from which to build forward. The "file transfer" portion of the server is already installed.

*** HNDFILSV.APP ***

NAME: HNDFILSV.APP
TITLE: FILE SERVER TYPE 3 [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application is intended as a starting point for developers to build file transfer servers of various sorts that act like FTP servers but utilize the HTTP protocol including encryption and compression.

Several demo client applications are provided to CHT subscribers: HNDFLCLN.APP, HNDLIVUP.APP, HNDFILCL.APP and HNDINSTALLMYFILES.APP. Another client application called HNDHTGT.APP illustrates making on-the-fly file requests from this server using the HNDHTTP and HNDHTTPClient classes.

The server maintains complete control of which files and directories are visible to the client. File transfers are optionally compressed and/or encrypted at the request of the client application.

This is an "Authenticating" server. The client logs in by providing "LastName", "LoginID" and "EmailAddress". The server, unless reconfigured uses a member file called HNDFILSVMEM.TPS located in the \HNDAPPS\HNDFILSV\ directory. New members may be added as required to customize the users list.

This application is entirely template-built and contains almost no hand-embedded code.

NOTE: This is a starting-point server from which to build forward. The "file transfer" portion of the server is already installed and working.

*** HNDFILSVSPECIAL.APP ***

NAME: HNDFILSVSPECIAL.APP
TITLE: FILE SERVER-S TYPE 3 [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

These extension/control templates provide TCP/IP connectivity with a browser or client application. In this particular example, application functionality has been limited to acting as an HTTP file transfer server. It requires a file transfer client to interact with it.

This "special" version of the HNDFILSV.APP server is dedicated to working with CHT demo app HNDINSTALLMYFILES.APP, which is a basic file client aimed at getting you to across-the-web app installation.

The server maintains complete control of which files and directories are visible to the client. File transfers are optionally compressed and/or encrypted at the request of the client application.

This application is entirely template-built and contains almost no hand-embedded code.

*** HNDMTSSV.APP ***

NAME: HNDMTSSV.APP
TITLE: CLIENT SERVER TYPE 3 [V10.19.02]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This server application requires a matching demonstration client called HNDMTSCL.APP and a dictionary called HNDMTSSV.DCT.

The server provides both compressed and encrypted data streams consisting of browse data, form data, report and process data based on interactive queries placed from the client side targeted at pre-determined data views (single or multi-table) provided in the server. This application is entirely template-built and contains almost no hand-embedded code.

NOTE: This server is configured for a two-piece login. It omits the Email Address Key on the BrowserServerInitialization controls template. In a two-piece login only the Login ID and Last Name (in this case) are used. This is a client server for reading and posting remote messages -- from which to both learn and with which to build forward. The back end views portions of the server are already installed and working.

*** HNDSVLV.APP ***

NAME: HNDSVLV.APP
TITLE: HTTP SERVER TYPE 3 [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 3
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Several demo client applications are provided to subscribers: HNDFLCLN.APP, HNDLIVUP.APP and HNDFILCL.APP. Another client application called HNDHTGT.APP illustrates making on-the-fly file requests from this server using the HNDHTTP and HNDHTTPClient classes.

The server maintains complete control of which files and directories are visible to the client. File transfers are optionally compressed and/or encrypted at the request of the client application.

This application is entirely template-built and contains almost no hand-embedded code. It is based on a dictionary called HNDFILSV.DCT.

NOTE: This is a starting-point server from which to build forward. The "file transfer" portion of the server is already installed.

**CATEGORY:
HTTP SERVER TYPE 4**

*** HNDSLFSV.APP ***

NAME: HNDSLFSV.APP
TITLE: HTTP SERVER TYPE 4 [V10.19.07]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDSLFSV.APP is a TYPE 4 CHT server, which means it serves files. It is our default web page server. Web pages and images, are after all, files. Normal HTTP page views from a browser are public and do not require password enabled access privileges. File transactions into and out of server-visible directories are security-controlled via a 2-part server login.

HNDSLFSV.AP can securely transact file transfers with several CHT file client applications. All secure clients require this server to set up a single, 2-part server-level login (username + password), to control file upload/download permissions.

HNDSLFCMD.APP is a CHT "Batch-Bot" upload/download application, enabling transfers into and out of the server root and any server subdirectory below the server root. "Batch-Bot" style applications may be used from the DOS command line, from DOS batch scripts, and, naturally, inside Clarion applications where they are known as "CHT Snap-Ins".

HNDSLFLCL.APP is a simple, example client single-file uploader, downloader illustrating four different client-side upload/download methods.

CHTSNAPGET.EXE is a pre-compiled C# utility application included in the CHT Toolkit. It may be used to securely download files from this server when a server username:password combination is configured.

This HNDSLFSV.APP is entirely template-built and contains no significant hand-embedded code.

*** HNDSLFSVFX.APP ***

NAME: HNDSLFSVFX.APP
TITLE: HTTP SERVER - TYPE 4 [V10.19.08]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDSLFSVFX.APP is a TYPE 4 CHT server, which means it serves files. It is our default web page server. Web pages and images, are after all, files. Normal HTTP page views from a browser are public and do not require password enabled access privileges. File transactions into and out of server-visible directories are security-controlled via a 2-part server login.

HNDSLFSV.AP can securely transact file transfers with several CHT file client applications. All secure clients require this server to set up a single, 2-part server-level login (username + password), to control file upload/download permissions.

HNDSLFCMD.APP is a CHT "Batch-Bot" upload/download application, enabling transfers into and out of the server root and any server subdirectory below the server root. "Batch-Bot" style applications may be used from the DOS command line, from DOS batch scripts, and, naturally, inside Clarion applications where they are known as "CHT Snap-Ins".

HNDSLFLCL.APP is a simple, example client single-file uploader, downloader illustrating four different client-side upload/download methods.

This application is entirely template-built and contains almost no hand-embedded code.

*** HNDSLFSVXSA.APP ***

NAME: HNDSLFSVXSA.APP
TITLE: HTTP SERVER TYPE 4 [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This uses a server-side configured 2-part username password setup (standard XSA) using CHT's HNDCONFIG class. Any client requesting an .HZO installation container is forced, via server function GetGetFileApproval() to present the correct username/password string.

The installation clients accessing this server must have pre-knowledge of the server's currently configured

username/password in order to obtain installation containers. This username/password information is not exchanged between the server and the client or vice versa. The installation client either knows the login information ahead of time, and gains access or it is prohibited from obtaining installation containers.

This application is primarily template-built and contains almost no hand-embedded code.

This server and its example client installer HNDLFSVXSAINSTALL.APP, an XSA installer, in combination with CHT's container builder app, HNDSETUP.APP combine to form a very flexible and secure, across-the-web, software, and/or document delivery system to Windows-based computers.

*** HNDLFSVXSA2PART_100_82.APP ***

NAME: HNDLFSVXSA2PART_100_82.APP
TITLE: HTTP SERVER TYPE 4 [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This uses a server-side configured 2-part username/password setup (standard XSA) which can be configured from the server menu via menu "Set uName and Pwd" menu.

This application uses a dictionary that defines an SQLite "Customer" file containing customer status information which can, depending on status field contents, allow or disallow an installation call from the customer's installation client.

This application is primarily template-built and contains almost no hand-embedded code.

This server and its example client installer HNDLFSVXSAINSTALL3PART.APP, an XSA-3 installer, in combination with CHT's container builder app, HNDSETUP.APP combine to form a very flexible and secure, across-the-web, software, and/or document delivery system to Windows-based computers.

This server also utilizes a user/customer maintenance application called HNDLFSVCUS.APP which it calls via RUN() under the "Customer Maintenance Utility" menu. The path to this executable may need to be adjusted in the embed area where it is called. The customer maintenance app shares the HNDLFSV.DCT with this server application.

*** HNDLFSVXSA2PART_100_88.APP ***

NAME: HNDLFSVXSA2PART_100_88.APP
TITLE: HTTP SERVER TYPE 4 [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This uses a server-side configured 2-part username password setup (standard XSA) to authenticate requests from its various clients.

This application uses a dictionary that defines an SQLite "Customer" file containing customer (bogus) status information which can be polled by customer email address to provide (bogus) receipt information when requested by HNDXML2HTMLGEN.EXE running any of the "receipt" templates: 000chtreceipt.xml and 000chtrenewalreminder.xml (and others).

This application is primarily template-built and contains almost no hand-embedded code.

This server also utilizes a user/customer maintenance application called HNDLFSVCUS.APP which it calls via RUN() under the "Customer Maintenance Utility" menu. The path to this executable may need to be adjusted in the embed area where it is called. The customer maintenance app shares the HNDLFSV.DCT with this server application.

*** HNDLFSVXSA3PART.APP ***

NAME: HNDLFSVXSA3PART.APP
TITLE: HTTP SERVER TYPE 4 [V10.18.12]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 4

CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This uses a server-side configured 2-part username password setup (standard XSA) as well as a 3-part email,loginid,username setup (XSA-3) looked up in a user (customer, subscriber) file.

Which of these two security systems is applied by the server depends on the authorization header applied to a file request by the CHT XSA client application.

This application uses a dictionary that defines an SQLite "Customer" file containing customer status information which can depending on status field contents allow or disallow an installation call from the customer's installation client.

This application is primarily template-built and contains almost no hand-embedded code.

This server and its example client installer:
HNDSETUPMKR_XSA3_LOOKUP.APP is an XSA-3 (3-part login) installer, in combination with CHT's container builder app, HNDSETUP.APP combine to form a very flexible and secure, across-the-web, software, and/or document delivery system to Windows-based computers.

Another example client installer called:
HNDSETUPMKR_XSA3_HC.APP, is an XSA-3 (3-part login) installer, with the installer login hard-coded to a specific user in the data base (see template interface).

This server also utilizes a user/customer maintenance application called HNDSLFVSVUS.APP which it calls via RUN() under the "Customer Maintenance Utility" menu. The path to this executable may need to be adjusted in the embed area where it is called. The customer maintenance app shares the HNDSLFVSV.DCT with this server application.

*** HNDTEAMDOCSEVER.APP ***

NAME: HNDTEAMDOCSEVER.APP
TITLE: HTTP SERVER TYPE 4 [V10.19.08]
DATE: 8/24/2019
CATEGORY: HTTP SERVER TYPE 4
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

For developers working in a team, especially teams with remote members, it's useful for members to know which versions of the applications for which they are responsible have been centrally compiled for QA. With each re-build, using CHT Compile Manager, a set of HTML pages is generated into this server's "run" directory, detailing the applications with a variety of information from version number, to procedure docs, to templates in use.

Note that the base page required by this server is docindex.html. CHT compile manager generates a matching index page with this name.

This application is entirely template-built and contains almost no hand-embedded code.

CATEGORY:
INTERNET - CONNECTIVITY

*** HNDDUN.APP ***

NAME: HNDDUN.APP
TITLE: CHT HNDIDial Class Demonstration [V10.18.12.0202]
DATE: 8/24/2019
CATEGORY: Internet - Connectivity
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDIDial can be used alone, as it is here, or in combination with other classes. The CHT tool kit classes, HNDIMail, HNDInetFTP and HNDHttp automatically derive HNDIDial for you in order to handle connectivity issues.

See the header file HNDIDIAL.INC for a list of properties and methods available.

CATEGORY: INTERNET - HTTP CLIENTS

*** HNDCLIENTCLEAN.APP ***

NAME: HNDCLIENTCLEAN.APP
TITLE: Web Data Client Lean [V10.18.07.0259]
DATE: 8/24/2019
CATEGORY: Internet - HTTP Clients
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application dictionary is HNDCLIENTSVLEAN.DCT.

The server, HNDCLIENTSVLEAN.APP, is an example Client Server with no data views, so it is expected that the developer will use that server (saved under a different name) to act as the basis for his Client server views. This "lean" version server has had all non-essential components removed and focusses entirely on the CHT templates required to build a CHT Web Server.

Once that server contains at least one completed data view, then this application (saved under a different name) can begin to act as the basis for an HTTP client that connects to your new server and displays its views.

This application is entirely template-built and jumpstart-built and contains almost no hand-embedded code.

*** HNDGOOGLECALENDAREXPLORER.APP ***

NAME: HNDGOOGLECALENDAREXPLORER.APP
TITLE: Eds Easy Google Calendar Explorer [V10.18.12.0160]
DATE: 8/24/2019
CATEGORY: Internet - HTTP Clients
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application will ask you for your login credentials if they have not already been established by this application. Once entered, it will save them to the registry. These can always be changed using the "Configure Google Credentials" button provided on the toolbar.

HNDGOOGLECALENDAREXPLORER.APP applies primarily a CHT template called HandyComInternetExplorerControl. The template has been configured with specific web form settings that allow it to insert information directly into the web page variables having to do with Google Calendar Login.

The application communicates with the Google Calendar login page's DOM (Document Object Model) using methods provided in CHT class, HNDInternetExplorer.

This same principle can be used to interact with almost any web page by writing to it's form variables from inside the Clarion application hosting the Internet Explorer control.

Any questions, feel free to ask: news.cwhandy.com

*** HNDHTGT.APP ***

NAME: HNDHTGT.APP
TITLE: CHT Client Classes Demo [V10.18.08]
DATE: 8/24/2019
CATEGORY: Internet - HTTP Clients
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application illustrates how to send a variety of requests to a server. It is preconfigured to interact with demo servers HNDFILSV.APP and HNDSLFSV.APP.

The HTTP_CHTSecureFileOperations() procedure can securely upload files, download files, delete files and check file existence against a CHT Browser Server application configured to accept these operations.

At the server side (see HNDFILSV.APP) you have full control over which operations are allowed.

The HTTP_CHTServerInsecureFileOperations() procedure illustrates sending User Custom Requests (UCR\$) to HNDSLFSV.APP. The two example commands provided here, demonstrate remotely running threaded example procedures in the server.

At the server side (see HNDSLFSV.APP) any number of threaded procedures may be inserted, each with it's own command and parameter list, by following the example embeds in Server.ProcessUserCustomRequest().

Any questions, feel free to ask: news.cwhandy.com

*** HNDLIVUP.APP ***

NAME: HNDLIVUP.APP
TITLE: CHT Live-Update Client Demo [V10.18.04.0217]
DATE: 8/24/2019
CATEGORY: Internet - HTTP Clients
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application acts as secure file transfer client in combination with a secure file transfer server also built with the CHT tool kit. Server examples provided are called HNDFILSV.APP and HNDSVLV.APP.

A variation of this application was used to provide "Live-Update" services for your Clarion Handy Tools tool kit before we introduced WEBUPDATER.

This demo client can easily be configured to your own purposes to provide product updates or to service a variety of file transfer requirements for which you might otherwise be inclined use less secure transfer technologies such as FTP.

CATEGORY: INTERNET - JSON

*** HNDCHATCLNT.APP ***

NAME: HNDCHATCLNT.APP
TITLE: CHT Chat Client [V10.18.08]
DATE: 8/24/2019
CATEGORY: Internet - JSON
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application uses CHT-original classes called HND SOCKET and HNDJSON. It has been built to contrast an SV example application called QuickChat.APP which uses SV-provided socket and JSON data transfer classes.

We're using this application for study purposes and to act as a robustness testing platform for Socket connection and JSON classes of our own design. We found after considerable study that SV's JSON and SOCKET classes just weren't up to CHT standards in flexibility and robustness. So we built our own from scratch.

We also separated the server and client functionalities into two applications. This way, one centrally placed, duty server can handle hundreds of clients, the way Skype and other chat functionalities do it.

Any questions, feel free to ask: news.cwhandy.com

*** HNDCHATSVR.APP ***

NAME: HNDCHATSVR.APP
TITLE: CHT Chat Server 74:8088 [V10.18.08.1375]
DATE: 8/24/2019
CATEGORY: Internet - JSON
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It uses CHT-original classes called HND SOCKET and HNDJSON. It has been built to contrast an SV example application called QuickChat.APP which uses SV-provided socket and JSON data transfer classes.

We're using this application for study purposes and to act as a robustness testing platform for Socket connection and JSON classes of our own design. We found after considerable study that SV's JSON and SOCKET classes just weren't up to CHT standards in flexibility and robustness. So we built our own from scratch.

We also separated the server and client functionalities into two applications. This way, one centrally placed, duty server can handle hundreds of clients, the way Skype and other chat functionalities do it.

Any questions, feel free to ask: news.cwhandy.com

CATEGORY: MAPI EMAIL

*** HNDMLMAPI.APP ***

NAME: HNDMLMAPI.APP
TITLE: MAPI BASIC EMAIL [V10.18.12.0209]
DATE: 8/24/2019
CATEGORY: MAPI EMAIL
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates primarily, building MAPI email send procedures using the CHT classes HNDIMAIL.INC and HNDIMAIL.CLW, attached to your application via the EMBEDOBJECT template.

These are BASIC send procedures using a minimum of HNDIMAIL features, designed specifically for easy understanding. More complex mail procedures are available in other demonstration applications.

All procedures in this application are available as JUMP START procedures via the CHTJumpStartProceduresUtility Template. Import any of them into your application as a model from which to work forward. Or simply import any or all of the following procedures into your application directly from this application.

```
** JumpStartMapiSendProcedureNoAttachments()  
** JumpStartMapiSendProcedureWithOneAttachment()  
** JumpStartMapiSendProcedureWithMultipleAttachments()  
** JumpStartMapiSendMessageFromAddressBook()  
** JumpStartMapiInsertMessageWithClientInterface()  
** JumpStartMapiSendMessageWithClientInterface()
```

NOTE: These procedures all require some in-code customization on your part to make them specific to your end use. Study the code and modify or adjust as needed.

CATEGORY: MS OFFICE EXCEL

*** HNDTESTEXCEL.APP ***

NAME: HNDTESTEXCEL.APP
TITLE: CHT Excel Class Example [V10.19.02.0300]
DATE: 8/24/2019
CATEGORY: MS Office Excel
CLASSES: [List App Classes](#)

TEMPLATES: [List App Templates](#)

DESCRIPTION:

The HNDEXCELCLASS class is introduced to the procedure via a template called EmbedHndExcelClassFunctions.

Various controls are provided on the window that call into class functions to illustrate HNDEXCELCLASS features.

Any questions, feel free to ask.

**CATEGORY:
MS OFFICE INTERFACE**

***** HNDWORDINTERFACEDEMO.APP *****

NAME: HNDWORDINTERFACEDEMO.APP
TITLE: HandyWordInterfaceControls Demo [V10.18.09.0154]
DATE: 8/24/2019
CATEGORY: MS Office Interface
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The HNDOfficeWord class is introduced to the procedure via a template called EmbedHndWordClassFunctions. Then a second template called HandyWordInterfaceControls has been added to the main window to illustrate a variety of functions available.

This application is a work in progress intended for expansion as CHT Word templates are added and improved.

Any questions, feel free to ask.

**CATEGORY:
MS OFFICE OUTLOOK**

***** HNDOUTLKDEMO.APP *****

NAME: HNDOUTLKDEMO.APP
TITLE: HNDOfficeOutLook Class Demo [V10.18.12.0162]
DATE: 8/24/2019
CATEGORY: MS Office Outlook
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The HNDOfficeOutlook class is introduced to the procedure via a template called EmbedHndOutlookClassFunctions. Then a second template called HandyOutlookInterfaceControls has been added to the main window to illustrate a variety of functions available.

This application is a work in progress intended for expansion as CHT Outlook templates are added and improved.

Any questions, feel free to ask: news.cwhandy.com

***** HNDOUTLOOK.APP *****

NAME: HNDOUTLOOK.APP
TITLE: MS Outlook View Control [V10.18.12.0157]
DATE: 8/24/2019
CATEGORY: MS Office Outlook
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

A CHT control template called HandyCOMOutLookControl helps you drop this control onto any application window and instantiate a Clarion-style interface to the COM IViewCtl control.

With the HndComOutlookControl template and it's underlying CHT OOP classes you can have a fully interactive MS Outlook application running in your Clarion Window, including such Outlook features as Inbox, OutBox, Sent Items, Calendar, Contacts, Journal, Notes, Tasks and Drafts.

Any questions, feel free to ask: news.cwhandy.com

**CATEGORY:
MS OFFICE WORD******* HNDWORDDEMO.APP *****

NAME: HNDWORDDEMO.APP
TITLE: MS Word Interface Demo [V10.18.09.0153]
DATE: 8/24/2019
CATEGORY: MS Office Word
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The HNDMSWORD class is introduced to the procedure via a template called EmbedWordFunctions.

This application, in an of itself uses a window with no controls that minimizes itself and drops an icon on your tool bar from where several command menu items illustrate programmatic control over MS Word.

Any questions, feel free to ask.

**CATEGORY:
POP-UP MENUS******* HNDPOPUPMENUDEMO.APP *****

NAME: HNDPOPUPMENUDEMO.APP
TITLE: Handy Popup Menu Demo [V10.18.12.0180]
DATE: 8/24/2019
CATEGORY: Pop-Up Menus
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Developers can drop one or more instances of this template on any Clarion Window procedure. It's easy to build colorful popup menus in seconds. While all of the work can be done on the template interface, optional embed points are provided for developers advanced enough to embed popup menu execution code.

This application illustrates 5 instances of the template starting a variety of CHT demo applications.

**CATEGORY:
PROCESSES******* HNDSQL5.APP *****

NAME: HNDSQL5.APP
TITLE: HandyQueryProcessLimiterQBControl [V10.18.09.0193]
DATE: 8/24/2019
CATEGORY: Processes
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It illustrates these concepts:

- *** A CHT HandyQueryProcessLimiter Template
- *** A CHT HandyQueryProcessLimiterQBControl Template

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

**CATEGORY:
QUERYING, LOCATING AND FILTERING******* HNDSQLPOPFVHMB.APP *****

NAME: HNDSQLPOPFVHMB.APP
TITLE: Demonstrate PopFavoriteQueries_HMB (SQL) [V10.17.12]
DATE: 8/24/2019
CATEGORY: Querying, Locating And Filtering
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It illustrates these concepts:

- *** A Explorerbrowse implementation running on SQL Data Tables.
- *** CHT Query template PopFavoriteQueries_EXPB.

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

**CATEGORY:
REPORTS******* HNDSQL6.APP *****

NAME: HNDSQL6.APP
TITLE: HandyQueryReportLimiterQBControl [V10.18.09.0188]
DATE: 8/24/2019
CATEGORY: Reports
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It illustrates these concepts:

- *** A CHT HandyQueryReportLimiter Template
- *** A CHT HandyQueryReportLimiterQBControl Template

This demonstration application requires the sample data base provided with MICROSOFT ACCESS, called Northwind. The application will find the data base and register it with ODBC if not already registered.

**CATEGORY:
SMS MESSAGES******* HNDSMSMESSAGE.APP *****

NAME: HNDSMSMESSAGE.APP
TITLE: CHT SMS Messaging Utility [V10.18.11.0414]
DATE: 8/24/2019
CATEGORY: SMS Messages
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Further it is dependent on correct configuration of a valid SMTP server or SMS gateway setup with UserName, Password, Email/SMS Server and Email/SMS Return Address. Our template and SMS classes provide a configuration function with user-controllable registry location settings.

The SMS template illustrated is EMBEDSMSFUNCTIONS. The CHT class underlying this template is HNDSMS located in files HNDSMS.INC and HNDSMS.CLW.

Please read and understand the proper formation of SMS addresses before experimenting, using this URL: (https://en.wikipedia.org/wiki/SMS_gateway).

CATEGORY: SNAP-IN C#

*** CHTSNAP2PDF.EXE ***

NAME: CHTSNAP2PDF.EXE
TITLE: CHT Snap Edit CMD Utility -- Version: 23.19.09.01
DATE: 8/24/2019
CATEGORY: SNAP-IN C#

DESCRIPTION:

CHTSNAP2PDF.EXE (AKA: CHT Snap To PDF), is a command-line translation tool that is able to translate well-formed HTML 5 documents into exact-copy PDF documents that include full use of correctly formatted internal and external links.

CHTSNAP2PDF.EXE requires no pre-configuration and has no DLL dependencies in this version or later. It can be run from the DOS prompt, from a standard Windows short cut or from inside a Clarion application. A number of CHT demo and utility applications, for example *HNDDOCUMENTBUILDER* make extensive use of this document translation application.

CHTSNAP2PDF.EXE can be incorporated into your application from the CHT Snap-In interface where you can find a "CHTSnap2PDF" option. Our template sets an IDE directive to copy the CHTSNAP2PDF.EXE into your application's compile directory. The template also enables a CHT Class called HNDSNAPCALL which contains functions for calling the editor from your application.

For DEMO application which incorporate CHTSNAP2PDF.EXE as described above, see HNDDOCUMENTBUILDER.APP, HNDDPREVIEWER.APP and HNDTPXHTNEXT.APP. These applications are described here: [HNDDOCUMENTBUILDER.APP](#).

Contact CHT at: support@cwandy.ca for price and availability of C# source code information.

Gus M. Creces
The Clarion Handy Tools Page
<http://www.cwandy.ca>
<http://news.cwandy.ca>

*** CHTSNAPDISK.EXE ***

NAME: CHTSNAPDISK.EXE
TITLE: CHT Snap Disk CMD Utility -- Version: 23.19.08.02
DATE: 8/24/2019
CATEGORY: SNAP-IN C#

DESCRIPTION:

CHTSNAPDISK.EXE (AKA: CHT Snap Disk) is a command-line disk information tool included with the CHT toolkit. It may be used directly from the DOS prompt or it may be incorporated into your application via CHT templates.

CHT Snap Disk does not require any kind of pre-configuration involving a configuration dialog or maintaining registry entries

These DOS command-line inputs are required to use CHT Snap Disk.

(©) The Clarion Handy Tools Page
CHT Snap Disk -- Version: 23.19.08.03

These separate DOS command-line inputs are required to obtain disk or processor information.

To display CHTSNAPDISK.EXE instructions to console:
chtnapdisk --help, -help, or help

To display full disk information to console:
chtnapdisk --info, -info, or info

To generate full disk information to file: chtsnapdisk.txt.
chtnapdisk --list, -list or list

To generate full disk information to file: chtsnapdisk.txt,
and open with chtsnapedit.exe:
chtnapdisk --listedit, -listedit or listedit

To display processor unique ID to console:
chtnapdisk --processor, -processor or processor

To change the LABEL of the designated drive number,
Drive 0 is assumed if [OPT] "DrvNUM" parameter is omitted:
chtnapdisk --label, -label or label "Your label inside quotes" [DrvNum]

To display free space information of all drives to console:
chtnapdisk --free, -free or free

CHT Snap Disk -- Version: 23.19.08.03

Contact CHT at: support@cwandy.ca for price and availability of C# source code information.

Gus M. Creces
The Clarion Handy Tools Page
<http://www.cwandy.ca>
<http://news.cwandy.ca>

*** CHTSNAPEDIT.EXE ***

NAME: CHTSNAPEDIT.EXE
TITLE: CHT Snap Edit CMD Utility -- Version: 23.19.08.01
DATE: 8/24/2019
CATEGORY: SNAP-IN C#

DESCRIPTION:

CHT Snap Edit requires no pre-configuration and has no DLL dependencies in this version or later. It can be run from the DOS prompt, from a standard Windows short cut or from inside a Clarion application. A number of CHT demo and utility applications, for example *HNDDOCUMENTBUILDER* make extensive use of this editor application.

This editor is branched from the same C# source code from which the Clarion IDE code editor was originally built. In fact, if you view Clarion source .CLW files from this editor, editing will feel familiar. The hot keys provided, however, are optimized for editing raw HTML and XML files.

Hot keys are provided for most of standard mark-up operations such as **BOLD**, *ITALIC* Paragraph, Image Links, Web Anchors and numerous others.

CHTSNAPEDIT.EXE can be incorporated into your application from the CHT Snap-In interface where you can find a "CHT Snap Edit" option. Our template sets an IDE directive to copy the CHTSNAPEDIT.EXE into your application's compile directory. The template also enables a CHT Class called HNDSNAPCALL which contains functions for calling the editor from your application.

Two procedure-local templates called SNAPEDITSTARTBUTTON, RUNCHTSNAPEDIT provide an easy way to incorporate various text, HTML or XML editing tasks into a Clarion Window procedure.

For DEMO application which incorporate CHTSNAPEDIT.EXE as described above, see HNDDOCUMENTBUILDER.APP, HNDBULKMAILBATCHER.APP and HNDBULKNETMAILPROMO.APP. These applications are described here: [HNDDOCUMENTBUILDER.APP](#).

Contact CHT at: support@cwandy.ca for price and availability of C# source code information.

Gus M. Creces
The Clarion Handy Tools Page
<http://www.cwandy.ca>
<http://news.cwandy.ca>

*** CHTSNAPGET.EXE ***

NAME: CHTSNAPGET.EXE
TITLE: CHT Snap Get CMD Utility -- Version: 23.19.08.01
DATE: 8/24/2019
CATEGORY: SNAP-IN C#

DESCRIPTION:

CHTSNAPGET.EXE is added as a component to your Clarion application, from the CHT's global "Application Snap-Ins" template. A procedure-local template called SNAPGETSTARTBUTTON provides an easy way to configure the application two different ways from a Clarion window procedure and to trigger download of a file named in a procedure entry field.

CHTSNAPGET.EXE can also be used directly from the DOS command-line or in a DOS BATCH script.

For a DEMO application, see HNDTESTCHTSNAPGET.APP. That application is described here: [DOCS FOR HNDTESTCHTSNAPGET.APP](#).

These DOS command-line inputs are required to fetch a remote file:

** CHTSNAPGET.EXE ** (©) The Clarion Handy Tools Page

These DOS command-line inputs are required to fetch a remote file:
chtsnapget --fetch FILENAME

chtsnapget --fetchuz FILENAME (to unzip on receipt of .zip file)

FILENAME must be the raw name, no path, e.g. "myfile.pdf".
Remote subdirectories are configured in the setup dialog.
Wild card file names are not permitted. URL processing is stopped.

This application must be configured before use.
Configure manually via the command line, or by programmatically completing registry values with Clarion HNDSnapCall.CHTSnapGetConfig() class function (HNDSNAPCALL.INC/.CLW).
To configure from the DOS command line, type: "chtsnapget --config" and complete the provided dialog values.

** CHTSNAPGET.EXE ** Version: 23.19.08.01 CHT-XSA **

Contact CHT at: support@cwandy.ca for price and availability.

Gus M. Creces
The Clarion Handy Tools Page
<http://www.cwandy.ca>
<http://news.cwandy.ca>

***** CHTSNAPMAPI.EXE *****

NAME: CHTSNAPMAPI.EXE
TITLE: CHT Snap MAPI CMD Utility -- Version: 23.19.08.01
DATE: 8/24/2019
CATEGORY: SNAP-IN C#

DESCRIPTION:

CHTSNAPMAPI.EXE (AKA: CHT Snap MAPI) is able to send emails using the system's default MAPI client if such a client is already properly configured. Emails sent this way may be found, after sending, in the "Sent" mail repository of the default client as if they were sent manually from the client interface.

CHT Snap Mapi does not require any kind of pre-configuration involving a configuration dialog or maintaining registry entries. All information required for any individual email-send task can be passed on the application command line, since the installed email client is already pre-configured.

CHTSNAPMAPI.EXE is added as a component to your Clarion application, from the CHT's global "Application Snap-Ins" template. A procedure-local template called SNAPMAPISTARTBUTTON provides an easy way send MAPI email from a Clarion window procedure.

For a DEMO application, see HNDTESTCHTSNAPTOOLS.APP. That application is described here: [DOCS FOR HNDTESTCHTSNAPTOOLS.APP](#).

CHTSNAPMAPI.EXE accepts DOS command-line inputs in order to send text or HTML emails.

Three or four command-line inputs are required:

** CHTSNAPMAPI.EXE ** (© The Clarion Handy Tools Page - www.cwhandy.ca)
-- Three or four command-line inputs are required:
-- CHTSNAPMAPI "TO" "SUBJECT" "MESG.TXT OR MESG.HTML" ["ATTACH.FILE"]
-- Inputs with spaces or extended characters must be surrounded by double quotes.
-- Otherwise, inputs may be provided without surrounding double quotes.
-- ATTACH.FILE is optional. This option represents optional attachment file(s).
-- Mail results are written to standard output, namely the console.
-- MESG.TXT, MESG.HTML and ATTACH.FILE may include full paths to different folder locations.
-- Multiple TO addresses may be separated by either semi-colons or commas.
-- Multiple attachments may be separated by either semi-colons or commas.
** CHTSNAPMAPI.EXE ** Version: [23.19.08.01] **

Contact CHT at: support@cwhandy.ca for price and availability of C# source code information.

Gus M. Creces
The Clarion Handy Tools Page
<http://www.cwhandy.ca>
<http://news.cwhandy.ca>

*** CHTSNAPSEND.EXE ***

NAME: CHTSNAPSEND.EXE
TITLE: CHT Snap SEND CMD Utility -- Version: 22.18.01.00
DATE: 8/24/2019
CATEGORY: SNAP-IN C#

DESCRIPTION:

CHTSNAPSEND.EXE source code, because it is not in Clarion app format, is available from CHT via special request in the form of a C# wizard. Once imported as a project into Visual Studio, the developer can customize the application to his own specifications should that ever be necessary.

CHTSNAPSEND.EXE (AKA: CHT Snap Send), is an error-reporting tool which integrates, with your Clarion application via CHT templates. It is able to provide screen-snaps of error-causing procedure windows as well as lower-level error codes and error messages issued by the application when an error is encountered. CHTSNAPSEND greatly increases the accuracy of error reporting from applications as it eliminates most of the error-reporting steps that end-users are likely to have difficulties with.

CHT Snap Send email settings can be configured manually using its internal setup dialog, or by leveraging a Clarion application to write the necessary registry entries from our Snap-Ins template interface.

This application does not take command-line inputs. It is leveraged by CHT's Snap-In template interface provided globally on the CHTSNAPGET tab and procedure-locally on a procedure tab that appears when the global CHTSNAPGET tab is enabled.

The "CHT Snapsend" template element, when enabled, writes global SnapSend configuration entries to the Windows Registry in order to customize some of the actions and messages that CHT SnapSend performs and displays, while taking application screen snaps. Further, this template causes the auto-population of multiple instances of a procedure-level template (one per window procedure) called "ProcedureSnapins(HndTools)". ProcedureSnapins(HndTools) instances are de-populated when ApplicationSnapins(HndTools) is removed. See the docs for ProcedureSnapIns to discover the procedure-specific configurations that are possible with that auto-populated template.

EmbedSnapCall: This template may be used to attach to your application, at the procedure level, a CHT Code module containing a single class called HNDSnapCall. This extension creates a single, local instance of the HNDSnapCall Class which is used to call a variety of "CHT SNAP-IN" executable tools in a manner that resembles standard DLL calls, using parameter passing on the command line.

ProcedureSnapIns: On a procedure-by-procedure basis, this template allows the developer to optionally expose the state of procedure-local and global variables while the procedure is running. These variables are added into a text file that is sent via email along with the procedure-window snapshot taken by CHT SnapSend. This template is also the parent of a button template called "HandyWindowsStopStartSnapSendButton(HndTools)". That means the button template does not appear in the IDE's control templates list until ApplicationSnapins' "Enable?" switch is checked.

Contact CHT at: support@cwhandy.ca for price and availability information.

Gus M. Creces
The Clarion Handy Tools Page
<http://www.cwhandy.ca>
<http://news.cwhandy.ca>

*** CHTSNAPSMTP.EXE ***

NAME: CHTSNAPSMTP.EXE
TITLE: CHT Snap SMTP CMD Utility -- Version: 23.19.08.04
DATE: 8/24/2019
CATEGORY: SNAP-IN C#

DESCRIPTION:

CHTSNAPSMTP.EXE (AKA: CHT Snap SMTP), is able to send email via SMTP protocol from any Windows system with an internet connection and a viable SMTP server. Even web-based email providers such as HOTMAIL, GMAIL and YAHOO may be leveraged by CHTSNAPSMTP since they provide also for unique SMTP setups. CHTSNAPSMTP can also send secure, encrypted email assuming that the SMTP server being accessed provides for it.

CHT Snap Get can be configured manually from the command line, or by leveraging a Clarion application to write the necessary registry entries. Once configured, to a specific SMTP server and account emails can be sent from the DOS command line.

CHTSNAPSMTP.EXE is added as a component to your Clarion application, from the CHT's global "Application Snap-Ins" template. A procedure-local template called SNAPSMTPSTARTBUTTON provides an easy way send SMTP email from a Clarion window procedure.

For a DEMO application, see HNDTESTCHTSNAPTOOLS.APP. That application is described here: [DOCS FOR HNDTESTCHTSNAPTOOLS.APP](#).

These DOS command-line inputs are required to send text or HTML emails:

Three or four DOS command-line inputs are required in order to send text or html mail with optional attachments:

```
** CHTSNAPSMTP.EXE ** (©) 2019 The Clarion Handy Tools Page
-- Three or four DOS command-line inputs are required
-- in order to send text or html mail with optional attachments:
-- CHTSNAPSMTP "TO" "SUBJECT" "MSG.TXT" OR "MSG.HTML" ["ATTACH.FILE"]
-- Inputs with spaces or extended characters use double quotes.
-- Otherwise, inputs may be provided without surrounding double quotes.
-- Individual inputs must be separated by one or more spaces.
-- ATTACH.FILE is optional. It represents optional attachment file(s).
-- CHT Snap SMTP must be configured before mail can be sent.
-- Type "CHTSNAPSMTP --config" at the DOS prompt to configure.
-- Mail results are written to the console and to chtsnapsmtp.log.
-- MSG.TXT or MSG.HTML and ATTACH.FILE [OPT] may include full paths to folders.
-- Multiple TO addresses may be separated with either semi-colons or commas.
-- Multiple attachments may be separated with either semi-colons or commas.
-- Issue "--rcpt" alone on command line prior to sending for read-receipt.
-- Issue "--norcpt" prior to sending to cancel read-receipt setting.
```

```
** CHTSNAPSMTP.EXE ** Version: [23.19.08.04] **
```

Contact CHT at: support@cwhandy.ca for price and availability of C# source code information.

Gus M. Creces
The Clarion Handy Tools Page
<http://www.cwhandy.ca>
<http://news.cwhandy.ca>

*** CHTSNAPZIP.EXE ***

NAME: CHTSNAPZIP.EXE
TITLE: CHT Snap Zip CMD Utility -- Version: 23.19.08.03
DATE: 8/24/2019
CATEGORY: SNAP-IN C#

DESCRIPTION:

CHTSNAPZIP.EXE (AKA: CHT Snap Zip) is a command-line zip/unzip tool included with the CHT toolkit. It may be used directly from the DOS prompt or it may be incorporated into your application via CHT templates.

CHT Snap Zip does not require any kind of pre-configuration involving a configuration dialog or maintaining registry entries. All information required for any individual "zipping" task can be passed on the application command line.

CHTSNAPZIP.EXE is added as a component to your Clarion application, from the CHT's global "Application

Snap-Ins" template. Three procedure-local templates called SNAPZIPSTARTBUTTON, SNAPUNZIPSTARTBUTTON and SNAPZIPDIRSTARTBUTTON, provides an easy way to incorporate various zipping tasks into a Clarion Window procedure.

For a DEMO application, see HNDTESTCHTSNAPZIP.APP. That application is described here: [DOCS FOR HNDTESTCHTSNAPZIP.APP](#).

These DOS command-line inputs are required to zip files or directories.

```
** CHTSNAPZIP.EXE - A CHT Snap Utility ** Version: [23.19.08.05]
** (©) 2019 The Clarion Handy Tools Page - www.cwhandy.ca
```

```
-- The ZR parameter recurses subdirectories and accepts 3 inputs.
-- It produces a zip with folder names included inside the zip.
-- The second input may include a target path for placement of the output zip file.
-- When no zip target path is given, the zip file is created in the same directory
-- as the CHTSNAPZIP.EXE executable.
-- The "rootdir" parameter used without a mask must end with a backslash.
-- [mask] is optional after the backslash [e.g. \hnd*.tps] to filter files.
```

```
chtsnapzip <[\path\]zipfilename>
```

```
-- The ZZ parameter skips subdirectories and accepts 3 inputs.
-- It produces a flat zip without the folder names in the zip.
-- The second input may include a target path for placement of the output zip file.
-- When no zip target path is given, the zip file is created in the same directory
-- as the CHTSNAPZIP.EXE executable.
-- The "sourcedir" parameter used without a mask must end with a backslash.
-- [mask] is optional after the backslash [e.g. \hnd*.tps] to filter files.
```

```
chtsnapzip <[\path\]zipfilename>
```

```
-- The ZL parameter lists zip contents and accepts 2 inputs.
It produces a comma-delimited list of zip files for pickup by calling apps.
```

```
chtsnapzip <[\path\]zipfilename>
```

```
-- The ZF paramter list zip contents and accepts 2 inputs.
It produces a fully formatted list intended for easy reading from the DOS console.
```

```
chtsnapzip <[\path\]zipfilename>
```

```
-- The UZ parameter unzips the zip file contants and accepts 3 inputs.
-- The second input may include a full path to the zip file.
-- When no zip path is given, the zip file is assumed to be located
-- in the same directory as the CHTSNAPZIP.EXE executable.
-- The "targetdir" parameter used without a mask must end with a backslash.
-- [mask] is optional after the backslash [e.g. \hnd*.tps] to filter files.
```

```
chtsnapzip <[\path\]zipfilename>
```

```
** CHTSNAPZIP.EXE - A CHT Snap Utility ** Version: [23.19.08.05]
```

Contact CHT at: support@cwhandy.ca for price and availability of C# source code information.

Gus M. Creces
The Clarion Handy Tools Page
<http://www.cwhandy.ca>
<http://news.cwhandy.ca>

*** HNDCLEAN.EXE ***

NAME: HNDCLEAN.EXE
TITLE: CHT Clarion Gen-Code Cleaner -- Version: 22.18.01.00
DATE: 8/24/2019
CATEGORY: SNAP-IN C#

DESCRIPTION:

HNDCLEAN.EXE source code, because it is not in Clarion app format, is available from CHT via special request in the form of a C# wizard. Once imported as a project into Visual Studio, the developer can customize the application to his own specifications should that ever be necessary.

HNDCLEAN.EXE (AKA: CHT Clarion Gen-Time Code Cleaner), is a replacement for the Clarion IDE's inadequately functioning "Clean Solution" operation. It erases APPNAME*.CLW and APPNAME*.OBJ files,

and more, when configured to match the IDE's .RED file [RELEASE] or [DEBUG] configurations.

HNDCLEAN.EXE can be installed to the IDE's "External Tools" interface under the name "CHT Clarion Gen-Code Cleaner". The "Arguments" field should be completed as explained in our document located at <http://www.cwhandy.ca/howtousehndclean.htm>.

If you've made modifications to your application that requires the application to fully regenerate, with this tool installed, you can erase the .CLW and .OBJ files associated with the application currently loaded. Then by using the IDE's "Build Generate and Make" menu, the entire application is regenerated from scratch, and compiled with all hand-embedded changes and template interface changes being correctly applied.

Contact CHT at: support@cwhandy.ca for price and availability of C# source code information.

Gus M. Creces
The Clarion Handy Tools Page
<http://www.cwhandy.ca>
<http://news.cwhandy.ca>

CATEGORY: SNAP-IN EXAMPLE

*** HNDSLFCMDSNAPTEST.APP ***

NAME: HNDSLFCMDSNAPTEST.APP
TITLE: SNAP-IN EXAMPLE [V10.18.12]
DATE: 8/24/2019
CATEGORY: SNAP-IN EXAMPLE
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application illustrates how to incorporate HNDSLFCMD.EXE into a Clarion application via the global SNAP-IN interface template "CHT ApplicationSnapIns".

HNDSLFCMD.EXE is a "BATCH-BOT" command-line utility to upload and download files to or from a standard, unprotected web url like a generally accessible web page or open file delivery URL. It uses the HTTP protocol.

HNDSLFCMD.EXE also has secure-file access capabilities. This requires user knowledge of the user-name/password combination protecting the file or files being downloaded or uploaded.

*** HNDSNAPTEST.APP ***

NAME: HNDSNAPTEST.APP
TITLE: SNAP-IN ExAMPLE [V10.18.12]
DATE: 8/24/2019
CATEGORY: SNAP-IN EXAMPLE
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application illustrates how to incorporate CHTSNAPGET.EXE into a Clarion application via the global SNAP-IN interface template "CHT ApplicationSnapIns".

CHTSNAPGET.EXE is a command-line utility to download files from a standard, unprotected web url like a generally accessible web page or open file delivery URL. It is able to use standard HTTP:// or HTTPS:// protocols. It is also able to download from CHT-XSA password protected servers.

*** HNDTESTCHTSNAPGET.APP ***

NAME: HNDTESTCHTSNAPGET.APP
TITLE: SNAP-IN EXAMPLE [V10.19.08]
DATE: 8/24/2019
CATEGORY: SNAP-IN EXAMPLE
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application illustrates how to incorporate CHTSNAPGET.EXE into a Clarion application via the global SNAP-IN interface template "CHT ApplicationSnapIns".

CHTSNAPGET.EXE is a command-line utility to download files from a standard, unprotected web url like a generally accessible web page or open file delivery URL. It uses the HTTP protocol.

CHTSNAPGET.EXE also has secure-file access capabilities. This requires user knowledge of the user-name/password combination protecting the file or files being downloaded. This secure mode can optionally use HTTPS protocol or HTTP protocol depending on server configuration.

An optional configuration switch for "CHT Servers" is available to interface with CHT Server only file fetch commands. This mode can be tested with HNDSLFSV.APP or HNDFILSV.APP available also as demo applicaitons in your \HNDAPPS\ directory.

*** HNDTESTCHTSNAPTOOLS.APP ***

NAME: HNDTESTCHTSNAPTOOLS.APP
TITLE: SNAP-IN EXAMPLE [V10.19.08]
DATE: 8/24/2019
CATEGORY: SNAP-IN EXAMPLE
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates the CHT template global template called ApplicationSnapIns.

This global template provides an interface to your application whereby several "Snap-In" components may be incorporated.

This application illustrates CHTSNAPEDIT.EXE, CHTSNAPMAPI.EXE and CHTSNAPSMTP.EXE.

CHT "Snap-Ins" are C# executables (EXE's) which provide various functionalities, which can be used standalone from the DOS prompt or programmatically from a Clarion application. The above "Snap-Ins" provide text editing, MAPI email and SMTP email, respectively.

Procedure-level button control templates are provided for each of the "Snap-In" executables in order to pass information to them and launch them from your application, seamlessly, just like dll-based procedures. Button templates are: HandyWindowsStopStartSnapSendButton, SnapEditStartButton, SnapSMTPStartButton, and SnapMAPIStartButton respectively.

*** HNDTESTCHTSNAPZIP.APP ***

NAME: HNDTESTCHTSNAPZIP.APP
TITLE: CHTSNAPZIP.EXE EXAMPLE [V10.19.08]
DATE: 8/24/2019
CATEGORY: SNAP-IN EXAMPLE
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application illustrates how to incorporate CHTSNAPZIP.EXE into a Clarion application via the global SNAP-IN interface template "CHT ApplicationSnapIns".

CHTSNAPZIP.EXE is a command-line utility that auto-zips a target directory while OPTIONALLY including any sub-directories of the source directory. CHTSNAPZIP.EXE also unzips to a target directory. This application illustrates how to do all of those things from inside a Clarion application using CHT Templates to write the code.

This application presents a SnapZipStartButton template and two input fields, a target zip file name field and a source directory name field. Complete these two fields and click the zip start button.

Other related templates used are SnapZipDirStartButton and SnapUnZipStartButton.

CHTSNAPZIP.EXE warns if the target zip file exists and if the source directory doesn't exist. Messages from CHTSNAPZIP.EXE appear in a window text control to tell you the outcome of pushing the zip button.

*** HNDTESTMAILCDO.APP ***

NAME: HNDTESTMAILCDO.APP
TITLE: SNAP-IN EXAMPLE [V10.18.12]
DATE: 8/24/2019
CATEGORY: SNAP-IN EXAMPLE
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates the CHT template global template called ApplicationSnapIns.

This global template provides an interface to your application whereby any of sixteen (16) "Snap-In" components may be incorporated.

This application illustrates use of CHT Batch-Bot HNDMAILCDO.EXE to send email from a Clarion window procedure.

This specific Batch-Bot is engaged with the procedure window using a control template called SnapHNDMAILCDOStartButton.

Procedure-level button control templates are provided for each of the "Snap-In" executables in order to pass information to them and launch them from your application, seamlessly, just like dll-based procedures.

***** HNDTESTMAILMAPI.APP *****

NAME: HNDTESTMAILMAPI.APP
TITLE: SNAP-IN EXAMPLE [V10.18.12]
DATE: 8/24/2019
CATEGORY: SNAP-IN EXAMPLE
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates the CHT template global template called ApplicationSnapIns.

This global template provides an interface to your application whereby any of sixteen (16) "Snap-In" components may be incorporated.

This application illustrates use of CHT Batch-Bot HNDMAILMAPI.EXE to send email from a Clarion window procedure.

This specific Batch-Bot is engaged with the procedure window using a control template called SnapHNDMAILMAPIStartButton.

Procedure-level button control templates are provided for each of the "Snap-In" executables in order to pass information to them and launch them from your application, seamlessly, just like dll-based procedures.

***** HNDTESTMAILNET.APP *****

NAME: HNDTESTMAILNET.APP
TITLE: SNAP-IN EXAMPLE [V10.18.12]
DATE: 8/24/2019
CATEGORY: SNAP-IN EXAMPLE
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates the CHT template global template called ApplicationSnapIns.

This global template provides an interface to your application whereby any of sixteen (16) "Snap-In" components may be incorporated.

This application illustrates use of CHT Batch-Bot HNDMAILNET.EXE to send email from a Clarion window procedure.

This specific Batch-Bot is engaged with the procedure window using a control template called SnapHNDMAILNETStartButton.

Procedure-level button control templates are provided for each of the "Snap-In" executables in order to pass information to them and launch them from your application, seamlessly, just like dll-based procedures.

*** HNDTESTMAILSMTP.APP ***

NAME: HNDTESTMAILSMTP.APP
TITLE: SNAP-IN EXAMPLE [V10.19.01]
DATE: 8/24/2019
CATEGORY: SNAP-IN EXAMPLE
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application demonstrates the CHT template global template called ApplicationSnapIns.

This global template provides an interface to your application whereby any of sixteen (16) "Snap-In" components may be incorporated.

This application illustrates use of CHT Batch-Bot HNDMAILSMTP.EXE to send email from a Clarion window procedure.

This specific Batch-Bot is engaged with the procedure window using a control template called SnapHNDMAILSMTPStartButton.

Procedure-level button control templates are provided for each of the "Snap-In" executables in order to pass information to them and launch them from your application, seamlessly, just like dll-based procedures.

CATEGORY: SOUND AND VIDEO

*** HNDQMCI.APP ***

NAME: HNDQMCI.APP
TITLE: MCI Quick Player [V10.18.12.0152]
DATE: 8/24/2019
CATEGORY: Sound And Video
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Other significant templates used in this application:

- * HandyFileSelector - select files with advanced querying
- * ListBoxBrowseExtender - list box header click column reorder, record marking and more

Other significant classes used in this application:

- * HNDisk - advanced, low level disk functions
- * HNDFileLoadBrowse - list box record marking, querying, sorting
- * HNDWindow - advanced, low level window functions
- * HNDAnimate - MS Windows animation control

CHT subscribers should study the header file for our HNDAVI class to get a deeper understanding of the properties and methods available.

Any questions, feel free to as.

CATEGORY: SPELLING-CHECKER

*** HNDSPELLCHECKDEMO.APP ***

NAME: HNDSPELLCHECKDEMO.APP
TITLE: Spell-Check Local (ENG) [V10.18.0004]
DATE: 8/24/2019
CATEGORY: Spelling-Checker
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It drops a single "Spell Check" button onto your window to trigger spell-checking in entry and text controls populated on the procedure window. Checking can be triggered for the entire window using the button provided or individual controls can be checked using a configurable hot key.

Further, individual words may be checked by selecting the word and using a configurable hot key. Initially we are providing, on request, lexicons for the following languages: English, Afrikaans, Croatian, French, German, Greek, Italian, Netherlands, Portugese, Russian, Spanish, Swedish, Ukranian.

Since these lexicons are derived from open-source word lists, we will ask developers seeking non-English dictionaries to help us develop a common-words list for their chosen language. A common-words list is a set of the top 25-50 most commonly used words. In English, for example, these are words like "the", "and", "to", and so forth.

As of CHT build 112A1.01 CHT spell checking has been taken out of BETA and placed into RELEASE status. Four dictionaries are presently available: English, French, Afrikaans, and Portugese. Aside from English, we chose to release these languages first because developers contacted us and specifically encouraged us to do so.

If you speak and work in any of the above listed, but not-yet-implemented langauges, feel free to contact us and encourage us to give priority to your language in an upcoming build.

Any questions, feel free to ask.

*** HNDSPELLCHECKGLOBAL.APP ***

NAME: HNDSPELLCHECKGLOBAL.APP
TITLE: CHT Global Spell-Check English [V10.18.05.0178]
DATE: 8/24/2019
CATEGORY: Spelling-Checker
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

It drops a local spell check template called LocalIncludeSpellingCheck onto your window to trigger spell-checking in entry and text controls populated on the procedure window. Checking can be triggered for the entire window using CtrlF7 (reconfigurable) or individual controls can be checked using another configurable hot key.

Further, individual words may be checked by selecting the word and using a configurable hot key. Initially we are providing, on request, lexicons for the following languages: English, Afrikaans, Croatian, French, German, Greek, Italian, Netherlands, Portugese, Russian, Spanish, Swedish, Ukranian.

Since these lexicons are derived from open-source word lists, we will ask developers seeking non-English dictionaries to help us develop a common-words list for their chosen language. A common-words list is a set of the top 25-50 most commonly used words. In English, for example, these are words like "the", "and", "to", and so forth.

Four dictionaries are presently available: English, French, Afrikaans, and Portugese. Aside from English, we chose to release these languages first because developers contacted us and specifically encouraged us to do so.

If you speak and work in any of the above listed, but not-yet-implemented langauges, feel free to contact us and encourage us to give priority to your language in an upcoming build.

Any questions, feel free to ask.

CATEGORY:
TOOLKIT UTILITY

*** HNDBACKUPCONFIG.APP ***

NAME: HNDBACKUPCONFIG.APP
TITLE: Auto-Backup Config (DLL) [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Settings include, the ability to:
1) establish the auto-backup timer,
2) modify the target backup directory,
3) modify the target zip directory
4) modify the target restore directory
5) maintain, delete and zip backup files
6) locate and restore backup files from zip

From this application you are able to select a range of backups and compress them into a single zip file for archiving to a configured location.

You are also able to search and examine zip archives and restore backed up applications and dictionaries to a configurable location.

This application now includes procedure(s) using the template CHTBestPracticeEmbedClass as an example for developers.

As of Build 21D.00.00 this application is now built on HNDNETZIPPER.DLL.

*** HNDBACKUPCONFIGSNAP.APP ***

NAME: HNDBACKUPCONFIGSNAP.APP
TITLE: Auto-Backup Config (SNAP) [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Settings include, the ability to:
1) establish the auto-backup timer,
2) modify the target backup directory,
3) modify the target zip directory
4) modify the target restore directory
5) maintain, delete and zip backup files
6) locate and restore backup files from zip

From this application you are able to select a range of backups and compress them into a single zip file for archiving to a configured location.

You are also able to search and examine zip archives and restore backed up applications and dictionaries to a configurable location.

This specially named version (i.e. SNAP in the app name) is built on CHTSNAPZIP.EXE incorporated as a CHT "Snap-In" on the global template "ApplicationSnapins".

*** HNDBASE64CVT.APP ***

NAME: HNDBASE64CVT.APP
TITLE: Base64 File Command Utility [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDBASE64CVT.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

The application accepts several command line parameters that describe the input file and output path.

These parameters are as follows:

--help [OPT] Displays a help screen. Writes hndbase64cvthelp.txt.
--input [REQ] Path and name of input file
--output [OPT] Output path only - local to this exe if omitted.
--mode [OPT] Encode or decode - defaults to encode if omitted.

The output file name is formatted from the input file with a .h64 extension. For example: test.pdf becomes testpdf.h64

*** HNDBATCHRUNNER.APP ***

NAME: HNDBATCHRUNNER.APP
TITLE: Batch File Run Utility [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

"Batch Bots" include a number of command-line utilities that at time of writing are as follows:

HNDBASE64CVT.EXE (Base 64 encoder/decoder)
HNDENCRYPT.EXE (File encrypter/decrypter with compression)
HNDMAKEHZO.EXE (CHT Container building utility)
HNDFTPNET.EXE (.NET-based FTP utility)
HNDMAILCDO.EXE (Windows CDO-based email send)
HNDMAILNET.EXE (.NET based SMTP email send)
HNDMAILSMTP.EXE (SMTP-based email send)
HNDMAILMAPI.EXE (MAPI-based email send)
HNDZIPNET.EXE (.NET-based file zip utility)
HNDSEARCH.EXE (Text file search/replace utility)
HNDSHCUT.EXE (Shortcut to app creation utility)
HNDSLFCMD.EXE (HTTP upload utility)
CHTSNAPGET.EXE (HTTP download utility)
CHTSNAPZIP.EXE (Zip, Unzip utility)
CHTSNAPSMTP.EXE (SMTP-based email send)
CHTSNAPMAPI.EXE (SMTP-based email send)
CHTSNAPEDIT.EXE (Clarion code and HTML editor)
CHTSNAPSEND.EXE (Error reporting utility)

All of the above are provided to CHT developers pre-compiled as well as in source .APP format so they can be expanded, emulated or customized by CHT developers.

Many command-line utilities of this sort are available that perform a variety of useful tasks. For example:
CLARIONCL.EXE (Clarion generate/compile utility)
UPX.EXE (Executable file compressor)
SIGNCODE.EXE (Windows code-signing utility)
REGASM.EXE (.NET assembly registration utility)

HNDBATCHRUNNER.EXE supports writing, storing, searching and executing of batch file scripts that support your day-to-day use of command-line utilities of the sort described above.

*** HNDBULKSIGNER.APP ***

NAME: HNDBULKSIGNER.APP
TITLE: Bulk Code Signing Utility [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Select a target directory containing several to-be-signed .EXE, .DLL or .OCX files; click select all, followed by the bulk sign button, and watch as in a few seconds the entire batch is signed with your designated code signing certificate.

Log files (*_SIGN.LOG) are created to document each executable as it is signed. If any single executable fails signing, an err file (*_SIGN.ERR) is created indicating what went wrong.

Don't have a code signing certificate? Create your own signing certificate, free of charge using CHT Code Signing Utility (HNDCODESIGN.EXE).

*** HNDCMP.APP ***

NAME: HNDCMP.APP
TITLE: Clarion Project Builder Utility [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

With Clarion Project Builder you can perform the following operations directly without having to open your applications to change their internal configuration settings:

- 1) Switch between EXE and DLL compilation of an app
- 2) Assign a DLL memory load address for your DLL (Optional)
- 3) Change link mode from LIB (library) to DLL
- 4) Switch app compile between Debug and Release mode
- 5) Switch app compile between expiration and non-expiration mode
- 6) Switch from HNDFBACK.TRN (English) to HNDCUSTOM.TRN (User)
- 7) Adjust processor affinity of the EXE for multi processor machines
- 8) Assign code signing information into the executable
- 9) Export the app to TXA
- 10) Compress the executable [Optional]
- 11) Copy the executable to a configured target directory
- 12) Stop compilation on error or continue to next
- 13) Generate app discovery HTML into a configured target directory
- 14) View a compile log for each application giving full details of what happened during compile, code-signing, affinity assigning and compression.

CHT Clarion Project Builder requires that the HNDTOOLS.TPL (template) be present in your Clarion template directory and a CHT template called HNDCMPControlPanel be applied to your applications. This template and file are supplied only to full-fledged CHT subscribers.

*** HNDCODESIGN.APP ***

NAME: HNDCODESIGN.APP
TITLE: Code Signing Utility [V10.18.12]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Have your installation product, supply and install your customer's hardware with a security certificate created with this valuable tool. There is really no reason to purchase an expensive code signing certificate from a third party, except perhaps for the convenience of not having to install a certificate at product installation time.

The latest revision, MARCH 2018, includes the ability to create Server Certificates, Client Certificates, Lifetime Signing Certificates, for CurrentUser or LocalMachine usage, SHA256 or SHA512 encryption level, and Root level or Personal level certificates.

*** HNDFBACKBUILDER.APP ***

NAME: HNDFBACKBUILDER.APP
TITLE: Translation File (TRN) Builder [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

In the past, when CHT shipped a HNDFBACK.TRN (CHT Classes Translation File) that contained some new variables, it was the responsibility of the developer to copy these new variables to their personal HNDCUSTOM.TRN to avoid missing-variable errors during compilation.

This application automates that once manual process.

By opening this application and importing their HNDCUSTOM.TRN under the appropriate language menu, developers are able to generate a new HNDCUSTOM.TRN which merges their personal HNDCUSTOM.TRN with the latest version of HNDFBACK.TRN. Already-translated variables replace the English language versions and new variables are present and marked as "NEW" to make them easy to find.

After merging, and generating a new HNDCUSTOM.TRN based on a combination of their already-translated variables and new ones from HNDFBACK.TRN, developers are able to continue using their own HNDCUSTOM.TRN without having to manually synch new variables to avoid compilation errors.

*** HNDFTPWZ.APP ***

NAME: HNDFTPWZ.APP
TITLE: FTP Wizard (CLARION) [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

- * EmbedInternetFTP (parent template)
- * HandyFTPInitializationControls (gather logon and user information)
- * HandyFTPLocalFilesControls (gather upload file information)
- * HandyFTPLocalFilesULControls (upload files control panel)
- * HandyFTPRemoteFilesControls (gather download file information)
- * HandyFTPRemoteFilesDLControls (download files control panel)

Note that the application can be easily rebuilt for developer customization with the CHTJumpStartProceduresUtility template. Navigate to the FTP Functions tab and select item four, "FTP Wizard".

Or the app may be reproduced and modified by making a copy of HNDFTPWZ.APP and introducing your own modifications.

This demo is a fully functional FTP application with file upload and download capability, proxy support, directory listing and creation, file deletion and moving. While this application shares the features of many commercial FTP applications, it alone uses the CHT query control which allows you to filter your upload and download file lists and create callable queries that really boost your productivity.

*** HNDRUNBAT.APP ***

NAME: HNDRUNBAT.APP
TITLE: Batch File Runner Utility [V10.18.12]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application is part of a batch-file-oriented suite including: HNDBATRUN.EXE, HNDSTOPBATCH.EXE, HNDBATRUNNER.EXE and HNDCMPCX.EXE. They may be used together or individually, though their abilities interconnect.

HNDRUNBAT.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

HNDRUNBAT.EXE accepts several command line parameters that describe the email to be sent. These parameters are as follows:

SENDING:
--bat

--path [REQ] Name of batch file to run, without path
--log [REQ] Path of batch file to run
[OPT] Requires no parameter, opens log on completion
HELP:
--help [OPT] Pops up a splash help screen
--video [OPT] Displays a help video or help document from the web.
--update [OPT] Self-update HNDRUNBAT.EXE to the latest version from the web. Utilizes CHTSNAPGET.EXE if / when utilized.

*** HNDSETUP.APP ***

NAME: HNDSETUP.APP
TITLE: Installation Container Utility [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

CHT's WEBUPDATER installer/updater also assigns this application as the default Windows handler for .HZO files so that you can click on any .HZO file to open it and even unpack it to a target directory if the optional password is known.

A query allows you to describe the files to be containerized. Or files to be containerized may be dropped from MS Windows Explorer.

This application can be used to build standard .HZO installation and update files for transport and decontainerization by applications like CHT WEBUPDATER and HNDSETUPMKR.APP to provide continuous, on-line updating of your software and data products.

A new utility application called HNDSETUPMKR.APP (intended to be cloned and customized) uses a new install-maker template to generate a custom web-installer using the .HZO containers created by this HNDSETUP.APP application.

This application now incorporates three CHT-XSA security settings and a "Post File To Server" dialog, in order to make it quick and easy to upload your .HZO container to a waiting CHT installation server enabled with a CHT-XSA security dialog.

*** HNDSHCUT.APP ***

NAME: HNDSHCUT.APP
TITLE: Shortcut Command Utility [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDSHCUT.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

It has no interface and should be run from the DOS prompt or called with clarion RUN() from an application button or menu.

Parameters must appear in one continuous line on the application command line separated by spaces with each item surrounded by double quotes.

Parameters are as follows:

1) Application name and path
eg: "c:\c10\accessory\hnd\hndcmp.exe"

2) Application description.
eg: "CHT Clarion Project Builder"

- 3) Short cut file name.
eg: "CHT Clarion Project Builder.Ink"
- 4) Application work path
eg: "c:\clarion10\hndapps"
- 5) Installation location (usually "5" for \program files\
eg: "5"
- 6) Installation Icon (usually "0" for first application icon)
eg: "0"

*** HNDSQLTRACE.APP ***

NAME: HNDSQLTRACE.APP
TITLE: SQL Trace Utility [V10.19.08]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Start tracing by configuring the specific driver(s) used by your SQL application followed by a click on the "Minimize" button . Once this application is started, then start your SQL application.

Navigate in your SQL application to the specific area to be tested and bring HNDSQLTRACE.APP back up using the windows task bar icon menu provided.

Manual tracing:

When HNDSQLTRACE.APP re-appears all SQL statements executed to that point are displayed in reverse chronological order, with newest statements at the top. From this point forward, to trace specific actions, click the "Clear" button and interact briefly with your SQL application. Click the "Read Next" button to view the most recent SQL activity raised by your application. Repeat as necessary.

Automatic tracing:

Check the "Auto Trace" switch. A window timer will periodically refresh to the latest SQL select or group of selects issued by your application.

File Trace or Direct Trace:

These modes are mutually exclusive.

File trace is performed by communicating with SQL trace information issued by Clarion's file drivers into a text file. In "File Trace" mode a developer can trace any and all SQL select outputs from various browses in an application as long as HNDSQLTRACE.EXE is started and configured before starting the application.

"Direct Trace" mode is performed by low-level communication, usually from a specific procedure under scrutiny. This mode requires one of CHT's browse templates to be configured to issue trace information during interaction with the user.

This application includes the template CHTBestPracticeEmbedClass as an example for developers to study.

NOTE: As of November 1, 2015, and C10 Version 11975 file trace configuration information is now stored in "DRIVERS.INI" located at CSIDL_APPDATA\SoftVelocity\Clarion\CLARION_VERSION\
In all earlier versions of clarion, file trace configuration information was located in long obsolete, win.ini.

*** HNDSTOPBATCH.APP ***

NAME: HNDSTOPBATCH.APP
TITLE: Batch File Stop Utility [V10.18.12]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Batch scripts can be set to check for this file, in order to stop long batch scripts in mid-stream. CHT Clarion Project Builder generates a STOP.TXT file check into its scripts to stop them before they complete.

HNDSTOPBATCH.EXE source code, because it is in Clarion .APP format, is included as a standard part of the CHT toolkit, provided both in .APP format as well as ready-to-run, in .EXE format.

This splash screen is only added to the application for documentation purposes.

*** HNDTPLBKC10DLL.APP ***

NAME: HNDTPLBKC10DLL.APP
TITLE: Template Extension DLL
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This DLL is called from within CHT templates to perform various background tasks during the execution of our template code.

Some Specific tasks performed:

Back up .APP and .DCT files as you're working on them...
Run various CHT Utility Executables that extend IDE capabilities...
Display CHT's Website and help documents located there...
Display CHT's Forum via auto-login...

*** HNDUTILITIESDLL.APP ***

NAME: HNDUTILITIESDLL.APP
TITLE: HNDUTIL Classes DLL [V10.18.11]
DATE: 8/24/2019
CATEGORY: TOOLKIT UTILITY
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The DLL is used for illustration purposes in another CHT demo application called HNDTESTUTILITIES.C9.APP where you can see how the exported functions available in this DLL are used.

This application, besides providing a useful DLL that extends the Clarion language serves to illustrate how a CHT class (or classes) can be packaged as a DLL and used in your applications as if they were part of the Clarion language.

Powerful stuff this Clarion Handy Tools! Blows other 3rdparty tools right out of the water.

Any questions, feel free to ask on the CHT Subscriber forum: [http://news.cwhandy.ca/LOG\\$](http://news.cwhandy.ca/LOG$)

**CATEGORY:
UTILITY FUNCTIONS**

*** HNDTESTUTILITIES.APP ***

NAME: HNDTESTUTILITIES.APP
TITLE: Demonstrate HNDUTILITIES.DLL [V10.18.09.0242]
DATE: 8/24/2019
CATEGORY: Utility Functions
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Categories of functionality available are as follows:
Manipulating/Finding Files and Paths, Windows SHELL, Email Address Validity Check, Case Adjustment, Disassembling Character Delimited Strings, Directories, Drives, Windows GUID, Determining Hash Values, Determining Data Type, Windows Environment, Reading/Writing Text Files, Windows System Manipulation, HTML HEX Colors, HTML Escape Characters, Windows Error Log, Runtime Error Viewing/Tracing, Memory Files, Manipulating/Extracting Strings

CATEGORY: VIDEO PLAYER

*** HNDVIDEOEDUCATION.APP ***

NAME: HNDVIDEOEDUCATION.APP
TITLE: VIDEO PLAYER (EDU) [V10.19.08]
DATE: 8/24/2019
CATEGORY: VIDEO PLAYER
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The player is built on CHT Template HandyComMediaControl which uses the Windows Media Player COM object based in WMP.DLL.

The player is capable of playing off-line any videos which have been previously viewed to completion. On-line viewing of uncached videos begins quickly, once approximately 5 percent of the video has downloaded, and already-cached videos begin immediately since they are not downloaded again.

If you plan to adapt this application to act as a video player for your own videos, a useful, accompanying application called HNDBUILDVIDEOLIST.APP can build video index files on your CHT-based video server, of the type utilized by this application and by the HNDWindowsMediaPlayer class (HNDWMP.INC, HNDWMP,CLW, HNDWMP.TLB).

*** HNDVIDEOPLAYER.APP ***

NAME: HNDVIDEOPLAYER.APP
TITLE: VIDEO PLAYER (USR) [V10.19.04]
DATE: 8/24/2019
CATEGORY: VIDEO PLAYER
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This "CHT Utility Application" is designed to be used as a general use, private video player (variation 3). It streams .MP4 versions of MP4 videos from one of our CHT Video Servers. At the same time the application also serves as a model for developers wishing to build their own streaming video players.

By way of example, we will present a variety of science and music videos that reflect our interests. The chief advantage of using a private video player rather than running the standard Windows media player is that this one does not reveal the location of videos and can act as a private video player or even an in-home player hooked to a private in-home video server.

The player is built on CHT Template HandyComMediaControl which uses the Windows Media Player COM object based in WMP.DLL.

The player is capable of playing off-line any videos which have been previously viewed to completion. On-line viewing of uncached videos begins quickly, once approximately 5 percent of the video has downloaded, and already-cached videos begin immediately since they are not downloaded again.

If you plan to adapt this application to act as a video player for your own videos, a useful, accompanying application called HNDBUILDVIDEOLIST.APP can build video index files on your CHT-based video server, of the type utilized by this application and by the HNDWindowsMediaPlayer class (HNDWMP.INC, HNDWMP,CLW, HNDWMP.TLB).

CATEGORY: WINDOWS API

*** HND5.APP ***

NAME: HND5.APP
TITLE: CHT HNDDisk Classes Demo [V10.19.02.0517]

DATE: 8/24/2019
CATEGORY: Windows API
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Some examples of the things illustrated:

- Creating nested directories
- Copying and moving files
- Finding the directory of a file using only its file name
- Determining drive size and space remaining, the number of drives and drive types
- Determining the unique ID of a system's hard disk
- Launching Windows File Explorer from inside your app
- For file copying, moving, erasing, renaming in bulk see HNDCPYDM.APP.
- For file compression and decompression in bulk see HNDZDEMO.APP.
- For file containerization and decontainerization in bulk see HNDSETUP.APP.

*** HND8.APP ***

NAME: HND8.APP
TITLE: Window Functions And Callbacks [V10.18.12.0224]
DATE: 8/24/2019
CATEGORY: Windows API
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

HNDWindow has numerous window-related capabilities some of which are demonstrated in this application. Other templates that create (or use) instances of HNDWindow and use other features of that class are:

- * HandyTellMeControl - scrollable text message areas
- * VisitMeControl - window hot links to your website.
- * VisitMeControl_EWF - window hot links advanced
- * HandyMailMeControl - window hotlinks for email
- * HandyMailMeControl_EWF - window hotlinks for email advanced
- * HandyHelpMessageBox - easy popup message boxes
- * ApplicationImagesEx - image management application global
- * ProcedureImagesEx - image management procedure specific
- * ExtendedEvents - trap and act on extended MS Windows Events

This application has a companion called hnd8a.app which must also be compiled in order to illustrate inter-application messaging and data transfer. If you're using the source versions of CHT's example applications, make sure to compile hnd8.app before running this one.

*** HND8A.APP ***

NAME: HND8A.APP
TITLE: Window Functions And Callbacks [V10.18.12.0213]
DATE: 8/24/2019
CATEGORY: Windows API
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

*** HNDMONITORDIRECTORYSRVC.APP ***

NAME: HNDMONITORDIRECTORYSRVC.APP
TITLE: CHT Monitor Directory Service [V10.18.12.0284]
DATE: 8/24/2019
CATEGORY: Windows API
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The class attached to your procedure by this template, once initialized, gives your procedure the ability to detect all changes occurring in a nominated directory and, optionally, its subdirectories. Whenever a new file is added to the target area, or a file is changed, your procedure is sent an event called EVENT:FileChanged to signal that this has occurred. At that point, what you do as a programmer to act on that event is wide open.

This application illustrates one use of this powerful, directory monitoring capability. When a file is added to the designated source directory, that file is copied into a target directory somewhere else on the local network.

The file or files meeting the filter setting provided on the application interface, eventually then, end up sitting somewhere else on the network, for instance, in a server directory. There is also an optional switch to remove the processed source file(s) from the directory being monitored.

Any questions, feel free to ask.

CATEGORY: WINDOWS FAVORITES UTILITY

*** HNDFAVORITER.APP ***

NAME: HNDFAVORITER.APP
TITLE: CHT FavorRiter [V10.18.08.0317]
DATE: 8/24/2019
CATEGORY: Windows Favorites Utility
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Once organized, favorites can be exported selectively or entirely to a desktop folder called "Favorites1", "Favorites2", and so on, organized by topics as determined by the user.

The exported folder does not interfere in any way with the existing MS Internet Explorer favorites. Yet the folder produced can be used to access favorites directly without first needing to open I.E.

The "Favorites" folders produced can be zipped and shared with others in whole or in part.

Local documents and applications and even email addresses can be installed as favorites. These, naturally, are less portable than remote web addresses, unless they're located in the same directory structure on the machine of a recipient of your "Favorites" folder.

CATEGORY: WINDOWS REGISTRY

*** HND7.APP ***

NAME: HND7.APP
TITLE: CHT Registry Demonstration [V10.18.12.0236]
DATE: 8/24/2019
CATEGORY: Windows Registry
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This template instantiates an OOP class called HNDRegistry which deals exclusively with reading, writing and deleting from the MS Windows (tm) registry. For a full listing of methods and properties available in this class open the class header module called HNDREG.INC.

This application will let you test reading, writing and deleting registry entries on your system. Follow the instructions on the "Registry Demo" window.

CATEGORY: WINDOWS SERVICES

*** HNDBACKUPSERVICE_A.APP ***

NAME: HNDBACKUPSERVICE_A.APP
TITLE: CHT Backup Service A [V10.18.12.0161]
DATE: 8/24/2019
CATEGORY: Windows Services
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application runs as a service to back up user-designated directories using file-limiting masks at user-designated times. A configuration file called HNDBACKUPSERVICE_A.INI provides a backup time, a target copy directory and a backup hzo directory. Up to 100 source backup descriptions that include source directory and file-mask are possible.

To install the service, from the DOS prompt type:
HNDBACKUPSERVICE_A /IS
To start the service - once installed - from the DOS prompt type:
HNDBACKUPSERVICE_A /SS
To stop the service from the DOS prompt type:
HNDBACKUPSERVICE_A /STOP
To remove the service, from the DOS prompt type:
HNDBACKUPSERVICE_A /RS

*** HNDBACKUPSERVICE_B.APP ***

NAME: HNDBACKUPSERVICE_B.APP
TITLE: CHT Backup Service B [V10.18.12.0249]
DATE: 8/24/2019
CATEGORY: Windows Services
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application runs as a service to back up user-designated directories using file-limiting masks at user-designated times. A configuration file called HNDBACKUPSERVICE_A.INI provides a maximum of 10 backup times and 100 backup descriptions that include source directory, file-mask, target directory.

To install the service, from the DOS prompt type:
HNDBACKUPSERVICE_A /IS
To start the service - once installed - from the DOS prompt type:
HNDBACKUPSERVICE_A /SS
To stop the service from the DOS prompt type:
HNDBACKUPSERVICE_A /STOP
To remove the service, from the DOS prompt type:
HNDBACKUPSERVICE_A /RS

*** HNDHOTKEYSERVICE.APP ***

NAME: HNDHOTKEYSERVICE.APP
TITLE: CHT Hot-Key Service [V10.18.07.0317]
DATE: 8/24/2019
CATEGORY: Windows Services
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This particular demo application runs as a Windows service that enables a set of system-wide hot-keys configured launch a variety of CHT support applications as follows:

**** Getting CHT Help (F5 Key) ****
F5 Key = Restore or minimize web-service window
Alt-F5 = Open browser to CHT Home Page
Ctrl-F5 = Open browser to CHT Video Education Page

**** CHT Forum Access (F6 Key) ****
F6 Key = CHT Hybrid Web Client

Shift-F6 = CHT Support Forum Client
Ctrl-F6 = CHT Support Forum Via Browser

**** CHT Installation And Update (F7 Key) ****

F7 Key = CHT Webupdater
Shift-F7 = CHT Webupdater Download Page
Ctrl-F7 = CHT Installation Tuner
Alt-F7 = CHT Auto Backup Configurator

**** CHT Development Utilities (F8 Key) ****

Shift-F8 = CHT Color Configurator
Ctrl-F8 = CHT Toolkit Synchronizer
Alt-F8 = CHT Web Scripser

**** CHT Other Utilities (F9Key) ****

F9 = CHT Installation Containerizer
Shift-F9 = CHT FTP Wizard
Ctrl-F9 = CHT Handy Zip'N Email
Alt-F9 = CHT Handy Zip'N HTTP Post
CtrlShift-F9 = CHT Handy Zip'N FTP

*** HNDSERVICEUP.APP ***

NAME: HNDSERVICEUP.APP
TITLE: CHT Service-Up Local [V10.18.09.0153]
DATE: 8/24/2019
CATEGORY: Windows Services
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This specific example application interacts with another CHT demo application called hndwindowsservicedemo.app, but an application of this sort could be made to control multiple services running on the same hardware.

Key templates installed are EmbedWindowFunctions to handle inter-application communication by posting events to the target service's window, EmbedShellFunctions to handle starting and stopping the target service and HandyServiceRemoteControlButtons_ESHF to write all the remote control code.

Any questions, feel free to ask: news.cwhandy.com

*** HNDSERVICEUPREMOTE.APP ***

NAME: HNDSERVICEUPREMOTE.APP
TITLE: CHT Service-Up Remote [V10.18.09.0149]
DATE: 8/24/2019
CATEGORY: Windows Services
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This specific example application interacts with another CHT demo application called HNDWINDOWSSERVICEDEMO.APP, but an application of this sort could be made to control multiple services running on the other network-connected hardware.

Key templates installed are HandyCOMMailSlotControl to handle starting and stopping the target service and HandyServiceRemoteControlButtons_HCMSC to write all the remote control code.

Any questions, feel free to ask: news.cwhandy.com

*** HNDSTARTERSERVICE_A.APP ***

NAME: HNDSTARTERSERVICE_A.APP
TITLE: CHT Application Starter Service A [V10.18.09.0159]
DATE: 8/24/2019
CATEGORY: Windows Services
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application runs as a service to start user-designated applications (like servers) listed in the application's configuration file HNDSTARTERSERVICE_A.INI. Up to 100 applications may be started by this service. A delay time setting allows sufficient time after a re-boot for the machine to settle down before your applications are started, each about 30 seconds apart.

To install the service, from the DOS prompt type:

```
HNDBACKUPSERVICE_A /IS
```

To start the service - once installed - from the DOS prompt type:

```
HNDBACKUPSERVICE_A /SS
```

To stop the service from the DOS prompt type:

```
HNDBACKUPSERVICE_A /STOP
```

To remove the service, from the DOS prompt type:

```
HNDBACKUPSERVICE_A /RS
```

*** HNDWINDOWSSERVICEDEMO.APP ***

NAME: HNDWINDOWSSERVICEDEMO.APP
TITLE: CHT Demonstration NT Service [V10.18.09.0154]
DATE: 8/24/2019
CATEGORY: Windows Services
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

To install this service application, at the DOS prompt type:
HNDWINDOWSSERVICEDEMO /IS or /INSTALL

To start the service once installed, at the DOS prompt type:
HNDWINDOWSSERVICEDEMO /SS or /START

To stop the service once installed, at the DOS prompt type:
HNDWINDOWSSERVICEDEMO /SP or /STOP

To remove the service once installed, at the DOS prompt type:
HNDWINDOWSSERVICEDEMO /RS or /REMOVE

See HNDSERVICEUP.APP and HNDSERVICEUPREMOTE.APP for examples of how to control and configure this service remotely from another application.

This application starts with it's window minimized and hidden. It's application icon appears on the Windows Task Bar from which it can be commanded. By unchecking the "Application In Tray" switch on the EmbedWindowFunctions template in the main procedure, this application can be made to run completely hidden, while still being configured remotely from HNDSERVICEUPREMOTE.APP or HNDSERVICEUP.APP.

Any questions, feel free to ask.

CATEGORY:
XML2HTML

*** HNDMAILBATCHER.APP ***

NAME: HNDMAILBATCHER.APP
TITLE: XML2HTML Bulk Mail Utility #1 [V19.08.23]
DATE: 8/24/2019
CATEGORY: XML2HTML
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Generate HTML Mail With Internal Style Sheets

It is important for HTML email to use internal style sheets so that the email, on arriving in the recipient's inbox, does not need to request external style sheets from the web. It is therefore deemed "safe" by email clients and it displays immediately, like text mail, without the user needing to click any kind of "allow" switch.

Style Sheet Generating Template

This application illustrates several "_mail_" XML2HTML example templates that allow the developer to change every important aspect of numerous useful HTML style sheets so that different "looks" can easily be incorporated into different HTML-generating-procedures. To create HTML outputs with new and different looks, simply copy the existing example procedure and add your own CSS styles (or modify ours) by adjusting template switches and by removing or adding to the various "sections" incorporated into our examples.

Select Single Or Multiple Email Addresses

Select one or more email addresses from the email browse(s) using standard MS Windows record marking and an .HTML file written ahead of time using CHTSNAPEDIT.EXE or generated by one of this application's "AutoGen" procedures.

A batch file is generated, which in a loop, successively calls HNDMAILSMTP.EXE or CHTSNAPSMTP.EXE (depending which one you choose to use) to send the HTML email body to the addresses selected above. The batch file may be run either from the DOS prompt or from a simple desktop link or from inside your own Clarion applications, or using CHT's HNDBATCHRUNNER.EXE. utility.

An example of this run-from-Clarion concept, is provided in the "Run (Send Mail With) Last Generated Batch Script" menu of this application.

CHT Bulk Mail Batcher is able to import email addresses exported from other applications such as Windows Live Mail.

Preconfigure your SMTP email-send account beforehand and select that account for sending before generating your batch file.

*** HNDNETMAILPROMO.APP ***

NAME: HNDNETMAILPROMO.APP
TITLE: XML2HTML Docs Mail Utility #2 [V10.19.08]
DATE: 8/24/2019
CATEGORY: XML2HTML
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This application processes the HTML-email directly via the above-mentioned DLL, rather than by starting an executable batch utility. Email creation and generation features, and XML and HTML source files are identical to those provided in HNDBULKMAILBATCHER.APP for cross-compatibility.

Generate HTML Mail With Internal Style Sheets

It is important for HTML email to use internal style sheets so that the email, on arriving in the recipient's inbox, does not need to request external style sheets from the web. It is therefore deemed "safe" by email clients. It displays immediately, like text mail without the user needing to click any kind of "allow" switch.

Style Sheet Generating Template

This application illustrates a CHT template that allows the developer to change every important aspect of numerous useful HTML style sheets so that different "looks" can easily be incorporated into different HTML-generating-procedures. To create HTML outputs with new and different looks, simply copy the existing example procedure and add your own style and flare by adjusting template switches and by removing or adding to the various "sections" incorporated into our example procedures.

Select Single Or Multiple Email Addresses

Select one or more email addresses from the email browse(s) using standard MS Windows record marking and an .HTML file written ahead of time using CHTSNAPEDIT.EXE or generated by one of this application's HTML-generating procedures.

When "Send" is invoked, on a group of selected records, the HNDNETSMTP.DLL sends the HTML email body to the addresses selected. Each email is processed and a success or failure outcome string is written into the email file. Images inserted into the email body are processed and included as always.

CHT Bulk Net Mail Promo is able to import email addresses exported from other applications such as Windows Live Mail. And up to 9 different email address source files can be maintained in this application. These files are shared with CHT Bulk Mail Batcher.

Preconfigure your SMTP email-send account beforehand and select that account for sending before attempting to send any mail.

*** HNDDOCUMENTBUILDER.APP ***

NAME: HNDDOCUMENTBUILDER.APP

TITLE: XML2HTML Docs Builder Utility [V10.19.08]
DATE: 8/24/2019
CATEGORY: XML2HTML
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

To create a PDF of any HTML document generated by CHT HTML Document Builder, just print it to disk with Windows native PDF printer driver.

Style Sheet Generating Template

This application illustrates a new template that allows the developer to change every important aspect of numerous useful HTML style sheets so that different "looks" can easily be incorporated into different HTML-generating-procedures. To create HTML outputs with new and different looks, simply copy the existing example procedure and add your own style and flare by adjusting template switches and by removing or adding to the various "sections" incorporated into our examples.

*** HNDPREVIEWER.APP ***

NAME: HNDPREVIEWER.APP
TITLE: XML2HTML Previewer [V10.19.08]
DATE: 8/24/2019
CATEGORY: XML2HTML
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Menus are provided to instantly set the previewer's window size to one of a variety of viewport sizes for common device sizes from small phones like the Apple iPhone to larger phones, tablets and, of course, Windows desktop.

Interacting with "CHT Document Builder" applications, like CHTSNAPEDIT.EXE and HNDDOCUMENTBUILDER.EXE, this application picks up an HTML directory location or URL from the command line passed to it from the calling application, and displays the HTML content just generated. This provides instant feedback relative to the the document changes just made.

*** HNDPRINTHTML.APP ***

NAME: HNDPRINTHTML.APP
TITLE: XML2HTML Preview-Print [V10.19.08]
DATE: 8/24/2019
CATEGORY: XML2HTML
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

Buttons are supplied:

- 1) to select and load another local HTML file
- 2) to print the currently loaded file to the print device (PRN, PDF) selected by the user from a printer select dialog.

A URL entry control is supplied with which a remote web-site HTML file may be loaded, displayed and printed.

This app picks up an HTML file name (with path) from the command line if you wish to utilize the app as a command-line utility using Clarion RUN() or from a batch file.

If you wish to import the display/print procedure PreviewPrintHTMLMain() into your application from our app, that's easy enough to do as well.

*** HNDSLFVVCUS.APP ***

NAME: HNDSLFVVCUS.APP
TITLE: XML2HTML Data Utility [V10.18.12]
DATE: 8/24/2019
CATEGORY: XML2HTML

CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

** HNDSLFSVXMLDATASVR.APP
** HNDNEWSLINKLISTSVR.APP
*** HNDSLFSVXMLFILESVR.APP

It provides maintenance of the data tables in HNDSLFSVCUS.SQLITE.

It also shares a dictionary named HNDSLFSV.DCT with these server applications. In this case, the default table name in use is HNDSLFSVCUS.SQLITE.

***** HNDTILEAPPLAUNCHER.APP *****

NAME: HNDTILEAPPLAUNCHER.APP
TITLE: XML2HTML EXAMPLES [V10.19.08]
DATE: 8/24/2019
CATEGORY: XML2HTML
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The browser control on the MAIN window, launches "hndtileapplaunchermenus.html" automatically for purposes of demonstration but changing the HTML file being displayed by this application is trivial. This file is assumed to be in your application directory and if missing it is regenerated by HNDXML2HTMLGEN.EXE from an XML template we've provided called "000hndtileapplauncher.xml".

The finished HTML file is copied by the XML template to the name hndtileapplaunchermenus.html for the purposes of this example. Finished HTML names of generated entities is totally under developer control.

This application, illustrates six other .XML templates by opening them into CHTSNAPEDIT.EXE and suggesting you preview the XML file as HTML by pressing the F5 key in the editor after first clicking anywhere in the editor work surface.

Dependency files required for this application are listed in the "Version Notes To Myself" on the AACHTControlPanel information tab.

***** HNDXML2HTMLGEN.APP *****

NAME: HNDXML2HTMLGEN.APP
TITLE: XML2HTML Command Utility [V10.19.08]
DATE: 8/24/2019
CATEGORY: XML2HTML
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

This "CHT Batch Bot" application runs from various other CHT Batch-Bots and Snap-Ins in order to generate user-embedded XML template files into stand-alone HTML pages and forms.

It may also be run directly from a DOS batch file or developer application implementing XML to HTML processing.

CHT Snap-Edit now calls this HTML generator exclusively to transform XML template files into HTML before previewing.

This generator implementation can access and call remote procedures established in a CHT client data server to request data packages which are then merged into the HTML document. Several example XML templates have been provided to illustrate insertion of remote data from a CHT server by UCR (user custom request): 000CHTRECEIPT.XML for example.

A document is in preparation which will explain in detail the structuring of various interchangeable XML template designs with which this generator-application will operate to create HTML. For example, mail-merge letters, static web pages, interactive web forms, tile menus and more.

GENERATING:

--input [REQ] name of the XML input template file
--output [OPT] name of the HTML output template file
--preww [OPT] preview HTML after generate
--nopreww [OPT] no preview HTML after generate
CONFIGURE DATA SERVER:
--config [OPT] configure remote data server for secure access
HELP:
--help [CFG] Pops up a splash help screen
--video [OPT] Displays a video or document from the web that further explains use of this application.

CATEGORY: ZIP-N-POST

*** HNDZIPNEMAILBBOT.APP ***

NAME: HNDZIPNEMAILBBOT.APP
TITLE: ZIP-N-POST (B-BOT EMAIL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured using up to 20 potential configurations to back up a particular directory using a CHT Query that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

An SMTP configuration dialog lets you preconfigure up to 3 selectable SMTP servers to handle the sending of your email and zip attachment.

Once configured, your application can build a zip with one button click and send it to one or more recipients with a second click. Up to twenty separate, unique configurations are retained for instant re-use.

*** HNDZIPNEMAILBBOTCDO.APP ***

NAME: HNDZIPNEMAILBBOTCDO.APP
TITLE: ZIP-N-POST (B-BOT CDO) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured using up to 20 potential configurations to back up a particular directory using a CHT Query that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

An SMTP configuration dialog lets you preconfigure up to 3 selectable SMTP servers to handle the sending of your email and zip attachment.

Once configured, your application can build a zip with one button click and send it to one or more recipients with a second click. Up to twenty separate, unique configurations are retained for instant re-use.

*** HNDZIPNEMAILBBOTMAPI.APP ***

NAME: HNDZIPNEMAILBBOTMAPI.APP
TITLE: ZIP-N-POST (B-BOT MAPI) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured using up to 20 potential configurations to back up a particular directory using a CHT Query that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

This application's configuration button reminds only that MAPI email leverages the default email client to send mail, so it is only necessary that a working email client (e.g. Windows Live, Outlook) be present in order for email sending to operate correctly.

Once configured, your application can build a zip with one button click and send it to one or more recipients with a second click. Up to twenty separate, unique configurations are retained for instant re-use.

*** HNDZIPNEMAILCDO.APP ***

NAME: HNDZIPNEMAILCDO.APP
TITLE: ZIP-N-POST (CDO MAIL) [V10.19.08]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured using up to 20 potential configurations to back up a particular directory using a CHT Query that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

An SMTP/CDO configuration dialog lets you preconfigure up to three target SMTP servers to handle the sending of your email and zip attachment.

Your application can build a zip with one button click and send it to one or more recipients with a second click. Up to twenty separate, unique configurations are retained for instant re-use.

*** HNDZIPNEMAILMAPI.APP ***

NAME: HNDZIPNEMAILMAPI.APP
TITLE: ZIP-N-POST (MAPI MAIL) [V10.19.08]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured using up to 20 potential configurations to back up a particular directory using a CHT Query that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

Email configuration is carried out on the system's default MAPI client software. Success with this MAPI client is dependent on a MAPI-compliant email application being installed and configured on the hosting computer.

This application's email configuration button reminds only that MAPI email leverages the default email client to send mail, so it is only necessary that a working email client (e.g. Windows Live, Outlook) be present in order for email sending to operate correctly.

Your application can build a zip with one button click and send it to one or more recipients with a second click. Up to twenty separate, unique configurations are retained for instant re-use.

*** HNDZIPNEMAILNET.APP ***

NAME: HNDZIPNEMAILNET.APP

TITLE: ZIP-N-POST (NET MAIL) [V10.19.08]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured using up to 20 potential configurations to back up a particular directory using a CHT Query that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

An SMTP configuration dialog lets you preconfigure up to three target SMTP servers to handle the sending of your email and zip attachment.

Once configured, your application can build a zip with one button click and send it to one or more recipients with a second click. Up to twenty separate, unique configurations are retained for instant re-use.

*** HNDZIPEMAILSMTP.APP ***

NAME: HNDZIPEMAILSMTP.APP
TITLE: ZIP-N-POST (SMTP MAIL) [V10.19.08]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured using up to 20 potential configurations to back up a particular directory using a CHT Query that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

An SMTP configuration dialog lets you preconfigure up to three target SMTP servers to handle the sending of your email and zip attachment.

Once configured, your application can build a zip with one button click and send it to one or more recipients with a second click. Up to twenty separate, unique configurations are retained for instant re-use.

*** HNDZIPEMAILSNAP.APP ***

NAME: HNDZIPEMAILSNAP.APP
TITLE: ZIP-N-POST (SNAP MAIL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured using up to 20 potential configurations to back up a particular directory using a CHT Query that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

An SMTP configuration dialog lets you preconfigure up to three SMTP servers to handle the sending of your email and zip attachment.

Once configured, your application can build a zip with one button click and send it to one or more recipients with a second click. Up to twenty separate, unique configurations are retained for instant re-use.

*** HNDZIPEMAILSNAPMAPI.APP ***

NAME: HNDZIPEMAILSNAPMAPI.APP
TITLE: ZIP-N-POST (MAPI MAIL) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)

TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured using up to 20 potential configurations to back up a particular directory using a CHT Query that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

This application's email configuration button reminds only that MAPI email leverages the default email client to send mail, so it is only necessary that a working email client (e.g. Windows Live, Outlook) be present in order for email sending to operate correctly.

Once configured, your application can build a zip with one button click and send it to one or more recipients with a second click. Up to twenty separate, unique configurations are retained for instant re-use.

*** HNDZIPNFTP.APP ***

NAME: HNDZIPNFTP.APP
TITLE: ZIP-N-POST (FTP CLIENT) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application provides 20 unique pre-configurations each of which can back up a specified directory using a CHT file query that describes the files to be backed up. This results in a standard zip file for each configuration which can include source subdirectories right inside the zip.

An FTP setup dialog lets you configure a target FTP server with an optionally separate target web directory to accept your zip files.

Once configured, your application can build a backup with one button click and send it to an FTP server with a second click.

Up to 20 different backup configurations may be pre-created for immediate re-use and moving from one configuration to the other requires only one button click.

*** HNDZIPNFTPNET.APP ***

NAME: HNDZIPNFTPNET.APP
TITLE: ZIP-N-POST (FTP CLIENT) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application provides 20 unique pre-configurations each of which can back up a specified directory using a CHT file query that describes the files to be backed up. This results in a standard zip file for each configuration which can include source subdirectories right inside the zip.

An FTP setup dialog lets you configure a target FTP server with an optionally separate target web directory to accept your zip files.

Once configured, your application can build a backup with one button click and send it to an FTP server with a second click.

Up to 20 different backup configurations may be pre-created for immediate re-use and moving from one configuration to the other requires only one button click.

*** HNDZIPNPOSTTOCHTSERVER.APP ***

NAME: HNDZIPNPOSTTOCHTSERVER.APP
TITLE: ZIP-N-POST (HTTP CLIENT) [V10.19.08]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)

TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured to 20 different settings to back up a variety of directories using a CHT File Query that describes the files to be backed up. Each setting results in a standard zip file that can optionally include source subdirectories right inside the zip. Moving from one configuration to the next requires only one button click.

The HTTP configuration dialog also lets you preconfigure the target CHT file server designated to accept your zip file(s).

This application utilizes CHT-XSA secure, encrypted authentication, so that you can authenticate against any CHT Server file server that has been provided with a CHT-XSA configuration dialog.

Once configured, your application can build a backup with one button click and PUT it to a CHT-Style secure XSA web server with a second click.

***** HNDZIPNPOSTTOCHTSERVEROCX.APP *****

NAME: HNDZIPNPOSTTOCHTSERVEROCX.APP
TITLE: ZIP-N-POST (HTTP CLIENT) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured to back up a particular directory using a file mask that describes the files to be backed up, resulting in a standard zip file that can include source subdirectories right inside the zip.

An HTTP configuration dialog lets you preconfigure a target CHT file server to accept your zip file(s).

Once configured, your application can build a backup with one button click and PUT it to a web server with a second click.

Up to 20 different backup configurations may be pre-created and moving from one to the other requires only one button click.

NOTE: This application uses HNDZIPR.OCX to perform its compression duties.

***** HNDZIPNPOSTTOCHTSETUP.APP *****

NAME: HNDZIPNPOSTTOCHTSETUP.APP
TITLE: ZIP-N-POST (HTTP CLIENT) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured to 20 different settings to back up a variety of directories using a CHT File Query that describes the files to be backed up. Each setting results in a standard zip file that can optionally include source subdirectories right inside the zip. Moving from one configuration to the next requires only one button click.

The HTTP configuration dialog also lets you preconfigure the target CHT file server designated to accept your zip file(s).

This application utilizes CHT-XSA authentication, so that you can authenticate against any CHT Server which has been provided with a CHT-XSA configuration dialog.

Once configured, your application can build a backup with one button click and PUT it to a CHT-Style secure XSA web server with a second click.

***** HNDZIPNPOSTTOCHTXSA.APP *****

NAME: HNDZIPNPOSTTOCHTXSA.APP

TITLE: ZIP-N-POST (HTTP CLIENT) [V10.18.12]
DATE: 8/24/2019
CATEGORY: ZIP-N-POST
CLASSES: [List App Classes](#)
TEMPLATES: [List App Templates](#)

DESCRIPTION:

The application can be pre-configured to 20 different settings to back up a variety of directories using a CHT File Query that describes the files to be backed up. Each setting results in a standard zip file that can optionally include source subdirectories right inside the zip. Moving from one configuration to the next requires only one button click.

The HTTP configuration dialog also lets you preconfigure the target CHT file server designated to accept your zip file(s).

This application utilizes CHT-XSA 2-part authentication, so that you can authenticate against any CHT Server which has been provided with a CHT-XSA configuration dialog.

Once configured, your application can build a backup with one button click and PUT it to a CHT-Style secure XSA web server with a second click.