

# CHT Applications

[HOME](#) [CLASSES](#)

[TEMPLATES](#) [APPLICATIONS](#)

## CHT SNAP-IN & BATCH-BOT UTILITIES (BUILD: 22C.00.00) JUNE 27, 2018

### CATEGORY: BATCH BOT

\*\*\* HNDENCRYPT.APP \*\*\*

**NAME:** HNDENCRYPT.APP  
**TITLE:** BATCH-BOT ENCRYPTION [V10.18.05]  
**DATE:** 6/27/2018  
**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 [V10.18.05]  
**DATE:** 6/27/2018  
**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  
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.  
  
--update  
[OPT] This command causes the application to update  
to the latest version available from CHT's website.  
Utilizes CHTSNAPGET.EXE if / when invoked.

**NAME:** HNDMAILCDO.APP  
**TITLE:** BATCH-BOT CDO MAIL [V10.18.05]  
**DATE:** 6/27/2018  
**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 [CFG] Pops up a splash help screen  
--update [CFG] Self-update HNDMAILCDO.EXE to the latest version.  
Utilizes CHTSNAPGET.EXE if / when unvoked.  
--video [OPT] Displays a video or document from the web that further explains use of this application.

\*\*\* HNDMAILMAPI.APP \*\*\*

**NAME:** HNDMAILMAPI.APP  
**TITLE:** BATCH-BOT MAPI MAIL [V10.18.05]  
**DATE:** 6/27/2018  
**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 "hndmailmapi.log" located in the same directory as the hndmailmapi.exe.

**HELP:**

--help Pops up a splash help screen  
--update Auto-update HNDMAILMAPI.EXE to the latest version

\*\*\* HNDMAILNET.APP \*\*\*

**NAME:** HNDMAILNET.APP  
**TITLE:** BATCH-BOT NET MAIL [V10.18.05]  
**DATE:** 6/27/2018  
**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=

#### 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 this splash help screen

--video [OPT] Displays from the web a video or a document further explaining use of the application.

--update [OPT] Auto-update HNDMAILNET.EXE to the latest version  
Utilizes CHTSNAPGET.EXE if / when invoked.

\*\*\* HNDMAILSMTP.APP \*\*\*

**NAME:** HNDMAILSMTP.APP  
**TITLE:** BATCH-BOT SMTP MAIL [V10.18.05]  
**DATE:** 6/27/2018  
**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 this splash help screen
```

\*\*\* HNDMAKEHZO.APP \*\*\*

**NAME:** HNDMAKEHZO.APP  
**TITLE:** BATCH-BOT HZO MAKER [V10.18.05]  
**DATE:** 6/27/2018  
**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 wth Build or Extract
--pass [OPT] Encrypting password
--query [OPT] CHT Query to describe files
```

\*\*\* HNDSEARCH.APP \*\*\*

**NAME:** HNDSEARCH.APP  
**TITLE:** BATCH-BOT SEARCH TEXT [V10.18.05]  
**DATE:** 6/27/2018  
**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.

### \*\*\* HNDSLFCMD.APP \*\*\*

**NAME:** HNDSLFCMD.APP  
**TITLE:** BATCH-BOT HTTP TOOL [V10.18.05]  
**DATE:** 6/27/2018  
**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, a Windows batch file or 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  
[CFG] Set up HTTP configuration information #0.

--config1  
[CFG] Set up HTTP configuration information #1.

--config2  
[CFG] Set up HTTP configuration information #2.

--config3 -o --config6  
[CFG] Set up HTTP configuration information #3 to #6

#### SET COMMANDS:

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

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

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

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

--set3 to --set6  
[CFG] Select and use configuration #3 to #6.

--set  
[CFG] Manually inserted url and port, ignore config settings.  
E.G. --set http://www.cwhandy.org:9000/

--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] Names the remote directory with which to interact.  
Remote directory always rooted in server /run/ directory.  
Example usage: subdir1/subdir2/  
Defaults to the server /run/ 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 hndslfcmdhelp.txt which contains the contents of  
the same splash screen.

--video

[OPT] Displays a video or a document from the web which  
further explains use of the application.

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 [V10.18.05]  
**DATE:** 6/27/2018  
**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

--noedit  
[OPT] --noedit requires no parameter use this for "silent" running so that Snap Edit does not appear afterward

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: SNAP-IN C#

\*\*\* CHTSNAP2PDF.EXE \*\*\*

**NAME:** CHTSNAP2PDF.EXE  
**TITLE:** CHT Snap Edit CMD Utility -- Version: 21.18.01.00  
**DATE:** 6/27/2018  
**CATEGORY:** SNAP-IN C#

**DESCRIPTION:**

CHTSNAP2PDF.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.

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*, *HNDPREVIEWER.APP* and *HNDTPXHTNEXT.APP*. These applications are described here:

[HNDDOCUMENTBUILDER.APP](#).

Contact CHT at: [support@cwandy.ca](mailto: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: 22.18.01.00  
**DATE:** 6/27/2018  
**CATEGORY:** SNAP-IN C#

## DESCRIPTION:

CHTSNAPEDIT.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.

CHTSNAPEDIT.EXE (AKA: CHT Snap Edit), is a command-line text editor complete with syntax highlighting and colorization. The editor is optimized for use with Clarion code, HTML and CSS. CHTSNAPEDIT.EXE may be used from the DOS prompt, in batch file operations or it may be incorporated into your Clarion application via CHT templates provided with the CHT toolkit.

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: 22.18.01.00  
**DATE:** 6/27/2018  
**CATEGORY:** SNAP-IN C#

## DESCRIPTION:

CHTSNAPGET.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.

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. 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 call into a specific server, and subdirectory, a file may be obtained from there with a single --fetch command.

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.

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 --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 (HNDNAPCALL.INC/CLW).  
To configure from the DOS command line, type:

**chtsnapget --config**

Then, complete the provided dialog values.  
CHT Snap Get -- Version: 20.16.03.00

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>

\*\*\* **CHTSNAPMAPI.EXE** \*\*\*

**NAME:** CHTSNAPMAPI.EXE  
**TITLE:** CHT Snap MAPI CMD Utility -- Version: 22.18.01.00  
**DATE:** 6/27/2018  
**CATEGORY:** SNAP-IN C#

**DESCRIPTION:**

CHTSNAPMAPI.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.

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 "TO" "SUBJECT" "MSG.TXT OR MSG.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.  
MSG.TXT, MSG.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.

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>

\*\*\* **CHTSNAPSEND.EXE** \*\*\*

**NAME:** CHTSNAPSEND.EXE  
**TITLE:** CHT Snap SEND CMD Utility -- Version: 22.18.01.00  
**DATE:** 6/27/2018  
**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@cwandy.ca](mailto:support@cwandy.ca) for price and availability information.

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

### \*\*\* CHTSNAPSMTP.EXE \*\*\*

**NAME:** CHTSNAPSMTP.EXE  
**TITLE:** CHT Snap SMTP CMD Utility -- Version: 22.18.01.00  
**DATE:** 6/27/2018  
**CATEGORY:** SNAP-IN C#

#### DESCRIPTION:

CHTSNAPSMTP.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.

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 "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 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.  
\*\* CHTSNAPSMTP.EXE \*\* Version: [20.16.03.00] \*\*

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: 22.18.01.00  
**DATE:** 6/27/2018  
**CATEGORY:** SNAP-IN C#

**DESCRIPTION:**

CHTSNAPZIP.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.

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.

Zip That Recurses Subdirectories Accepts 3 Inputs:  
chtsnapzip  
Folder names are included in the zip.

Zip That Skips Subdirectories Accepts 3 Inputs:

**chtsnapzip "zz" "zipfilename" "sourcedir[\mask]"**

Produces a flat zip without the folder names in the zip.

Unzip Accepts 3 Inputs:

**chtsnapzip "uz" "zipfilename" "targetdir[\mask]"**

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.  
[\mask] Represents an optional mask [e.g. \hnd\*.clw] to filter files.

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>

**NAME:** HNDCLEAN.EXE  
**TITLE:** CHT Clarion Gen-Code Cleaner -- Version: 22.18.01.00  
**DATE:** 6/27/2018  
**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](mailto: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:  
 TOOLKIT UTILITY**

**NAME:** HNDBASE64CVT.APP  
**TITLE:** Base64 File Command Utility [V10.18.05]  
**DATE:** 6/27/2018  
**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.
- video [OPT] Displays a help vidoe or help document from the web. .
- update [OPT] Causes the application to self-update from CHT servers. Utilizes CHTSNAPGET.EXE if / when invoked.
- 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

**NAME:** HNDRUNBAT.APP

**TITLE:** Batch File Runner Utility [V10.18.05]  
**DATE:** 6/27/2018  
**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 [REQ] Name of batch file to run, without path  
--path [REQ] Path of batch file to run  
--log [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.

\*\*\* HNDSHCUT.APP \*\*\*

**NAME:** HNDSHCUT.APP  
**TITLE:** Shortcut Command Utility [V10.18.05]  
**DATE:** 6/27/2018  
**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"

\*\*\* HNDSTOPBATCH.APP \*\*\*

**NAME:** HNDSTOPBATCH.APP



**TITLE:** Batch File Stop Utility [V10.18.05]  
**DATE:** 6/27/2018  
**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.

## CATEGORY: XML2HTML

\*\*\* HNDTILEAPPLAUNCHER.APP \*\*\*

**NAME:** HNDTILEAPPLAUNCHER.APP  
**TITLE:** XML2HTML Tiles Demo [V10.18.05]  
**DATE:** 6/27/2018  
**CATEGORY:** XML2HTML  
**CLASSES:** [List App Classes](#)  
**TEMPLATES:** [List App Templates](#)

**DESCRIPTION:**

It launches this hndtileapplaunchermenus.html automatically for purposes of demonstration but changing the HTML file being displayed is trivial. Display files can even be changed via hot-key CTRL-F and selecting a new tile-menu file from \accessory\hnd\html\.

The example HTML file hndtileapplaunchermenus.html is created with CHTSNAPEDIT.EXE and XML style sheet 000tilemenuexample\$\$\$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, while an example could easily be saved under a new name to become your own app-launcher by changing the name of the launched HTML file and developing new functionality via CHTSNAPEDIT.EXE from a renamed copy of 000tilemenuexample\$\$\$XML and modifying the tile menu links.

\*\*\* HNDXML2HTMLGEN.APP \*\*\*

**NAME:** HNDXML2HTMLGEN.APP  
**TITLE:** XML2HTML Command Utility [V10.18.05]  
**DATE:** 6/27/2018  
**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  
--prevw [OPT] preview HTML after generate  
--noprevw [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.

**CHT HTML Document Builder**  
© 1996-2018  
**The Clarion Handy Tools Page**