

## CodePort Utility Program Flow

1. CodePort Option Entry form gathers the following:
  - source version (Addon version 6 or 7)
  - location of tokenized source files
  - location for new source files
  - extension for new source files
  - number of arguments (option!.getOption() placeholders) to pass to source file
  - checkbox to allow/deny overwrite without prompting
  - builds grid of files in source directory
  - builds vector of files selected in grid (note: stored with callpoint!.setDevObject(), this vector drives the main processing loop)
2. Option Entry form runs `adx_codeport.bbj`
  - a. retrieves user input from Option Entry form
  - b. other miscellaneous initializations
  - c. creates/opens log file
  - d. creates progress meter
  - loads `replacements.ini` flat file into string arrays (note: `replacements.ini` is used for v7 text replacements; if user is converting v6 code, program will switch between `replacements_oldport.ini` and `replacements.ini` as it processes each file)
  - next overlay is `adx_codeportfile.bbj`
3. Overlay `adx_codeportfile.bbj`
  - a. top of processing loop; gets next file to convert
  - b. if done, runs `adx_codeportexit.bbj`
  - c. prompt for overwrite if override not checked on Option Entry form
  - d. if converting version 6, next overlay is `adx_oldport.bbj`
  - e. if converting version 7, goes directly to `adx_codeportconv.bbj`
4. Overlay `adx_oldport.bbj`
  - a. calls "`_label`" utility on incoming tokenized v6 file
  - b. saves that file as text/source using `bbjlst`
  - c. switch to `replacements_oldport.ini` file and perform text replacement
  - d. make further text modifications to bring v6 code to v7 standards:
    - change header lines (10 and 20)

- change call list on some programs
- add progress meter call instead of notification boxes
- changes to on V3 line
- change FNF\$ function
- change data file names within the source code
- convert data file names of AR to OP
- rename program names within the source code
- insert standard v7 code blocks (std\_input, std\_error, etc.)
- add standard labels (FLOW\_CONTROL, ITIAC, etc.)
- e. recompile the modified source code using bbjcpl
- f. switch back to replacements.ini
- g. next overlay is adx\_codeportconv.bbj

## 5. Overlay adx\_codeportconv.bbj

- coming from adx\_oldport.bbj if converting v6 files, incoming tokenized program is xxx\_xx.lbl
- coming from adx\_codeportfile.bbj if converting v7 files, incoming tokenized program is xxx\_xx.bbx
  - call "\_label" utility on incoming v7 file
- main processing given xxx\_xx.lbl file, done either in this program, or with calls to adc\_codeportconv.bbj
  - i. convert program to source listing; uses bbjlst command with -l switch to remove line numbers
  - ii. loads listed source from previous step into lines\$[ ] array
  - iii. convert each line to lower case and write lines\$[ ] to new string file
  - iv. replace standard sections with temporary #include's; clears out standard routines, leaves #include placeholders (note: any code beyond the "end" statement is preserved in custom\_lines! vector)
  - v. replace standard date handing functions
  - vi. format comments; standardizes comment format, inserts whitespace before/after
  - vii. update copyright rems
  - viii. remove any statements beginning with a reset verb
  - ix. move labels; puts labels on their own lines, except for rem
  - x. process simple LET statements; parses LET statements w/ multiple assignments into separate lines
  - xi. indent code as necessary; indents for/next, while/wend loops
  - xii. store IOLIST's; stores iolists in a string array
  - xiii. check code for channel references not using an IOLIST ; adds rem to beginning of new source file w/ info

- xiv. replace IOLIST references with read record syntax ; changes  
read(x)iol=abc\$ with readrecord(x)abc\$
- xv. remove IOLIST statements
- xvi. combine labels
- xvii. remove THEN clause from the IF statements
- xviii. further indent all code per newer standards
- xix. search and replace code as indicated in replacements.ini
- xx. replace #include's with actual code
- xxi. insert program initialization code; inserts standard  
initialization and file open routines
- xxii. remove any extra blank lines
- a. next overlay is adx\_codeportlbls.bbj

#### 6. Overlay adx\_codeportlbls.bbj

- a. analyze labels, replacing with symbolic labels (i.e., \*next) where possible, and removing unreferenced labels
- b. next overlay is adx\_codeporterrs.bbj

#### 7. Overlay adx\_codeporterrs.bbj

- o puts explanatory rems at beginning of new source
- o creates new string file w/ target name (i.e., xxx\_xx.aon) and prints lines\$[ ] array to it
- o prints any custom code from custom\_lines! vector to the target file
- o removes xxx\_xx.lbl and xxx\_xx.lst files used during "\_label" and bbjlst
- o back to adx\_codeportfile.bbj for next file (or quit).