**Programmer’s Manual**

**MPX**

**Version 1.0**

**Updated: 9-7-2013**

**Table of Contents**

Overview of MPX and Commhand.................................................................................................3

Summary of Contents of Files.........................................................................................................3

Date File..........................................................................................................................................3

Version File.....................................................................................................................................4

Help File..........................................................................................................................................4

Exit File...........................................................................................................................................5

LS File.............................................................................................................................................5

Common File...................................................................................................................................6

Commhand File...............................................................................................................................6

Main File.........................................................................................................................................7

**Overview of MPX and Commhand (Probably needs more info)**

MPX is for the user to be able to access information as they needed it. The principal need for command is to loop until the user enters the exit command. While the user is still using MPX, the commhand will print the prompt for the user to input some command, then the command will be match to the function and the function will be executed (with error handling).

**Summary of Contents of Files**

* **Date** – Functions: date, isLeapYear, checkDays
* **Version** – Functions: version
* **Help** – Functions: help, displayHelp
* **Exit** – Functions: exit
* **LS** – Functions: ls
* **Common** – Functions: InvalidArgs
* **Commhand** – Functions: commhand, tokenize
* **Main** – Functions: main

**Date File**

**Functions and parameters** – The function, date, will display the current system date or will change the current system date depending of the user input. The parameters for date are: int argc and char \*\*argv. The function, isLeapYear, will check to see if the current system date’s year is a leap year or not. The parameter for isLeapYear is int year. The function, checkDays, will check to see if the current system date is a valid date or not. The parameters for checkDays are: int month, int day, int year.

**Structures and Variables** – The function, date, has these structures and variables: date\_rec \*today, char buffer, int bufferSize, int month, int day, int year, int scanReturn. The function, isLeapYear, has these structures and variables: int returnVal. The function, checkDays, has these structures and variables: int returnVal, int bufferSize, char buffer[].

**Cross References** – The function, date, is called from the function commhand. The variables and structures in date are defined and used only in date. The function, isLeapYear, is called from the function date. The variables and structures in isLeapYear are defined and used only in isLeapYear. The function, checkDays, is called from the function date. The variables and structures in checkDays are defined and used only in checkDays.

**Version File**

**Functions and parameters** – The function, version, will display what current version MPX is in. The parameters for version are: int argc, char \*\*argv.

**Structures and Variables** – The function, version, has these structures and variables: char buffer[], int bufferSize.

**Cross References** – The function, version, is called from the function commhand. The variables and structures in version are defined and used in only version.

**Help File**

**Functions and parameters** – The function, help, will display either one or all of the commands and what they mean. The parameters for help are: int argc, char \*\*argv. The function, displayHelp, will print out the commands and what they are. The parameters for displayHelp are: int argc, char \*command.

**Structures and variables** – The function, help, has no variables or structures. The function, displayHelp, has these variables and structures: FILE \*fptr, char file[], char cReturn[], char invalidCommand[], char \*buffer, int bufferSize, int invCommandSize, int cReturnSize.

**Cross References** – The function, help, is called from the function commhand. The function, displayHelp, is called from the function help. The variables and structures in displayHelp are defined and used only in displayHelp.

**Exit File**

**Functions and parameters** – The function, exit, will end the use of MPX at the user discretion. The parameters for exit are: int argc, char \*\*argv.

**Structures and variables** – The function, exit, has these variables and structures: char confirm[], char ans[], int confirmSize, int maxSize, int yes, int response, int returnVal.

**Cross References** – The function, exit, is called from commhand. The variables and structures in exit are defined and used only in exit.

**LS File**

**Functions and parameters** – The functions, ls, will display the files and sub-directories in that current directory. The parameters for ls are: int argc, char \*\*argv.

**Structures and variables** – NEED MORE INFO

**Cross References –** The function, ls, is called from commhand. The variables and structures in ls are defined and used only in ls.

**Common File**

**Functions and parameters** – The function, InvalidArgs, checks to see if the command entered is valid or not. The parameter for InvalidArgs is char \*command.

**Variables and Structures –** The function, InvalidArgs, has these variables and structures: int bufferSize, char buffer[].

**Cross References** – The function, InvalidArgs, is called from the functions: date, exit, help, ls, version, and commhand. The variables and structures in InvalidArgs is defined and used only in InvalidArgs.

**Commhand File**

**Functions and parameters** – The function, commhand, reads the user’s command and takes the user to the right function based on the entered command. There are no parameters for commhand. The function, tokenize, takes the user input and separates the commands (based on spaces and tabs) and stores the commands. The parameters for tokenize are: int \*argc, char \*argv[], char \*input).

**Structures and variables** – The function, commhand, has these variables and structures: int maxSize, int promptSize, int i, int invalidCommandSize, int exitSize, char buffer[], char prompt[], char invalidCommand[], char exitMessage[], int (\*function[NUM\_COMMANDS] (int,char\*\*), char commands [NUM\_COMMANDS][], int argc, char \*argv[], int repl. The function, tokenize, has these variables and structures: char \*token, int tempArgC.

**Cross References** – The function, commhand, is called from the function main. The variables and structures in commhand are defined and used only in commhand. The function, tokenize, is called from the function commhand. The variables and structures in tokenize are defined and used only in tokenize.

**Main File**

**Functions and parameters** – The function, main, runs MPX. There are no parameters for main.

**Structures and variables** – There are no structures or variables for main.

**Cross References** – There are no variables or structures that are defined and the function is not called from another function.

**Index**

Commhand 3, 6-7

Common 3, 6

Date 3-4

Exit 3, 5

Help 3-5

LS 3, 5

Main 3, 7

MPX 3

Version 3-4