Parallax Propellent

The Parallax Propellent software is a Windows-based tool for compiling and downloading to the Parallax Propeller chip. Propellent is available as both a library (Propellent.dll) and as an executable (Propellent.exe).

- The Propellent Library (DLL) is for software developers to link into applications enabling immediate support of the Propeller using the same functions as the Parallax-made Propeller Tool development software.
- The Propellent Executable (EXE) is a program that includes the Propellent Library within it and provides many of the same functions to anyone wishing for command-line support of the Propeller chip.

This document is written for the Propellent Executable. For details about the Propellent Library, see the "Propellent Library.pdf" document. For more information on the Parallax Propeller chip and tools, please visit http://www.parallax.com/propeller

Features of the Propellent Executable:

- Small, command-line driven application that includes the Propellent Library inside of it; there's no need to have the Propellent.dll file in order to use Propellent.exe.
- Compiles and downloads Propeller source (.spin), and downloads Propeller Application images (.binary or .eeprom) to Propeller chips.
- Allows saving of compiled source as a binary or EEPROM image.
- Can run with full, limited, or no visual feedback and messages can be redirected to an optional message file.
- Includes the Propeller Tool's multi-threaded serial port handling and Propeller chip communication functionality.
- Includes the Propeller Tool's dialogs for indicating serial port access and download progress as well as the user-customizable serial port search options.
- Stores most user-modified preferences in the Windows Registry for use in future sessions; Source Library path, Serial Search Method, Reset Signal, and Graphical User Interface mode.
- Supports Win2K (and later) operating systems.

New in Propellent Executable v1.6

For users of the previous Propellent Executable (v1.5):

1) Enhanced Propellent Executable v1.6 to set its exit code to the last message ID value. Previous versions often exited with a code of 0 which wasn't helpful if trying to determine the exit status without parsing the output string. Now the exit status matches the message ID of the last output string and can be accessed programmatically as the value returned from Propellent's process termination, or via the % ERRORLEVEL% variable within a batch file.

For users of the previous Propellent Executable (v1.4):

1) Fixed bug in Propellent Executable v1.5 that output or displayed a source line from the wrong file when a compile error was caused by a missing object file.
2) Enhanced dialog display to make source line, and error message, more clear.

For users of the previous Propellent Executable (v1.3):

1) Propellent v1.4 now includes a more robust handshake algorithm (like that of Propeller Tool v1.3.2) for connecting with boards like the Propeller QuickStart board.

For users of the previous Propellent Executable (v1.2):

1) Propellent now defaults to maintaining a localized set of options based on its version number and the filesystem folder from which it was executed. Any other instances of Propellent will use their own settings so as not to interfere. To force global behavior, as was the default behavior for versions previous to Propellent v1.3, use the /GLOBAL switch, described below.
2) New switch, /GLOBAL, causes Propellent to use settings defined globally on the system; ie: by other instances of Propellent that may have been executed from other folder paths. See details in next section.
3) New switch, /PORTRULES, allows setting and retrieving serial port search rules via command-line; an alternative to using the graphical interface via the /EDITPORTS switch. See details in following sections.

Propellent Executable Command-Line Options

Syntax Definitions:

Load source/image and download to Propeller chip.

```
Propellent.exe {/GLOBAL} {/LIB lib_path} {/PORT (AUTO | COM#)} {/PORTRULES search_rules} {/SIGNAL (DTR | RTS | BOTH)} {/GUI (ON | STATUS | DIALOGS | OFF)} {/EEPROM} propeller_file {> message_file}
```

Load source, compile, and optionally save image.

```
Propellent.exe {/GLOBAL} /COMPILE {/LIB lib_path} {/GUI (ON | STATUS | DIALOGS | OFF)} {( /SAVEBINARY | /SAVEEEPROM ) {/IMAGE image_file}} source_file {> message_file}
```

Show the Serial Port Search List to allow editing.

```
Propellent.exe {/GLOBAL} /EDITPORTS
```

Return current persistent option(s).

```
Propellent.exe {/GLOBAL} /GET ( /LIB , /PORT , /PORTRULES , /SIGNAL , /GUI ) {> message_file}
```

Set persistent option(s).

```
Propellent.exe {/GLOBAL} /SET ( /LIB lib_path , /PORT (AUTO | COM#) , /PORTRULES search_rules , /SIGNAL (DTR | RTS | BOTH) , /GUI (ON | STATUS | DIALOGS | OFF) ) {> message_file}
```

Identify the Propeller chip.

```
Propellent.exe {/GLOBAL} /ID {/PORT (AUTO | COM#)} {/PORTRULES search_rules} {/SIGNAL (DTR | RTS | BOTH)} {/GUI (ON | STATUS | DIALOGS | OFF)} {> message_file}
```

Display version of Propellent library/executable.

```
Propellent.exe {/GLOBAL} /VERSION {/GUI (ON | STATUS | DIALOGS | OFF)} {> message_file}
```

Display this information.

```
Propellent.exe /HELP
```

Syntax Elements:

{ } – Denotes optional parameters; do not type the braces.
(|) – Denotes mutually exclusive parameters; exactly one item must be specified; do not type the parentheses or pipe symbol.
(,) – Denotes necessary parameter(s); one or more items must be specified; do not type the parentheses or comma symbol.

lib_path – Specifies a path to Propeller library files.
search_rules – Specifies the serial port search rules used to determine which ports are available to Propellent. See "Search Rules," below, for more information.
propeller_file – Specifies Propeller source or image file (.spin, .binary, or .eeprom) to be loaded, compiled if necessary, and downloaded to the Propeller chip.
message_file – Specifies a file to write status/info messages to; see "Messages," below. Messages that normally appear visually will be written to the file in text format. Use /GUI to hide visual messages.
image_file – Specifies path and filename (.binary, or .eeprom) to save the Propeller image to.
source_file – Specifies Propeller source file (.spin) to be loaded and compiled.

/COMPILE – Compile and return error, if any.
/EDITPORTS – Display the Serial Port Search List for possible editing. Port filtering rules and search order may be modified. Settings are saved in between sessions.
/EEPROM – Download to Propeller chip's EEPROM in addition to its RAM.
/GET – Retrieve options of all the switches that follow; see /LIB, /PORT, /SIGNAL, and /GUI. Operates globally on all switches on the command-line.
/GLOBAL – Read/write options globally for other Propellent instances to use. Without this switch, default behavior saves options in the regristry for the specific instance of Propellent, based on the version number and the filesystem folder from which it executed.
/GUI – Refers to the Propellent Graphical User Interface (GUI) preference.
  /GUI (ON | STATUS | DIALOGS | OFF) – Sets the GUI preference to show all messages, status messages only, dialogs only, or no messages; valid for this session only.
  /GET /GUI – Returns the current GUI preference; stored in between sessions.

/SET /GUI (ON | STATUS | DIALOGS | OFF) – Sets the GUI preference to show all messages, status messages only, dialogs only, or no messages; stored in between sessions.
– ON (default) indicates to show all status and dialog messages visually.
– STATUS indicates to show only status messages visually.
– DIALOGS indicates to show only dialog messages (user prompts) visually.
– OFF indicates to hide all visual messages during normal operation.
Note: the GUI setting is ignored for:
1) the /EDITPORTS function since it requires GUI interaction, and
2) rare-occurrence user prompts such as the "File already exists. Overwrite?" dialog.

/HELP – Display command-line help (this information). /? is equivalent.

/ID – Identify a Propeller chip connected to a serial port.

/IMAGE – Refers to the path and filename to save compiled source to. This option, if specified, must be used with /SAVEBINARY or /SAVEEEPROM. Without /IMAGE, the compiled image is saved with the same path and filename as the source_file, but with a .binary or .eeprom extension. If /IMAGE is specified with a trailing path delimiter, \, the image will be saved in that path with the original source filename and the appropriate extension. If /IMAGE is specified ending with a non-path delimiter, the image will be saved using that path and filename; any given extension will be replaced with .binary or .eeprom. Any folders in the path string that do not exist will be created automatically. This option is not persistent; it can not be used with /GET and /SET.
/IMAGE *image_file* – Sets the compiled image path, and optionally the filename, to image_file; valid for this session only.

/LIB – Refers to the Propeller source library path.
/LIB *lib_path* – Sets the path to lib_path; valid for this session only.
/GET /LIB – Returns the current path; stored in between sessions.
/SET /LIB *lib_path* – Sets the path to lib_path; stored in between sessions.

/PORT – Refers to the serial port search method.
/PORT (AUTO | COM#) – Sets the port search method to AUTO or COM#; valid for this session only.
/GET /PORT – Returns the current port search method; stored in between sessions.
/SET /PORT (AUTO | COM#) – Sets the port search method to AUTO or COM#; stored in between sessions.
– AUTO (default) indicates to search all available serial ports (according to port filtering and search order rules; see /EDITPORTS) for a Propeller chip.
– COM# indicates to search a specific serial port for a Propeller chip, ignoring all others. # must be one or more numeric digits.

/PORTRULES – Refers to the serial port search rules.
/PORTRULES *search_rules* – Sets the port search rules to that of the *search_rules* string; valid for this session only.
/GET /PORTRULES – Returns the current port search rules; stored in between sessions.
/SET /PORTRULES *search_rules* – Sets the port search rules to that of the *search_rules* string; stored in between sessions.
Note: Invalid rules will be removed or reformated to create a valid *search_rules* string for that session or setting. See the "Search Rules" section, below, for the format of the *search_rules* string.

/SAVEBINARY – Save successfully compiled source code as a binary file. Can be combined with /IMAGE to specify output file name.

/SAVEEEPROM – Save successfully compiled source code as an EEPROM file. Can be combined with /IMAGE to specify output file name.

/SET – Set options of all the switches that follow; see /LIB, /PORT, /SIGNAL, and /GUI. Operates globally on all switches on the command-line.

/SIGNAL – Refers to the serial port signal used to reset the Propeller chip.
/SIGNAL (DTR | RTS | BOTH) – Sets the reset signal to DTR, RTS, or both; valid for this session only.
/GET /SIGNAL – Returns the current reset signal; stored in between sessions.
/SET /SIGNAL (DTR | RTS | BOTH) – Sets the reset signal to DTR, RTS, or both; stored in between sessions.
– DTR (default) indicates to reset the Propeller chip using the serial port's DTR signal.
– RTS indicates to reset the Propeller chip using the serial port's RTS signal.
– BOTH indicates to reset the Propeller chip using the serial port's DTR and RTS signals.

/VERSION – Retrieve version of the Propellent library and this executable.

Examples:

- To search for and identify a Propeller chip connected to the system:
  Propellent.exe /id

- To perform the above operation and generate a message file, called 'results.txt':
  ```
  Propellent.exe  /id > results.txt
  ```

- To perform the above operation without any visual feedback:
  ```
  Propellent.exe  /id  /gui  OFF  > results.txt
  ```

- To download image file "Graphics_Demo.binary" to Propeller RAM from the current folder using current settings:
  ```
  Propellent.exe  Graphics_Demo.binary
  ```

- To compile and download application "Graphics_Palette.spin" to Propeller's RAM and EEPROM from the C:\Temp folder using current settings:
  ```
  Propellent.exe  /eeprom  C:\Temp\Graphics_Palette.spin
  ```

- To compile and download application "Inductor Demo.spin" to Propeller RAM (on serial port 2) from the default library examples folder:
  ```
  Propellent.exe  /port  COM2  "C:\Program Files\Parallax Inc\Propeller Tool v1.1\Examples\Library\Inductor Demo.spin"
  ```

- To compile and save an EEPROM image of the above application:
  ```
  Propellent.exe  /compile  /saveeeprom  "C:\Program Files\Parallax Inc\Propeller Tool v1.1\Examples\Library\Inductor Demo.spin"
  ```

- To see the current persistent settings for Propeller Source Library and Serial Port Search Method:
  ```
  Propellent.exe  /get  /lib  /port
  ```

- To change the persistent settings for Serial Port Search Method and Reset Signal to COM25 and BOTH, respectively:
  ```
  Propellent.exe  /set  /port  COM25  /signal  BOTH
  ```

- To view the current system serial ports and modify search order and port filtering rules:
  ```
  Propellent.exe  /editports
  ```

Search Rules:

The optional *search_rules* field of the **/PORTRULES** switch can be used to specify the rules determining which ports Propellent uses. It is recommended to let the user the graphical interface to specify the rules, via the **/EDITPORTS** switch, however, it may be handy, or necessary, to specifying them programmatically via **/PORTRULES** instead.

The *search_rules* field is a single string containing a set of comma-delimited rules.

- The most common rules begin with a + or - character to indicate whether the rule causes matching ports to be included (+) or excluded (-). Numbers or letters typically follow the leading + or - to indicate the rule's match pattern.
- For rules with a leading + or -, a wildcard character, *, can be used to match zero or more characters starting at that position of the rule. The wildcard character can occur anywhere within the rule after the leading + or - character.
- Rules with a leading + or - followed only by one or more digits indicate to include or exclude a port by its numeric ID. Numeric rules may by encased in parentheses (and) to indicate their search order, relative to each other.

A special global rule to "include all," or "exclude all," ports by default is indicated simply with a + (to include) or - (to exclude) followed by an *. For example, the rule +* tells Propellent to include all ports by default, except where excluded by later rules. Similarly, the rule -* tells it to exclude all ports by default, excepts where included by later rules.

Example Rules:

+* This is a single rule causing all serial ports to be included for use by Propellent.

+*,-*Bluetooth* There are two rules here, the first (before the comma) includes all ports by default, and the second excluding any ports containing the characters "Bluetooth" within their name.

+*,(3),(4),(-1) These three rules include all ports by default and specify to search port COM3 before COM4, and also before COM1 (which is also excluded due to the leading -). Any other ports, like COM2 for exame, would be searched after COM3 and COM4 (and COM1, if it were actually included).

Tips:

- If a rule needs to include spaces, enclose the entire rule string in quotes.
- To understand or find rule strings for your needs, it may be best to use the **/EDITPORTS** switch with graphically define the rules, then click Accept, then use the **/GET /PORTRULES** switch combination to see the resulting rule string.

Messages:

The following describes the Propellent Executable messages and format that is sent to standard out, or written to a file if the optional redirection option (**>** *message_file*) is used. This information can be ignored unless you're intending to either log the results or somehow interface to the Propeller Executable using custom software.

There are three types of messages:

1) Fatal Errors - indicate a problem that caused the Propellent Executable to terminate.
2) Information - indicates final status, just before termination.
3) Events - indicates intermediate status that will be followed by other messages.

Each message has two parts:

1) Message header - Fixed size (9 character) text indicating the type of message and its unique ID.
2) Message body - Variable size text containing the point of the message.

The message header consists of:

Character 1 - [Start of Header]; SOH character (ASCII value 1); this provides an easy "start of new message" character for software to parse out of the message stream.
Characters 2 - 5 - [Type]; 'ERR:', 'INF:', or 'EVT:' meaning Fatal Error, Information, or Event, respectively.
Characters 6 - 9 - [ID]; 'nnn-' where nnn is a 3-digit unique ID of the message.

The final message is terminated by an End of Transmit (EOT) character (ASCII value 4).

Below is a list of all message IDs and respective messages.

FATAL ERRORS (final messages)

~~~command-line errors~~~

001 - Invalid command-line switch: 'SWITCH'.

002 - Duplicate command-line switch: 'SWITCH'. Each command-line switch should only appear once.

003 - Illegal command-line switch: 'SWITCH'. This switch is not allowed in combination with the switches that appear before it.

004 - Too many command-line options.

005 - Missing parameter required. REQUIREMENTS.

006 - Invalid parameter: REQUIREMENTS.

007 - File needs an extension: SWITCH

008 - File not found: SWITCH

009 - Sorry, syntax help is missing.

010 - Library path not found: SWITCH

011-050 - (reserved)
~~~

```
~~~file errors~~~
051       - Image file save cancelled by user.
052       - <image file corrupt details.>
053       - Unable to save image file: 'FILENAME'.
054-100- (reserved)
~~~compiler errors~~~
101       - Error compiling file: 'FILENAME'  (Line/Char/Count: LINE/CHAR/COUNT)  ERRORMESSAGE  SOURCELINE
102-300- (reserved)
~~~communication errors~~~
301       - No Propeller chip found on any serial port.  SCAN_STATUS.
302       - No Propeller chip found on serial port.  SCAN_STATUS.
303       - No serial ports found.
304       - No usable serial ports available.
305       - Serial port COM# does not exist or is excluded.  Unable to scan for Propeller chip.
306       - Propeller chip lost on COM#.
307       - Read failure on COM#.
308       - Write failure on COM#.
309       - Unable to create I/O event object.
310       - Unable to complete I/O.
311       - Broken Pipe: Invalid Pipe Handle.  (NOTE: This message can not be seen, but application's exit code will be 311)
312-350- (reserved)
~~~device errors~~~
351       - RAM checksum error on COM#.
352       - EEPROM programming error on COM#.
353       - EEPROM verify error on COM#.
354-400- (reserved)
```

INFORMATION (final messages)

```
~~~file messages~~~
401       - Image file saved successfully.
402       - File(s) compiled successfully.
403-450- (reserved)
~~~communication messages~~~
451       - Success.  <indicates successful identifying/programming>
452-475- (reserved)
~~~configuration messages~~~
476       - Propellent Executable:  VERSION  Propellent Library:  VERSION
477       - PERSISTENT_SETTINGS
478-500- (reserved)
```

EVENTS (intermediate messages; will be followed by further messages from Propellent.exe program)

```
~~~communication events~~~
501       - Compiling: FILENAME.
502       - File(s) compiled successfully.
503       - Checking COM#.
504       - Scanned COM#.
```

505 - Propeller chip version # found on COM#.
506 - Loading ram.
507 - Verifying ram.
508 - Programming eeprom.
509 - Verifying eeprom.
510 - Unable to open COM#.
511 - Unable to configure buffer on COM#.
512 - Unusable COM#.
513 - Unable to read from COM#.
514 - Unable to write to COM#.
515-600 - (reserved)