

Change Log — GPPG Scanner Generator

(Version 1.5.0 May 2012)

John Gough QUT

May 11, 2012

Documentation Map

This paper is the changelog for the Gardens Point Parser Generator (*gppg*). The complete documentation for *gppg* consists of the following files —

- * *Gppg.pdf* – the main documentation file.
- * *Gppg-Changelog.pdf* – change log for *gppg* (this file).

1 Change Log

This section tracks the updates and bug fixes from version “0.9” of July 2006. Changes prior to that version were the addition of additional markers in the grammar files, and removing some runtime failures.

Changes in version 1.5.0 (May 2012)

- * Error messages are now in *MSBuild*-friendly format to interwork better with *Visual Studio*. This is the default behavior, with a new option allowing legacy messages to the console.
- * The format of the html output with the `/report` option has been enhanced to make tracing of the automaton easier. The format of the tracing output for parsers generated with the `TRACE_ACTIONS` symbol has also been improved.
- * This version changes the default file encoding to Unicode, with a fallback to raw (uninterpreted bytes) if there is no *BOM* prefix.
- * When *gppg* is invoked with the `/gplex` option a *ScanObj* class is defined that encapsulates symbol information.
- * Additional features have been added to *gppg* and to *gplex* to support symbol pushback for cases where a backtracking automaton is required.
- * Some new options allow two parsers to share the same token definition, including token associativity and precedence. One parser specification defines the token enumeration and serializes the token dictionary. Client parser specifications

deserialize the dictionary rather than defining their own token enumeration. This feature allows multiple parsers to interleave consumption of tokens from a shared scanner.

Changes in version 1.4.6 (January 2012)

- * The distribution has been built against the V4 .NET framework, and the project subdirectory contains a VS-2010 project. However, the sources will build with VS-2008 and the V2 framework, if necessary.
- * The *RealCalc* example has been corrected to work with current versions of *gppg*. The new example demonstrates how to define a constructor for the parser class.
- * An error in the handling of literal characters and strings has been fixed. Previously strings with an escaped backslash as their last character were incorrectly handled.
- * The accessibility of the *NextToken* field in the abstract scanner base class has been changed to *protected*. This allows for those rare cases where a parser action is required to overwrite the current value. Such an overwriting is an idiom of some techniques for resolving syntactic ambiguity by performing an *ad hoc* lookahead. See the December 2011 entries in the blog *John Gough on Software Tools* — <http://softwareautomata.blogspot.com>

Changes in version 1.4.5 (April 2011)

- * Fixed an error in the *gppg.y* file that disallowed declarations of semantic types including generics of more than one type parameter.
- * For convenience, the distribution includes a current GPLEX executable.

Changes in version 1.4.4 (January 2011)

- * *gppg* now returns an integer result, for the use of *msbuild* or other script engines.
- * The output filename may now be specified in a command line argument.
- * New command-line flags *line-filename*, *no-info*, *no-filename*, *noThrowOnError* have been added.
- * Semantic actions may now contain either *C#* verbatim literal strings or normal *C#* literal strings.
- * Several errors in the handling of literal strings in semantic actions have been fixed. Previously literal strings with “\$” or “@” characters were erroneously handled.
- * Literal strings terminated by end-of-line, or verbatim literal strings terminated by end-of-file get an explicit error message.
- * The second change reported below for V1.4.3 broke the format of the *#line* directive. This has now been fixed.

Changes in version 1.4.3 (November 2010)

- * The semantics of the /report option has been changed. Previously /report caused the generation of an html report *instead* of creating a parser source file. In this version the /report options creates the parser output file and produces the report *as well*.
- * The parser output file, the listing file and the html report file all identify the name *and last revision date* of the input specification “*.y” file.
- * An error in the html of the report file tripped up some browsers. This has been fixed.
- * Error reporting for unmatched brace errors have now been improved by identifying the unmatched left brace.
- * Error recovery within semantic actions has been improved.

Changes in version 1.4.2 (July 2010)

- * An error in the scanner definition for *gppg* has been corrected. *gppg* now correctly handles the character literal “” within semantic actions.

Changes in version 1.4.1 (April 2010)

- * The initialization of tables for parsers has been reorganized to make creation of parser instances faster. Multiple parser instances now share parsing tables.
- * *ShiftReduceParser* has a changed interface to allow table sharing among multiple parser instances. Previous parsers will continue to work with the new component, but new parsers **require** the 1.4.0 version of *ShiftReduceParser.dll*.
- * The previously undocumented feature for producing tracing parsers has been upgraded and documented.

Changes in version 1.3.6 (December 2009)

- * A number of small changes have been made to error recover in the tool. The new version is more robust in the face of erroneous input. The program now quits after 50 error messages in case of an input file of incorrect type.

Changes in version 1.3.5 (March 2009)

- * The parsers generated by *gppg* have been carefully restructured so as to be *Fx-Cop*-friendly. This has involved the reduction in visibility of some class members, and the renaming of others. *Some of these changes may be breaking for existing applications.*
- * The mechanisms for the instantiation of parser objects have been simplified and made safer. *This change may be breaking for existing applications.*

- * A new feature in the definitions section of the grammar allows overriding of the default naming of the scanner base class *ScanBase*. This class is used by *gplex*-generated scanners. The change simplifies the design of applications that include multiple scanners and parsers. New sections in the *gppg* and *gplex* documentation discuss the use of such renamings.
- * The *ShiftReduceParser* runtime component is now fully versioned, and has been systematically named as *QUT.ShiftReduceParser.dll*.
- * All source code of *ShiftReduceParser* component has been consolidated into a single file to allow source inclusion as an alternative to referencing the *DLL*. Default visibility of the parser class has been reduced to internal.

Changes in version 1.3.1 (November 2008)

- * The new frontend introduced in version 1.3.0 rejected some program constructs that were previously accepted. Version 1.3.1 corrects all such bugs that have been reported.
- * The v1.3.0 parser wrongly rejected productions, notifying an error for productions that started with an action, but were not empty production. This is now fixed.
- * The v1.3.0 parser wrongly rejected productions, notifying an error if a production had adjacent actions that were not separated by symbols. This is now fixed.
- * The *gppg* input grammar allows a precedence marker to be attached to mid-rule actions, although such a marker cannot have a semantic effect. The warning message that *gppg* issues in such cases has been clarified.

Changes in version 1.3.0 (October 2008)

- * The new release has a completely new frontend. The scanner and parser are constructed by *gplex* and *gppg* respectively.
- * Error handling is based around an error buffer and a listing generator, and the parser attempts error recovery. Most error messages have changed, and a number of syntactic errors that were previously undetected are now reported.
- * Type declarations for *YYSTYPE* and the `%union` declaration now allow arbitrary syntax including (even nested) generic types, and arrays. When both `%union` and *YYSTYPE* are defined *YYSTYPE* must be a simple identifier, naming the “union” type. Otherwise *YYSTYPE* may be an arbitrary type definition.
- * An extended syntax for token list declarations has been implemented. Every token has the option of declaring a “display form” that is used in all diagnostic messages. All token declaration constructs allow for optional “kind” markers.
- * Hexadecimal and unicode escapes are allowed in strings and character literals. Character literals are canonicalized before insertion in the dictionary.
- * Literal strings for filenames may use either the verbatim or normal form, and escape characters are interpreted in filenames. However, the `*.y` input file is still an 8-bit byte-file.

Changes in version 1.2.1 (August 2008)

- * The documentation has been revised.
- * *YYLTYPE* has the constraint that it must have a no-arg constructor, if it is a reference type.
- * With the */gplex* flag, the visibility of the generated *ScanBase* class is controlled by the *%visibility* marker.
- * Tests for grammars with non-terminating symbols have been implemented, with diagnostic messages for badly-formed grammars.
- * Report output is now hyperlinked *HTML* file.
- * New command-line option */diagnose* writes to the report file. New information is the shortest prefix leading to each automaton state, and shortest transition path leading to each state.

Changes in version 1.2.0 (November 2007)

- * *YYSTYPE* may be a reference type *or* a value type.
- * a number of minor bug fixes, particularly making the parsing of grammar files with errors more robust.

Changes in version 1.0.4 (August 2007)

- * New semantic markers *YYACCEPT*, *YYABORT*, *YYERROR*, *YYRECOVERING* have been added.
- * The indices of semantic value and location value references in semantic actions can now be arbitrary decimal numbers, and are bounds-checked at parser construction time.
- * The default semantic action for productions of length greater than one has been changed for greater compatibility with *Bison*.
- * The parsing code of *gppg* has been made more robust, eliminating several cases where syntactically incorrect input crashed the program.

Changes in version 1.0.3 (March 2007)

- * New command line option */babel* added.
- * Output with the */gplex* option expects to interface with version 0.5.1 or greater of *gplex*.

Changes in version 1.0.2 (February 2007)

- * Assembly attributes in *main.cs* now set the version string.
- * A bug in the string representation of productions with literal terminal symbols involving backslash-escapes has been fixed.

- * A bug in the parsing of semantic actions involving terms such “\$<kind>n” has been fixed.
- * Scanner variable *yylloc* defined in *IScanner* has become a property. This leads to better code in handwritten scanners.
- * When invoked with the */gplex* option, *gppg* emits a *ScanBase* file that defines a private backing field for the *yylloc* property, and the accessor methods. *Existing gplex scanners will require recompilation to use the new wrapper semantics.*
- * New command line option */conflicts* creates a file “*basename.conflicts*” that has complete production and item information for any conflicts in the grammar (see the section *Parser Conflict Messages* in the main documentation).

Changes in version 1.0.1 (January 2007)

- * New command line option */verbose* generates more informative warnings for shift/reduce and reduce/reduce conflicts.
- * Default location-tracking behaviour for empty productions.
- * Behavior of location-tracking when production starts with a nullable non-terminal symbol has been corrected.
- * *Static* bounds checks for semantic action indices introduced so that erroneous usages are trapped at build time.
- * Output stream was not correctly flushed when “%output=” marker used to redirect output from the specification file.
- * Unmatched left braces in semantic actions caused *gppg* to loop. This now generates a “block ended by EOF” error.

Changes in version 1.0 (October 2006)

- * Static grammar checking was added to the tool. *gppg* now checks that all non-terminals are reachable from the start symbol, and that all non-terminals possess at least one terminating production.
- * Strange case of the shift-reduce parser looping during error recovery was fixed.
- * New command line option */defines* added.
- * New command line option */gplex* customizes parser output as expected by *gplex*-generated scanners.