1 General Information: README

This is the README file for the distribution of ESS version 26.01.0

ESS is a GNU Emacs package for interactive statistical programming and data analysis. Languages supported: the S family (S, S-PLUS and R), SAS, BUGS/JAGS and Stata. ESS grew out of the desire for bug fixes and extensions to S-mode and SAS-mode as well as a consistent union of their features in one package.

Installation instructions are provided in sections for both Unix and Windows; see below.

The current development team is led by Martin Maechler since August 2004. Former project leader A.J. (Tony) Rossini (`rossini@blindglobe.net`) did the initial port to XEmacs and has been the primary coder. Martin Maechler (`maechler@stat.math.ethz.ch`) and Kurt Hornik (`Kurt.Hornik@R-project.org`) have assisted with the S family and XLispStat. Stephen Eglen (`stephen@gnu.org`) has worked mostly on R support. Richard M. Heiberger (`rmh@temple.edu`) has assisted with S/S-PLUS development for Windows. Richard and Rodney A. Sparapani (`rsparapa@mcw.edu`) have done much of the work improving SAS batch and interactive support. Rodney has also extended ESS to support BUGS/JAGS and has an interest in improving Stata support.

We are grateful to the previous developers of S-mode (Doug Bates, Ed Kademan, Frank Ritter, David M. Smith), SAS-mode (Tom Cook) and Stata-mode (Thomas Lumley).

1.1 License

The source and documentation of ESS is free software. You can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

ESS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License in the file COPYING in the same directory as this file for more details.

1.2 Installation

ESS supports GNU Emacs versions 25.1 and newer.

ESS is most likely to work with current/recent versions of the following statistical packages: R/S-PLUS, SAS, Stata, OpenBUGS and JAGS.

To build the PDF documentation, you will need a version of TeX Live or texinfo that includes texi2dvi.

There are two main methods used for installing ESS. You may install from a third-party repository or from source code. Once you install it, you must also activate or load ESS in each Emacs session, though installation from a third-party repository likely takes care of that for you. See Section 1.5 [Activating and Loading ESS], page 2, for more details.

1.3 Installing from a third-party repository

ESS is packaged by many third party repositories. Many GNU/Linux distributions package it, usually with the name "emacs-ess" or similar.

ESS is also available through Milkypostman's Emacs Lisp Package Archive (MELPA), a popular repository for Emacs packages. Instructions on how to do so are found on MELPA's website (`https://melpa.org/`). MELPA also hosts MELPA-stable with stable ESS builds. You may choose between MELPA with the latest and greatest features (and bugs) or MELPA-stable, which may lag a bit behind but should be more stable.

After installing, users should make sure ESS is activated or loaded in each Emacs session. See Section 1.5 [Activating and Loading ESS], page 2. Depending on install method, this may be taken care of automatically.

1.4 Installing from source

Stable versions of ESS are available at the ESS web page (`https://ess.r-project.org`) as a .tgz file or .zip file. ESS releases are GPG-signed, you should check the signature by downloading the accompanying `.sig` file and doing:

```
gpg --verify ess-18.10.tgz.sig
```

Alternatively, you may download the git repository. ESS is currently hosted on GitHub: `https://github.com/emacs-ess/ESS`. `git clone https://github.com/emacs-ess/ESS.git` will download it to a new directory `ESS` in the current working directory.

We will refer to the location of the ESS source files as `/path/to/ESS/` hereafter.

After installing, users should make sure they activate or load ESS in each Emacs session, see Section 1.5 [Activating and Loading ESS], page 2,

Optionally, compile Elisp files, build the documentation, and the autoloads:

```
cd /path/to/ESS/
make
```

Without this step the documentation, reference card, and autoloads will not be available. Uncompiled ESS will also run slower.

Optionally, you may make ESS available to all users of a machine by installing it site-wide. To do so, run `make install`. You might need administrative privileges:

```
make install
```

The files are installed into `/usr/share/emacs` directory. For this step to run correctly on macOS, you will need to adjust the `PREFIX` path in `Makeconf`. The necessary code and instructions are commented in that file.

1.5 Activating and Loading ESS

After installing ESS, you must activate or load it each Emacs session. ESS can be autoloaded, and if you used a third-party repository (such as your Linux distribution or MELPA) to install, you can likely skip this section and proceed directly to Section 1.6 [Check Installation], page 3,

Otherwise, you may need to add the path to ESS to `load-path` with:

```
(add-to-list 'load-path "/path/to/ESS/lisp")
```

You then need to decide whether to take advantage of deferred loading (which will result in a faster Emacs startup time) or require ESS when Emacs is loaded. To autoload ESS when needed (note that if installed from source, you must have run `make`):

```
(load "ess-autoloads")
```

To require ESS on startup, you can either put

```
(require 'ess-site)
```

or

```
(require 'ess-r-mode)
```

In your configuration file, depending on whether you want all ESS features or only R related features.

1.6 Check Installation

Restart Emacs and check that ESS was loaded from a correct location with `M-x ess-version`.

1.7 Starting an ESS process

To start an S session on Unix or on Windows when you use the Cygwin bash shell, simply type *M-x S RET*.

To start an S session on Windows when you use the MSDOS prompt shell, simply type *M-x S+6-msdos RET*.

1.8 Current Features

- Languages Supported:
  - S family (R, S, and S+ AKA S-PLUS)
  - SAS
  - BUGS/JAGS
  - Stata
  - Julia
- Editing source code (S family, SAS, BUGS/JAGS, Stata, Julia)
  - Syntactic indentation and highlighting of source code
  - Partial evaluation of code
  - Loading and error-checking of code
  - Source code revision maintenance
  - Batch execution (SAS, BUGS/JAGS)
  - Use of imenu to provide links to appropriate functions
- Interacting with the process (S family, SAS, Stata, Julia)
  - Command-line editing
  - Searchable Command history

    - Command-line completion of S family object names and file names
    - Quick access to object lists and search lists
    - Transcript recording
    - Interface to the help system
- Transcript manipulation (S family, Stata)
    - Recording and saving transcript files
    - Manipulating and editing saved transcripts
    - Re-evaluating commands from transcript files
- Interaction with Help Pages and other Documentation (R)
    - Fast Navigation
    - Sending Examples to running ESS process.
    - Fast Transfer to Further Help Pages
- Help File Editing (R)
    - Syntactic indentation and highlighting of source code.
    - Sending Examples to running ESS process.
    - Previewing

1.9 New Features

Changes and New Features in 26.01.0:

- ESS[BUGS] is still relevant due to NIMBLE keeping the language alive. However, only syntax highlighting and key-presses will be supported moving forward. For example, the `<` key now generates `<-` rather than the former `=` since equals is a valid character in NIMBLE BUGS.

Changes and New Features in 25.01.0:

- polymode: In our transition from literate libraries (such as noweb documented below with respect to 19.04), we now recommend the polymode packages as a more suitable replacement. Furthermore, we suggest the related polymodes including poly-noweb, poly-markdown and poly-R (installed in that order). The package polymode itself, as well as the polymodes packages, are all on MELPA rather than ELPA. Therefore, you need to add MELPA to the list of installation archives as follows. '`(add-to-list 'package-archives '("melpa-stable" . "https://stable.melpa.org/packages/"))`' for *M-x package-install*
- ESS[R]: The shorthand notation for lambda functions and the question mark are now fontified as keywords. Contributed by Maxime Pettinger.
- ESS[SAS]: Developed new comprehensive lists of PROCs and functions for syntax highlighting. See `etc/proc.sas` and `etc/func.sas`.

Changes and New Features in 24.01.1:

- Revert a bug introduced with the `ess-request-a-process` change

Changes and New Features in 24.01.0:

- fix docstring warnings in ess-custom

- :package-version is now set to "VERSION" in ess-custom. By make this is replaced with "24.01.0" (or similar).
- Better "collaboration" with org-mode. Now `ess-request-a-process` obeys `ess-gen-proc-buffer-name-function`, thanks to Ihor Radchenko.

Changes and New Features in 19.04 (unreleased):

- ESS[R]: When a background command is interrupted with C-g, ESS now asks the user if they want to disable background evaluations altogether. This is a resiliency measure against cases where background evals cause cascading errors or hangs.
- ESS[R]: Background commands now propagate errors to Emacs.
- ESS[R]: Background commands can now be disabled by process instad of globally. For instance when a process has failed to initialize properly, background evals are disabled for that particular process to avoid cascading errors. Other processes may still use background commands.
- ESS[R]: ESSR commands are now more robust when ESSR is not in scope. This can happen when using `browser()` in an environment that doesn't inherit from the search path.
- ESS[R]: Unexpected exits are now detected during startup. In that case an error is thrown with advice about how to recover.
- ESS[R]: `options(width = )` is now set on startup based on the width of the inferior window.
- ESS[R]: Add support for R projects and start R by default in the project folder.
- ESS[R]: Backticked symbols in the process buffer are no longer fontified as strings.
- ESS[R]: `ess-command` now runs R code in a sandboxed environment. Use `.ess.environment()` to inspect the current environment.
- ESS[R]: Added support for new syntax in R 4.0 and R 4.1. This concerns raw strings, lambda functions, and the pipe operator.
- ESS[R]: Highlight error locations in rlang style backtraces
- ESS[R]: Fixed issue that caused ESS-help to hang when usage blocks include R comments (#1025). Fix contributed by Bill Evans.
- ESS: New `ess-elisp-trace-mode` minor mode. Toggle it to start or stop tracing all `ess`-prefixed functions with `trace-function`. Tracing is useful for debugging background ESS behaviour.
- ESS[R]: `ess-get-help-aliases-list` now caches the aliases on the R side. This should speed up help lookup when the search path has changed and the aliases are read again.
- ESS: `ess-command` now uses a default timeout of 30 seconds. It should normally be avoided with long-running tasks because it causes Emacs to block while the command is running. If the timeout is reached, an error is thrown. An interrupt is also sent to the process in case of early exit.

  This is a behaviour change: you will now have to explicitly opt in blocking the whole Emacs UI for more than 30 seconds by supplying a larger timeout (use `most-positive-fixnum` for infinity).
- ESS: `ess-wait-for-process` now returns nil if a timeout is reached.

- ESS: `ess-get-words-from-vector` gains a `timeout` argument.
- ESS[R]: Fixed performance issue with argument completions. The help summary for the argument is no longer displayed in the echo area. This fixes delays and hangs (#1062).
- ESS[R]: `ess-command` is now more robust and resilient to hangs and custom prompts (#1043). It also strips continuation prompts (`+` prompts) automatically and reliably (#1116).
- ESS[R]: `ess-command` now handles sinked consoles correctly.
- ESS[R]: `ess-command` no longer changes `.Last.value`. As a result, background tasks like completions no longer affect the last value binding (#1058).
- ESS[R]: Namespaced evaluation is disable in roxygen examples (#1026). Part of this change is that namespaced evaluation has become a buffer-local rather than process-local setting (#1046). This makes it possible to disable namespaced evaluation in specific buffers or contexts.
- iESS: Inferior processes can now properly reuse frames (#987). Fixed issue that caused the current buffer to be incorrectly displayed in the new frame when `display-buffer` is set to pop up frames.
- ESS[R]: Better support for tramp. Fixed package evaluation on remote servers with Tramp (#950); process reloading (#1001); and an evaluation issue (#1024). These fixes were contributed by David Pritchard.
- ESS[R]: Automatic offsetting of R process output is now disabled by default because it produces undesirable output in some situations. To re-enable, set `inferior-ess-fix-misaligned-output` to t.
- ESS[R]: Improved `xref` lookup (`M-.`). Function locations are now always detected for package libraries listed in `ess-r-package-library-paths`.
- ESS[R]: Evaluated lines starting with the Roxygen prefix are now always stripped from the prefix, so they can be sent to the process easily. Previously, this was only the case inside the `examples` field. Since roxygen is switching to R markdown, it becomes useful to evaluate chunks of R outside examples.
- stata support is now obsolete since we were unable to elicit FSF paperwork from some of the original authors: see the lisp/obsolete sub-directory on the ESS github repo
- `ess-set-working-directory` no longer changes the active directory (as defined by the buffer-local variable `default-directory`) of the buffer where the command is called. Instead, the active directory of the inferior buffer is updated to the new working directory.
- The default of ess-eval-visibly is now `'nowait`. With this change you should no longer experience freezes while evaluating code.
- ESS[R]: There is a new menu entry for reloading the R process. It is otherwise bound to `C-c C-e C-r`. Reloading now reuses the same process name and start arguments that were used to start the process.
- iESS: Process runners now return the inferior buffer. Note that callers of inferior runners should not assume that the current buffer has been set to the inferior buffer. Instead, use `with-current-buffer` with the return value of the inferior.

- iESS[SAS]: The SAS keymap was only set in iESS buffers called '*SAS*'. This is now fixed.
- ESS[R]: Fixed longstanding indentation issues involving `::` and `:::` operators.
- Implement a more reliable check for the process busy state. Background actions such as completion and directory synchronization should not block the process and should not cause printing of the extraneous output to the interpreter.
- Activate `goto-address-mode` for url and email highlighting in inferior buffers.
- `smart-underscore` and `ess-smart-S-assign-key` have been removed. Users who liked the previous behavior (i.e. underscore inserting "<-") should bind `ess-insert-assign` to the underscore in their Emacs initialization file. For example, `(define-key ess-r-mode-map "_" #'ess-insert-assign)` and `(define-key inferior-ess-r-mode-map "_" #'ess-insert-assign)` will activate it in all ESS R buffers.
- ESS major modes are now defined using 'define-derived-mode'. This makes ESS major modes respect modern conventions such as having <language>-mode-hook and <language>-mode-map. Users are encouraged to place customizations under the appropriate mode.
- New option ess-auto-width controls setting the width option on window changes. Users can change it to 'frame, 'window, or an integer. See the documentation for details. `ess-auto-width-visible` controls visibility.
- ESS now respects `display-buffer-alist`. Users can now use `display-buffer-alist` to manage how and where windows appear. For more information and examples, see See Section "Controlling buffer display" in `ess`.
- `ess-roxy-mode` can now be enabled in non-R buffers. This is primarily intended to support roxygen documentation for cpp buffers. Preview functionality is not supported outside R buffers.
- ESS[R]: DESCRIPTION files now open in `conf-colon-mode`.
- `ess-style` now has effects when set as a file or directory local variable.
- `ess-default-style` is now obsolete, use `ess-style` instead.
- Options for 'ess-gen-proc-buffer-name-function' have been renamed. ess-gen-proc-buffer-name:projectile-or-simple was renamed to ess-gen-proc-buffer-name:project-or-simple and ess-gen-proc-buffer-name:projectile-or-directory was renamed to ess-gen-proc-buffer-name:project-or-directory. As the name suggests, these now rely on project.el (included with Emacs) rather than projectile.el, which is a third-party package.
- Eldoc fully honors `eldoc-echo-area-use-multiline-p`
- ESS[R]: `ess-r-rhub-check-package` gained new `RECOMMENDED`.
- ESS[R]: devtools commands ask about saving modified buffers before running. Users can disable the questioning with `ess-save-silently`.
- ESS[R] help pages now provide links to other help topics. This is similar with what you would see with, for example `options(help_type = ``html'')` but works with the plain-text version as well. This only works with `options(useFancyQuotes = TRUE)` (the default).
- `ess-rdired` buffers now derive from tabulated-list-mode. They should look better and be a bit faster overall. The size column now displays object sizes in bytes.

- `ess-rdired` buffers now auto-update. The frequency is governed by the new option `ess-rdired-auto-update-interval`.
- ESS[R]: `electric-layout-mode` is now supported. This automatically inserts a newline after an opening curly brace in R buffers. To enable it, customize `ess-r-mode-hook`.
- ESS[R]: imenu now supports assignment with the equals sign.
- ESS[Rd]: Rd no longer writes abbrevs to user's abbrev file.
- ESS removed support for many unused languages. This includes old versions of S+, ARC, OMG, VST, and XLS.
- ess-r-runner-prefixes was modified to find R-4 and later.
- ESS no longer activates eldoc if the user has disabled global-eldoc-mode.

The following have been made obsolete or removed, see their documentation for more detail:

- Libraries for literate data analysis are obsolete and not loaded by default. This includes `ess-noweb`, `ess-swv`, and related functionality like `Rnw-mode`. Users are encouraged to switch to one of several other packages that deal with these modes. For example, polymode `https://github.com/polymode/poly-R/`, `https://polymode.github.io/`, or markdown-mode with edit-indirect `https://jblevins.org/projects/markdown-mode`.
- Support for `auto-complete` is obsolete. The `auto-complete` package is unmaintained and so ESS support is now obsolete. Users are encouraged to switch to `company-mode` instead.
- User options for controlling display of buffers. This includes `ess-show-buffer-action`, `inferior-ess-same-window`, `inferior-ess-own-frame`, and `inferior-ess-frame-alist`. See above about ESS respecting `display-buffer-alist`.
- Variables `ess-tab-always-indent` and `ess-tab-complete-in-script`. Use the Emacs-wide setting of `tab-always-indent` instead.
- `inferior-ess-*-start-file` variables. All modes except Stata did not respect customization of this variable. In order to load a file on startup, you should put a function on `ess-*-post-run-hook`.

Bug Fixes in 18.10.3:

- More `Makefile` fixes, notably installing `*.el`s.

Bug Fixes in 18.10.2:

- ESS[R] Fix namespace evaluation in non-installed packages. Evaluation is directed into GlobalEnv as originally intended.
- `Makefile` fixes, notably for `make install` and including full docs in the tarballs.

Bug Fixes in 18.10.1:

- New functions `ess-eval-line-visibly-and-step` (*C-c C-n* and `ess-eval-region-or-line-visibly-and-step` (*C-RET*) which behave as the old versions of `ess-eval-line-and-step` and `ess-eval-region-or-line-and-step`.

Changes and New Features in 18.10:

- This is the last release to support Emacs older than 25.1. Going forward, only GNU Emacs 25.1 and newer will be supported. Soon after this release, support for older Emacs versions will be dropped from the git master branch. Note that MELPA uses the git master branch to produce ESS snapshots, so if you are using Emacs < 25.1 from MELPA and are unable to upgrade, you should switch to MELPA-stable.
- ESS now displays the language dialect in the mode-line. So, for example, R buffers will now show ESS[R] rather than ESS[S].
- The ESS manual has been updated and revised.
- The ESS initialization process has been further streamlined. If you update the autoloads (which installation from `package-install` does), you should not need to `(require 'ess-site)` at all, as autoloads should automatically load ESS when it is needed (e.g. the first time an R buffer is opened). In order to defer loading your ESS config, you may want to do something like `(with-require-after-load "ess" <ess-config-here>)` in your Emacs init file. Users of the popular `use-package` Emacs package can now do `(use-package ess :defer t)` to take advantage of this behavior. For more information on this feature, see See Section "Activating and Loading ESS" in `ess`.
- ESS now respects Emacs conventions for keybindings. This means that The `C-c [letter]` bindings have been removed. This affects `C-c h`, which was bound to `ess-eval-line-and-step-invisibly` in `sas-mode-local-map`; `C-c f`, which was bound to `ess-insert-function-outline` in `ess-add-MM-keys`; and `C-c h`, which was bound to `ess-handy-commands` in `Rd-mode-map`, `ess-noweb-minor-mode-map`, and `ess-help-mode-map`
- Functions `ess-eval-line-and-step` and `ess-eval-region-or-line-and-step` now behave consistently with other evaluation function inside a package.
- ESS[R]: `ess-r-package-use-dir` now works with any mode. This sets the working directory to the root of the current package including for example C or C++ files within `/src`).
- ESS[R]: Long + + prompts in the inferior no longer offset output.
- ESS[R]: New option `strip` for `inferior-ess-replace-long+`. This strips the entire + + sequence.
- ESS modes now inherit from `prog-mode`. In the next release, ESS modes will use `define-derived-mode` so that each mode will have (for example) its own hooks and keymaps.
- ESS[R]: Supports flymake in R buffers for Emacs 26 and newer. Users need to install the `lintr` package to use it. Customizable options include `ess-use-flymake`, `ess-r-flymake-linters`, and `ess-r-flymake-lintr-cache`.
- ESS[R]: Gained support for xref in Emacs 25+ See Section "Xref" in *The Gnu Emacs Reference Manual*.
- ESS[R]: The startup screen is cleaner. It also displays the startup directory with an explicit `setwd()`.
- ESS[R]: Changing the working directory is now always reflected in the process buffer.
- ESS[R]: `Makevars` files open with `makefile-mode`.

- New variable `ess-write-to-dribble`. This allows users to disable the dribble (`*ESS*`) buffer if they wish.
- All of the `*-program-name` variables have been renamed to `*-program`. Users who previously customized e.g. `inferior-ess-R-program-name` will need to update their customization to `inferior-ess-R-program`. These variables are treated as risky variables.
- `ess-smart-S-assign` was renamed to `ess-insert-assign`. It provides similar functionality but for any keybinding, not just '_'. For instance if you bind it to ';', repeated invocations cycle through between assignment and inserting ';'.
- `C-c C-=` is now bound to `ess-cycle-assign` by default. See the documentation for details. New user customization option `ess-assign-list` controls which assignment operators are cycled.
- ESS[R] In remote sessions, the ESSR package is now fetched from GitHub.
- Commands that send the region to the inferior process now deal with rectangular regions. See the documentation of `ess-eval-region` for details. This only works on Emacs 25.1 and newer.
- ESS[R]: Improvements to interacting with iESS in non-R files. Interaction with inferior process in non-R files within packages (for instance C or C++ files) has been improved. This is a work in progress.
- ESS[R]: Changing the working directory is now always reflected in the process buffer.
- ESS[JAGS]: *.jog and *.jmd files no longer automatically open in JAGS mode.

Many improvements to fontification:

- Improved customization for faces. ESS now provides custom faces for (nearly) all faces used and places face customization options into their own group. Users can customize these options using *`M-x customize-group RET ess-faces`*.
- Many new keywords were added to `ess-R-keywords` and `ess-R-modifiers`. See the documentation for details.
- ESS[R]: `in` is now only fontified when inside a `for` construct. This avoids spurious fontification, especially in the output buffer where 'in' is a common English word.
- ESS: Font-lock keywords are now generated lazily.That means you can now add or remove keywords from variables like `ess-R-keywords` in your Emacs configuration file after loading ESS (i.e. in the `:config` section for `use-package` users).
- ESS[R]: Fontification of roxygen `@param` keywords now supports comma-separated parameters.
- ESS[R]: Certain keywords are only fontified if followed by a parenthesis. Function-like keywords such as `if ()` or `stop()` are no longer fontified as keyword if not followed by an opening parenthesis. The same holds for search path modifiers like `library()` or `require()`.
- ESS[R]: Fixed fontification toggling. Especially certain syntactic elements such as `%op%` operators and backquoted function definitions.
- ESS[R]: `ess-font-lock-toggle-keyword` can be called interactively. This command asks with completion for a font-lock group to toggle. This functionality is equivalent to the font-lock menu.

Notable bug fixes:

- `prettify-symbols-mode` no longer breaks indentation. This is accomplished by having the pretty symbols occupy the same number of characters as their non-pretty cousins. You may customize the new variable `ess-r-prettify-symbols` to control this behavior.
- ESS: Inferior process buffers are now always displayed on startup. Additionally, they don't hang Emacs on failures.

Obsolete libraries, functions, and variables:

- The `ess-r-args.el` library has been obsoleted and will be removed in the next release. Use `eldoc-mode` instead, which is on by default.
- Functions and options dealing with the smart assign key are obsolete. The following functions have been made obsolete and will be removed in the next release of ESS: `ess-smart-S-assign`, `ess-toggle-S-assign`, `ess-toggle-S-assign-key`, `ess-disable-smart-S-assign`.

  The variable `ess-smart-S-assign-key` is now deprecated and will be removed in the next release. If you would like to continue using '_' for inserting assign in future releases, please bind `ess-insert-assign` in `ess-mode-map` the normal way.
- ESS[S]: Variable `ess-s-versions-list` is obsolete and ignored. Use `ess-s-versions` instead. You may pass arguments by starting the inferior process with the universal argument.

Changes and New Features in 17.11:

- The ESS initialization process has been streamlined. You can now load the R and Stata modes independently from the rest of ESS. Just put `(require 'ess-r-mode)` or `(require 'ess-stata-mode)` in your init file. This is for experienced Emacs users as this requires setting up autoloads for `.R` files manually. We will keep maintaining `ess-site` for easy loading of all ESS features.
- Reloading and quitting the process is now more robust. If no process is attached, ESS now switches automatically to one (prompting you for selection if there are several running). Reloading and quitting will now work during a debug session or when R is prompting for input (for instance after a crash). Finally, the window configuration is saved and restored after reloading to prevent the buffer of the new process from capturing the cursor.
- ESS[R]: New command `ess-r-package-use-dir`. It sets the working directory of the current process to the current package directory.
- ESS[R] Lookup for references in inferior buffers has been improved. New variable `ess-r-package-source-roots` contains package sub-directories which are searched recursively during the file lookup point. Directories in `ess-tracebug-search-path` are now also searched recursively.
- ESS[R] Namespaced evaluation is now automatically enabled only in the `R/` directory. This way ESS will not attempt to update function definitions from a package if you are working from e.g. a test file.

Changes and New Features in 16.10:

- ESS[R]: Syntax highlighting is now more consistent. Backquoted names are not fontified as strings (since they really are identifiers). Furthermore they are now correctly recognized when they are function definitions or function calls.
- ESS[R]: Backquoted names and `%op%` operators are recognized as sexp. This is useful for code navigation, e.g. with *C-M-f* and *C-M-b*.
- ESS[R]: Integration of outline mode with roxygen examples fields. You can use outline mode's code folding commands to fold the examples field. This is especially nice to use with well documented packages with long examples set. Set `ess-roxy-fold-examples` to non-nil to automatically fold the examples field when you open a buffer.
- ESS[R]: New experimental feature: syntax highlighting in roxygen examples fields. This is turned off by default. Set `ess-roxy-fontify-examples` to non-nil to try it out.
- ESS[R]: New package development command `ess-r-devtools-ask` bound to *C-c C-w C-a*. It asks with completion for any devtools command that takes `pkg` as argument.
- ESS[R]: New command *C-c C-e C-r* to reload the inferior process. Currently only implemented for R. The R method runs `inferior-ess-r-reload-hook` on reloading.
- ESS[R]: `ess-r-package-mode` is now activated in non-file buffers as well.

Bug fixes in 16.10:

- ESS[R]: Fix broken (un)flagging for debugging inside packages
- ESS[R]: Fixes (and improvements) in Package development
- ESS[R]: Completion no longer produces `...=` inside `list( )`.
- ESS[R]: Better debugging and tracing in packages.
- ESS[R]: Better detection of symbols at point.
- ESS[R]: No more spurious warnings on deletion of temporary files.
- ESS[julia]: help and completion work (better)
- ESS[julia]: available via `ess-remote`

Changes and New Features in 16.04:

- ESS[R]: `developer` functionality has been refactored. The new user interface consists of a single command `ess-r-set-evaluation-env` bound by default to *C-c C-t C-s*. Once an evaluation environment has been set with, all subsequent ESS evaluation will source the code into that environment. By default, for file within R packages the evaluation environment is set to the package environment. Set `ess-r-package-auto-set-evaluation-env` to `nil` to disable this.
- ESS[R]: New `ess-r-package-mode` This development mode provides features to make package development easier. Currently, most of the commands are based on the `devtools` packages and are accessible with *C-c C-w* prefix. See the documentation of `ess-r-package-mode` function for all available commands. With *C-u* prefix each command asks for extra arguments to the underlying devtools function. This mode is automatically enabled in all files within R packages and is indicated with `[pkg:NAME]` in the mode-line.
- ESS[R]: Help lookup has been improved. It is now possible to get help for namespaced objects such as pkg::foobar. Furthermore, ESS recognizes more reliably when you change `options('html_type')`.

- ESS[R]: New specialized breakpoints for debugging magrittr pipes
- ESS: ESS now implements a simple message passing interface to communicate between ESS and inferior process.

Bug fixes in 16.04:

- ESS[R]: Roxygen blocks with backtics are now correctly filled
- ESS[R]: Don't skip breakpoints in magrittr's `debug_pipe`
- ESS[R]: Error highlighting now understands 'testthat' type errors
- ESS[Julia]: Added getwd and setwd generic commands

1.10 Reporting Bugs

Please post bug reports, suggestions etc. on our github issue tracker (`https://github.com/emacs-ess/ESS/issues`); if not possible, e-mail them to `ESS-help@r-project.org`.

The easiest way to set this up, is within Emacs by typing

M-x ess-submit-bug-report

This also gives the maintainers valuable information about your installation which may help us to identify or even fix the bug.

If Emacs reports an error, backtraces can help us debug the problem. Type "M-x set-variable RET debug-on-error RET t RET". Then run the command that causes the error and you should see a *Backtrace* buffer containing debug information; send us that buffer.

Note that comments, suggestions, words of praise and large cash donations are also more than welcome.

1.11 Mailing Lists

There is a mailing list for discussions and announcements relating to ESS. Join the list by sending an e-mail with "subscribe ess-help" (or "help") in the body to `ess-help-request@r-project.org`; contributions to the list may be mailed to `ess-help@r-project.org`. Rest assured, this is a fairly low-volume mailing list.

The purposes of the mailing list include

- helping users of ESS to get along with it.
- discussing aspects of using ESS.
- suggestions for improvements.
- announcements of new releases of ESS.
- posting small patches to ESS.

1.12 Authors

- A.J. Rossini (`mailto:blindglobe@gmail.com`)
- Richard M. Heiberger (`mailto:rmh@temple.edu`)
- Kurt Hornik (`mailto:Kurt.Hornik@R-project.org`)
- Martin Maechler (`mailto:maechler@stat.math.ethz.ch`)
- Rodney A. Sparapani (`mailto:rsparapa@mcw.edu`)

- Stephen Eglen (mailto:stephen@gnu.org)
- Sebastian P. Luque (mailto:spluque@gmail.com)
- Henning Redestig (mailto:henning.red@googlemail.com)
- Vitalie Spinu (mailto:spinuvit@gmail.com)
- Lionel Henry (mailto:lionel.hry@gmail.com)
- J. Alexander Branham (mailto:alex.branham@gmail.com)