

1 General Information: README

This is the README file for the distribution of ESS version 12.04-2

ESS is a GNU Emacs and XEmacs mode for interactive statistical programming and data analysis. Languages supported: the S family (S, S-PLUS and R), SAS, BUGS/JAGS, Stata and XLispStat. 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 Kademian, 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 Stability

All recent released versions are meant to be release-quality versions. While some new features are being introduced, we are cleaning up and improving the interface. We know that there are many remaining opportunities for documentation improvements, but all contributors are volunteers and time is precious. Patches or suggested fixes with bug reports are much appreciated!

1.3 Requirements

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

ESS supports current, and recent, stable versions of GNU Emacs (specifically, the 22.x and 23.x series; alpha/beta/pre-release versions are NOT SUPPORTED). XEmacs currently does not allow all ESS features. Basic ESS functionality however is supported for the time being (XEmacs stable 21.4.14 or higher).

To build the PDF documentation, you will need a version of TeX Live or texinfo that includes texi2dvi (BEWARE: recent TeX Live, and some texinfo RPMs, do NOT include texi2dvi).

1.4 Getting the Latest Version

The latest released version of ESS is always available on the web at: [ESS web page](#) or [StatLib](#)

1.4.1 ESS subversion repository

The latest development version of ESS is available via <https://svn.R-project.org/ESS/>, the ESS Subversion repository. If you have a Subversion client (see <http://subversion.tigris.org/>), you can download the sources using:

```
% svn checkout https://svn.r-project.org/ESS/trunk path
```

which will put the ESS files into directory *path*. Later, within that directory, ‘svn update’ will bring that directory up to date. Windows-based tools such as TortoiseSVN are also available for downloading the files. Alternatively, you can browse the sources with a web browser at: [ESS SVN site](#). However, please use a subversion client instead to minimize the load when retrieving.

If you remove other versions of ESS from your emacs load-path, you can then use the development version by adding the following to .emacs:

```
(load "/path/to/ess-svn/lisp/ess-site.el")
```

Note that https is required, and that the SSL certificate for the Subversion server of the R project is

```
Certificate information:
- Hostname: svn.r-project.org
- Valid: from Jul 16 08:10:01 2004 GMT until Jul 14 08:10:01 2014 GMT
- Issuer: Department of Mathematics, ETH Zurich, Zurich, Switzerland, CH
- Fingerprint: c9:5d:eb:f9:f2:56:d1:04:ba:44:61:f8:64:6b:d9:33:3f:93:6e:ad
```

(currently, there is no “trusted certificate”). You can accept this certificate permanently and will not be asked about it anymore.

1.4.2 Git for development

For development and experimentation on new features, there is now a GitHub branch for ESS, available at <https://github.com/emacs-ess/ESS>.

1.5 Installation

1.6 Unix installation

For a **Unix or Unix-like installation**, please follow the next steps. Retrieve the latest tgz file (‘*ess-VERSION.tgz*’) from [ESS downloads area](#).

GNU Emacs Simple Instructions: for recent versions of Emacs (22.x or higher) and ESS, the installation process is simple.

1. Extract all the files from ‘*ess-VERSION.tgz*’ into the directory ‘*PREFIX/site-lisp*’ where ‘*PREFIX*’ is appropriate for GNU Emacs on your system; ‘*PREFIX*’ will most likely

be either `‘/usr/share/emacs’` or `‘/usr/local/share/emacs’` (on Mac OS X, `‘PREFIX’` will most likely be something like `‘/Applications/Emacs.app/Contents/Resources’`):

```
GNU tar % gtar zxf ess-VERSION.tgz -C PREFIX/site-lisp
Unix tar % gunzip < ess-VERSION.tgz | tar xf - -C PREFIX/site-lisp
```

2. Then, add the line

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

to `‘~/.emacs’` and restart Emacs.

3. If you see a buffer named `‘*ESS*’`, then the simple instructions were most likely successful. If not, then read further.
4. It could be that you have an older version of Emacs, some other problem with your installation or you are not a sysadmin. Whatever the case, you may need to edit `‘ess-site.el’` manually. If that is the case, then you should create a directory just for ESS like `‘~/ess’` or `‘/usr/local/ess’` and unpack ESS there. That way, your changes to `‘ess-site.el’` will not be lost if you update Emacs later.
5. Replace the line above with

```
(load "~/ess/ess-VERSION/lisp/ess-site")
```

in `‘~/.emacs’` and restart Emacs.

6. If you see a buffer named `‘*ESS*’`, then the manual instructions were most likely successful. If not, then send a query to ess-help@r-project.org explicitly describing your problem and your environment including operating system, Emacs version, ESS version, etc.

XEmacs Simple Instructions: for recent versions of XEmacs (21.x or higher) and ESS, the installation process is simple.

1. Extract all the files from `‘ess-VERSION.tgz’` when you are in the current working directory of `‘/usr/local/LOCATION/xemacs/site-packages’` which exists for packages like ESS where `‘LOCATION’` is `‘lib’` for legacy installations and `‘share’` now (and for Mac OS X create a link to it from the directory `‘/Applications/XEmacs.app/Contents/Resources/site-lisp’`):

```
%prompt gtar zxf ess-VERSION.tgz          # for GNU tar
%prompt gunzip < ess-VERSION.tgz | tar xf - # for Unix tar
```

2. Then, add the line

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

to `‘~/.xemacs/init.el’` and restart XEmacs.

3. If you see a buffer named `‘*ESS*’`, then the simple instructions were most likely successful. If not, then read further.
4. It could be that you have an older version of XEmacs, some other problem with your installation or you are not a sysadmin. Whatever the case, you may need to edit `‘ess-site.el’` manually. If that is the case, then you should create a directory just for ESS like `‘~/ess’` or `‘/usr/local/ess’` and unpack ESS there. That way, your changes to `‘ess-site.el’` will not be lost if you update XEmacs later.
5. Replace the line above with

```
(load "~/ess/ess-VERSION/lisp/ess-site")
```

in `‘~/.xemacs/init.el’` and restart XEmacs.

6. If you see a buffer named `*ESS*`, then the manual instructions were most likely successful. If not, then send a query to ess-help@r-project.org explicitly describing your problem and your environment including operating system, XEmacs version, ESS version, etc.

1. **(OPTIONAL) COMPILING E-LISP:**

Edit the default locations of `LISPDIR`, `INFODIR` and `ETCDIR` in Section 1 of `'Makeconf'` (if you are using XEmacs, then edit the XEmacs subsection in Section 1).

You can compile those files by:

```
make all
```

When that completes successfully, install the compiled files:

```
make install
```

1.7 Microsoft Windows installation

For **Microsoft Windows installation**, please follow the next steps. Retrieve the latest zip file (`'ess-VERSION.zip'`) from [ESS downloads area](#).

GNU Emacs Simple Instructions: for recent versions of Emacs (22.x or higher) and ESS, the installation process is simple.

1. Extract all the files from `'ess-VERSION.zip'` (by double clicking on it and selecting "Extract all files" which launches the Folders Extraction Wizard) into an `'ESS'` sub-directory of the `'site-lisp'` directory that exists for packages like ESS. If GNU Emacs was installed in the default location, then this directory can be found somewhere like `'C:\Program Files\GNU Emacs\emacs-22.x\site-lisp'`

2. Add the line

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

to `'~/.emacs'` and restart Emacs.

3. If you see a buffer named `*ESS*`, then the simple instructions were most likely successful. If not, then read further.
4. It could be you have an older version of Emacs or some other problem with your installation. Either way, you may need to edit `'C:\ess\ess-VERSION\lisp\ess-site.el'` manually. If that is the case, then you should create a directory just for ESS like `'C:\ess'` and unpack ESS there. That way, your changes to `'C:\ess\ess-VERSION\lisp\ess-site.el'` will not be lost if you update Emacs later.
5. Replace the line above with

```
(load "C:/ess/ess-VERSION/lisp/ess-site")
```

in `'~/.emacs'` and restart Emacs.

6. If you see a buffer named `*ESS*`, then the manual instructions were most likely successful. If not, then send a query to ess-help@r-project.org explicitly describing your problem and your environment including operating system, Emacs version, ESS version, etc.

XEmacs Simple Instructions: for recent versions of XEmacs (21.x or higher), the installation process is much simpler. Hopefully, these simple instructions will work for you. If not, then more detailed, manual instructions follow.

1. Extract all the files from 'ess-VERSION.zip' (by double clicking on it and selecting "Extract all files" which launches the Folders Extraction Wizard) into the 'site-packages' directory that exists for packages like ESS. If XEmacs was installed in the default location, then this directory can be found at 'C:\Program Files\XEmacs\site-packages'.
2. XEmacs requires the HOME environment variable to be defined. You can create it by visiting the following dialog: My Computer->Control Panel->System->Advanced->Environment Variables In the User variables window, press New. And create a variable named HOME with a value something like (you must use forward slashes / rather than backslashes \) c:/Documents and Settings/%USERNAME%/Application Data. Then press OK for that window and press OK for the main window. *If you also have GNU Emacs installed, GNU Emacs will recognize HOME and expand ~ accordingly.*
3. Now launch XEmacs and do a C-x C-f followed by a ~. From the Subdir menu, select Create Directory, and enter .xemacs
4. Add the line

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

to '~/.xemacs/init.el' and restart XEmacs.
5. If you see a buffer named '*ESS*', then the simple instructions were most likely successful. If not, then read further.
6. It could be you have an older version of XEmacs or some other problem with your installation. Either way, you may need to edit 'C:\ess\ess-VERSION\lisp\ess-site.el' manually. If that is the case, then you should create a directory just for ESS like 'C:\ess' and unpack ESS there. That way, your changes to 'C:\ess\ess-VERSION\lisp\ess-site.el' will not be lost if you update XEmacs later.
7. Replace the line above with

```
(load "C:/ess/ess-VERSION/lisp/ess-site")
```

in '~/.xemacs/init.el' and restart XEmacs.
8. If you see a buffer named '*ESS*', then the manual instructions were most likely successful. If not, then send a query to ess-help@r-project.org explicitly describing your problem and your environment including operating system, XEmacs version, ESS version, etc.

Now, you should be ready to use ESS. For example, to edit statistical programs, load the files with the requisite extensions (".sas" for SAS, ".S" or "s" or "q" or "Q" for S-PLUS, ".r" or ".R" for R, and ".lsp" for XLispStat). One further step is needed if you wish to run statistical processes, see below.

To run statistical processes under ESS, Windows users will need to make sure that the directories for the software they will be using is in the PATH environment variable.

On Windows NT/2000/XP, add the directories to the PATH using the My Computer->Control Panel->System->Advanced->Environment Variables menu. Note that the directory containing the program is added to the PATH, not the program itself. One such line is needed for each software program. Be sure to use the abbreviation `progra~1` and not the long version with embedded blanks as this may cause problems. Also, make sure to use backslashes \ since Windows requires them.

An alternative, for R users, is that rather than adjusting the PATH variable, you can add the following to your emacs initialization file (and restart emacs):

```
(setq inferior-R-program-name "c:/progra~1/R/R-2.2.1/bin/Rterm.exe")
```

This assumes that you have installed R-2.2.1 in the default location. Change the path otherwise to point to other locations.

Windows users who place S-PLUS anywhere other than the default location will also need to add the following three lines (properly adjusted for their location):

```
(setq-default inferior-S+6-program-name  
  "c:/progra~1/Insightful/SPLUS70/cmd/Splus")  
(setq-default inferior-Sqpe+6-SHOME-name  
  "c:/progra~1/Insightful/SPLUS70")  
(setq-default inferior-Sqpe+6-program-name  
  "c:/progra~1/Insightful/SPLUS70/cmd/Sqpe.exe")
```

The above example uses the default location of S-PLUS in `c:\progra~1\Insightful`. Please note that ESS considers S-PLUS 6, 7, and 8 to be variants of S+6.

These users may also need to modify the emacs variable `ess-SHOME-versions` to match their installation in order to get the full set of S-PLUS versions on their machine into the ESS menu.

To start the S-PLUS [678].x GUI from ESS under emacs:

1. If you use Cygwin bash as your primary shell, then

```
M-x S  
(or M-x S+6).
```

2. If you use the MSDOS prompt window as your primary shell, then

```
M-x S+6-msdos
```

You will then be asked for a pathname ("S starting data directory?"), from which to start the process. The prompt will propose your current directory as the default. ESS will start the S-PLUS GUI. There will be slight delay during which emacs is temporarily frozen. ESS will arrange for communication with the S-PLUS GUI using the DDE protocol. Send lines or regions from the emacs buffer containing your S program (for example, `'myfile.s'`) to the S-PLUS Commands Window with the `C-c C-n` or `C-c C-r` keys. (If you are still using S-PLUS 4.x or 2000,\ then use `M-x S+4` or `M-x S+4-msdos`.)

To start an S-PLUS [678].x session inside an emacs buffer—and without the S-PLUS GUI:

```
M-x Sqpe  
(or M-x Sqpe+6).
```

This works with both the bash and msdos shells. You will then be asked for a pathname ("S starting data directory?"), from which to start the process. The prompt will propose your current directory as the default. You get Unix-like behavior, in particular the entire transcript is available for emacs-style search commands. Send lines or regions from the emacs buffer containing your S program (for example, `'myfile.s'`) to the `*S+6*` buffer with the `C-c C-n` or `C-c C-r` keys. Interactive graphics are available with Sqpe by using the java library supplied with S-PLUS 6.1 and newer releases. Enter the commands:

```
library(winjava)
java.graph()
```

Graphs can be saved from the `java.graph` device in several formats, but not PostScript. If you need a PostScript file you will need to open a separate `postscript` device. (If you are still using S-PLUS 4.x or 2000, then use `M-x Sqpe+4.`)

To connect to an already running S-PLUS GUI (started, for example, from the S-PLUS icon):

```
M-x S+6-existing
```

or

```
M-x S+6-msdos-existing
```

You will then be asked for a pathname ("S starting data directory?"), from which to start the process. The prompt will propose your current directory as the default. ESS will arrange for communication with the already running S-PLUS GUI using the DDE protocol. Send lines or regions from the emacs buffer containing your S program (for example, `'myfile.s'`) to the S-PLUS Commands Window with the `C-c C-n` or `C-c C-r` keys. (If you are still using S-PLUS 4.x or 2000, then use `M-x S+4-existing` or `M-x S+4-msdos-existing.`)

If you wish to run R, you can start it with:

```
M-x R
```

XLispStat can not currently be run with

```
M-x XLS
```

Hopefully, this will change. However, you can still edit with emacs, and cut and paste the results into the XLispStat `*Listener*` Window under Microsoft Windows.

1.8 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.`

1.9 Current Features

- Languages Supported:
 - S family (S 3/4, S-PLUS and R)
 - SAS
 - BUGS/JAGS
 - Stata
 - XLispStat including Arc and ViSta
- Editing source code (S family, SAS, BUGS/JAGS, Stata)
 - 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)
 - 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
- Help File Editing (R)
 - Syntactic indentation and highlighting of source code.
 - Sending Examples to running ESS process.
 - Previewing

1.10 New Features

Changes/New Features in 12.04-1:

- `iESS[Stata]`: New interactive help invocation.
- `iESS[Stata]`: New custom variable `inferior-STA-start-file`.
- `iESS[Stata]`: `inferior-STA-program-name` is now “stata” and can be customized.
- `ESS[Stata]` New sections in stata help files `Syntax(s-S)`, `Remarks(r)`, `Title(t)`.

Bug Fixes in 12.04-1:

- `ESS[R]`: Better `ess-tracebug` error handling.
- `ESS[R]`: Corrected `ess-eldoc` help string filtering and improved argument caching.
- `ESS[R]`: Indentation of non-block `if/else/for/while` lines fixed.
- `M-x ess-version` should work better.
- `ESS`: Filename completion now again works inside strings.
- `iESS[Stata]`: Fixed prompt detection issue.
- `ESS[Rd]`: R is autostarted also from here, when needed.

Changes/New Features in 12.04:

- `ESS`: Reverting new behavior of 12.03, `TAB` in `ess-mode` no longer completes by default. If you want smart `TAB` completion in R and S scripts, similarly to `iESS` behavior, set the variable `ess-tab-complete-in-script` to `t`. Also see `ess-first-tab-never-complete` for how to customize where first `TAB` is allowed to complete.

- ESS: completion is consistently bound to *M-TAB* (aka *M-C-i*) in both Emacs23 and Emacs24.
- ESS: The variable `ess-arg-function-offset-new-line` introduced in ESS(12.03) now accepts a list with the first element a number to indicate that the offset should be computed from the indent of the previous line. For example setting it to '(2) results in:

```
a <- some.function(
  arg1,
  arg2)
```

Changes/New Features in 12.03:

- ESS indentation: new offset variable `ess-arg-function-offset-new-line` controlling for the indentation of lines immediately following open '('. This is useful to shift backwards function arguments after a long function call expression:

```
a <- some.function(
  arg1,
  arg2)
```

instead of the old

```
a <- some.function(
  arg1,
  arg2)
```

If '(' is not followed by new line the behavior is unchanged:

```
a <- some.function(arg1,
  arg2)
```

This variable should be set as part of indentation style lists, or in `ess-mode` hook.

- ESS[R]: `C-c .` sets (indentation) style.
- ESS: In ESS buffers `yank(C-y)` command accepts double argument `C-u C-u` to paste commands only. It deletes any lines not beginning with a prompt, and then removes the prompt from those lines that remain. Useful to paste code from emails, documentation, inferior ESS buffers or transcript files.
- Documentation: ESS user manual has been rearranged and completed with several new chapters and sections to reflect newly added features ("Completion", "Developing with ESS", "ESS tracebug", "ESS developer", "ESS Eldoc", "IDO Completion" and "Evaluating Code")
- RefCard: Reference card was updated to include new features.
- Eldoc: Eldoc was rewritten and is activated by default. See `ess-use-eldoc`, `ess-eldoc-show-on-symbol`, `ess-eldoc-abbreviation-style` variables for how to change the default behavior. *Note:* `skeleton-pair-insert-maybe` prohibits eldoc display, on (insertion.
- ESS[R]: Eldoc shows arguments of a generic function whenever found.
- ESS: *TAB* in `ess-mode` now indents and completes, if there is nothing to indent. Set `ess-first-tab-never-completes-p` to `t` to make *TAB* never complete on first invocation. Completion mechanism is similar to the completion in the `inferior-ess-mode` – a filename expansion is tried, if not found ESS completes the symbol by querying the process.

- ESS for emacs version 24 or higher: ESS is fully compatible with the emacs 24 completion scheme, i.e. all the completion is done by `completion-at-point`. Also in accordance with emacs conventions, ESS doesn't bind *M-TAB* for emacs 24 or higher. *M-TAB* calls the default `complete-symbol`.
- ESS[R]: Out of the box integration with Auto Completion mode (<http://cx4a.org/software/auto-complete/>). Three AC sources `ac-source-R-args`, `ac-source-R-objects` and `ac-source-R` are provided. The last one combines the previous two and makes them play nicely together. Set `ess-use-auto-complete` to `t` to start using it. Refer to documentation string of `ac-use-auto-complete` for further information.
- ESS[R]: New unified and fast argument completion system, comprised of `ess-funname.start`, `ess-function-arguments`, `ess-get-object-at-point`. Eldoc and auto-completion integration are using this system.
- ESS: `ess-switch-to-end-of-ESS(C-c C-z)`, and `ess-switch-to-ESS(C-c C-y)`: Automatically start the process whenever needed.
- ESS[R]: `roxy` knows about previewing text version of the documentation. Bound to `C-c C-e t`.
- ESS[R]: Solved the “nil filename” bug in roxygen support.
- ESS[R]: `ess-tracebug` is now part of ESS:

New Features:

- Source injection: Tracebug now can inject source references on the fly during code evaluation, i.e. you don't have to source your file, but just evaluate your code in normal fashion. Variable `ess-tracebug-inject-source-p` controls this behavior - if `t`, always inject source reference, if `'function`, inject only for functions (this is the default), if `nil`, never inject.
During the source injection the value of `'ess-eval-visibly-p` is ignored.
- Org-mode support: Visual debugger is now aware of the temporary org source editing buffer (`C-c '`) and jumps through this buffers if still alive, or in original org buffer otherwise.
- New keys in watch mode: `?` and `d`
- Two new hooks: `ess-tracebug-enter-hook` and `ess-tracebug-exit-hook`
- ESS[R]: New package `ess-developer` to evaluate R code directly in the package environment and namespace. It can be toggled on and off with `C-c d t`. When `ess-developer` is on all ESS evaluation commands are redefined to evaluate code in appropriate environments. Add package names to the list of your development packages with `C-d a`, and remove with `C-d r`. Source the current file with `C-d s`. Evaluation function which depend on `'ess-eval-region` ask for the package to source the code into, `ess-eval-function` and alternatives search for the function name in the development packages' environment and namespace and insert the definition accordingly. See the documentation section “Developing with ESS/ESS developer” for more details.

- ESS[R] help system:

New Features:

- `q` quits window instead of calling `ess-switch-to-end-of-ESS`. This is consistent with emacs behavior help and other special buffers (*breaking change*).
- `k` kills window without asking for the name (pointed by Sam Steingold)
- Help map inherits from `special-mode-map` as suggested by Sam Steingold.
- Package index: new function `ess-display-index` bound to `i` in help mode map.
- Package vignettes: new function `ess-display-vignettes` bound to `v` in help mode map.
- Display help in HTML browser: new function `ess-display-help-in-browser` bound to `w` in help mode map. It depends on R's `browser` option.
- New custom variable `ess-help-pop-to-buffer`: if non-nil ESS help buffers are given focus on display. The default is `t` (*breaking change*).
- New menu entries for the above functions.
- Bogus help buffers are no longer generated by default, i.e. buffers of the form “No documentation for ‘foo’ in specified packages and libraries: you could try ‘??foo’”. `ess-help-kill-bogus-buffers` now defaults to `t`. Beware, there may be instances where the default is unsatisfactory such as debugging and/or during R development. Thanks to Ross Boylan for making the suggestion, Sam Steingold for reminding us of this variable and Martin Maechler for the warning.
- ESS now uses IDO completing read functionality for all the interactive requests. It uses ido completion mechanism whenever available, and falls back on classical completing-read otherwise. You can set `ess-use-ido` to nil if you don't want the IDO completion. See the documentation string of `ess-use-ido` for more information about IDO and ESS configuration.
- ESS[S]: “,” is bound to `ess-smart-comma`: If comma is invoked at the process marker of an ESS inferior buffer, request and execute a command from ‘`ess-handy-commands`’ list. If `ess-R-smart-operators` is `t` ‘`ess-smart-comma`’ also inserts “ ” after comma.
- ESS[S], notably R: Variable ‘`ess-handy-commands`’ stores an alist of useful commands which are called by `ess-smart-comma` in the inferior buffer.

Currently containing:

```
change-directory      ess-change-directory
help-index            ess-display-index
help-object           ess-display-help-on-object
vignettes             ess-display-vignettes
objects[ls]           ess-execute-objects
search                ess-execute-search
```

```

set-width  ess-execute-screen-options
install.packages
            ess-install.packages
library    ess-library
setRepos   ess-setRepositories
sos        ess-sos

```

Handy commands: `ess-library`, `ess-install.packages`, etc - ask for item with completion and execute the correspond command. `ess-sos` is a interface to `findFn` function in package `sos`. If package `sos` is not found, ask user for interactive install.

- ESS: New dynamic mode line indicator: Process status is automatically reflected in all mode-lines of associated with the process buffers. Particularly useful for displaying debug status of `ess-tracebug` and developer status of `ess-developer` in all associated buffers.
- ESS: New `ess-completing-read` mechanism: ESS uses `ido` completions whenever possible. Variable `ess-use-ido` controls whether to use `ido` completion or not. Active by default.
- ESS now supports `comint` fields for output and input detection. This feature is not used by default, but might be useful in the future.
- ESS[S]: New custom variable `inferior-ess-S-prompt` to customize prompt detection regular expression in the inferior ESS buffers. You can customize this variable to enhance `comint` navigation (`comint-previous-prompt` and `comint-next-prompt`) the inferior buffers.
- ESS[R]: Internal R completion retrieval (`ess-R-complete-object-name`) was rewritten and is faster now.
- ESS is using process plist to store process specific variables, as opposed to buffer local variables as it was using before. The use of buffer local variables to store process variables is discouraged.
- ESS: new functions to manipulate process plists: `ess-process-get` and `ess-process-set`.
- ESS: Internal process waiting mechanism was completely rewritten. ESS no more relies on prompt regular expressions for the prompt detection. The only requirement on the primary process prompt is to end in `>`. This could be overwritten by setting `inferior-ess-primary-prompt`.
- ESS[S], notably R: Saved command history: `ess-history-file` now accepts `t` (default), `nil`, or a file name. By setting it to `nil` no command line history is saved anymore. `ess-history-directory` now allows to have the history all saved in one “central” file.
- ESS[R]: more Roxygen improvements.
- ESS[R]: `C-c .` to set (indentation) style.
- ESS[R]: Functions with non-standard names (for example `'aaa-bbb:cc'`) are properly handled by font-lock and evaluation routines.
- ESS[R]: Several regexp bugs (described in `etc/R-ESS-bugs.el`) were fixed in `ess-get-words-from-vector` and `ess-command`.

Changes/New Features in 5.14:

- ESS[BUGS/JAGS]: Batch BUGS is back! For recent OpenBUGS versions, 3.0.8+, a batch BUGS script is once again available, but for Linux only. Therefore, since it seems that BUGS and JAGS must co-exist (rather than a transition from BUGS to JAGS), .bug files are now in ESS[BUGS] mode and .jag files are in ESS[JAGS] mode. ESS[BUGS] now works like ESS[JAGS] rather than the original mode ESS[BUGS] mode which was difficult to maintain. Although, ESS[BUGS] appears to work, there still may be some features missing as well as bugs.
- ESS[R]: New customizable variable `ess-swv-plug-into-AUCTeX-p` Commands to Sweave current file and LaTeX the result are now available to AUCTeX users, if this variable is set to `t`.
- ESS[S]: `C-c C-c` (`ess-eval-function-or-paragraph-and-step`) is now skipping over comments as the other paragraph functions do. It (and similar functions) should no longer wrongly find ‘`function()`’ beginnings inside comments or strings.
- ESS[SAS]: improved by better support for GNU Emacs

Changes/New Features in 5.13:

- ESS[R]: On Windows, for R 2.12.0 and later, the Rterm executables (in subdirectories i386 / x64) now are found as well as for earlier R versions.
- ESS[S+]: on Windows, both 32- and 64-bit versions of S+ (“S-PLUS”) are found now and made available on the menu.
- ESS[R]: When prompting for a starting directory, the R version is (always?) correct now.
- ESS[R]: on non-Windows platforms, the `use-dialog-box` variable is no longer temporarily changed (to `nil` for R-x.y.z version functions and to `t` for R itself), but rather the user customization is obeyed.
- ESS[R]: more Roxygen improvements.
- ‘Rd-preview-help’ now generates preview buffers with navigation facilities the same as regular help buffers.
- ESS: New functions and keys `C-c [up]` / `[down]` for evaluating the buffer “from beginning till here”.

Changes/New Features in 5.12:

- ESS[SAS] Font-locking: update of PROCs keywords (up to SAS 9.22); error/warnings.
- ESS[R]: Roxygen improvements: S4 classes; also optionally keep spaces when filling arguments
- ESS[Rd]: support new keywords: section-name \subsection plus a dozen “new” keywords; should match R 2.12.x now.
- `ess-display-help-on-object` (`C-c C-v`) now *caches* the list of topics, thus speeding up the improvement feature introduced in 5.9.

Changes/New Features in 5.11:

- Filename completion within buffers now adds only trailing characters to complete the filename, rather than expanding to an absolute file path. This filename completion is bound to the TAB key.

- *M-n P* in Sweave buffers now prompts for the command to run instead of using `pdflatex` unconditionally, offering completion from customizable collection `ess-svw-pdflatex-commands`, the first of which is taken as default and that defaults to `texi2pdf`.
- *M-RET* is now also bound in S language (R and S+) buffers to `ess-use-this-dir`. It sends `setwd(..)` to the S process to set the working directory to the one of the source file.

Changes/New Features in 5.10:

- *M-RET* in **S** buffers is now bound to `ess-dirs`. This function will set Emacs's current directory to be the same as the **S** process. This is useful if you use `setwd()` within a **S** process.

Changes/New Features in 5.9:

- Toolbar: The toolbar now has an icon for starting Splus.
- Indentation: New documentation and code has been added to make it easier to change how ESS indents code. In particular, see `ess-default-style`, `ess-own-style-list` and the documentation subsection “Changing indentation styles”.
- `ess-display-help-on-object` (*C-c C-v*) now offers completion candidates for help file aliases, in addition to object names.
- Font locking: is now turned on even without `window-system` is `nil`, whenever `ess-font-lock-mode` is non-`nil`, i.e., by default.
- ESS script editing: `ess-eval-deactivate-mark` default is now `t`, as suggested by Leo Alekseyev and subsequent “unanimous” ESS-help discussion.
- ESS[R]: Editing support for “#!” (Rscript / littler) editing, thanks to Jeffrey Arnold.
- ESS[R]: Now finds all R versions, both 64-bit and 32-bit, on some 64-bit Windows machines. Please report back to `ess-core` success or failure on your 64-bit Windows machine.
- ESS Manual now more visually pleasing; <http://ess.r-project.org/Manual/ess.html>
- ESS[R]: Roxygen on XEmacs no longer font locks for now (as it required missing features and hence broke ESS startup, there).
- ESS[R]: Roxygen has a sub-menu on the [ESS] menu.
- ESS[R]: Function `ess-rutils-htmldocs` in ‘`ess-rutils.el`’ offers an alternative to `help.start()` for navigating R documentation, using the `browse-url` Emacs function.

Changes/New Features in 5.8:

- ESS[R]: New ‘`ess-rutils.el`’ with utilities for listing, loading, installing, and updating packages, as well as object manipulation (listing, viewing, and deleting). It also provides an alternative to `RSiteSearch()` that uses the `browse-url` function, so results can be viewed in an Emacs web browser.
- ESS[R]: much more extensive Roxygen interface, via `ess-roxy.el` from Henning Røedsteg. `Ess-roxy` supports filling of roxygen fields, generation and updating roxygen templates, completion of roxygen tags, basic navigation (marking and moving between entries), folding using `hs-minor-mode` and preview of the Rd file.

- Emacs lisp files have got better names (partly, for now).

Changes/New Features in 5.7:

- ESS[R]: loading a source file (`C-c C-l`) now works in Windows, similarly to other platforms; (further; it had accidentally been broken in ESS 5.6 on all platforms)

Changes/New Features in 5.6:

- ESS[R]: `help()` calls have to differ from old default, with newer versions of R; currently via `.help.ESS <- function(...)` hack.

Changes/New Features in 5.4:

- ESS[SAS]: The long overdue change from `make-regexp` to `regexp-opt` for font-locking is complete. The new `regexp-opt` is now the default since it is better than the old code in many ways (and especially more maintainable). However, there are certainly some special cases missed (bug reports and patches welcome!). Setting `ess-sas-run-regexp-opt` to `nil` will result in the old code being used.
- ESS[BUGS] and ESS[JAGS]: typing `=` now results in `<-`.
- ESS[R] function arguments “show” (`ess-r-args-show`) now uses the new (`tooltip-show-at-point`) contributed by Erik Iverson.
- Toolbar icons now also work in (beta) Emacs 23.
- ESS[S]: New function `ess-change-directory` for setting both emacs’ current directory and the directory of an `*R*` or `*S*` buffer.
- ESS[S] when `transient-mark-mode` is true, the mark is now kept, rather than deactivated, thanks to a patch from David Reitter.

Changes/New Features in 5.3.11:

- ESS[SAS]: work around bug in Emacs 22.2 & 22.3 which fails to set case-fold fontification automatically.
- Rd mode: support new keyword ‘`Rdversion`’
- ESS[R]: now again works with Emacs 21.x

Changes/New Features in 5.3.10:

- Fixed noweb-mode bug accidentally introduced into 5.3.9
- In noweb-mode, e.g., Rnw-mode, electric “`<`” also inserts closing “`@`”. Further, the code chunk boundaries are better kept up-to-date, such that `code[R] <-> text[LaTeX]` minor mode switching should happen more reliably.
- In noweb-mode, fix a buglet in rare [Enter] or [Tab] behavior; further, by default disable the former ‘`[‘ .. ‘]`’ code-protection-when-filling behavior which has been found to be buggy.

Changes/New Features in 5.3.9:

- ESS[SAS]: evince PDF viewer now supported as well; search order: evince, Xpdf, Adobe/Acrobat Reader
- ESS[R]: added support for Roxygen, potentially to be extended.
- ESS[S] (and R): inferior (`*R*`) and transcript modes no longer fontify language keywords (such as `for`, `in`, etc).

- iESS[Stata]: Customize the `ess-sta-delimiter-friendly` setting to `t` to convert embedded semi-colons to newlines for Stata processing.
- Sweave fix for embedded blanks in PDF reader and PDF files
- Several fixes for Major Mode Convention violations in `ess-mode` and `noweb-mode`.
- ESS[JAGS]: `M-x comment-region` now available!
- ESS[S]: The `ess-swv-*` commands (and keybindings) are now in a submenu of the “Noweb” menu, when editing Sweave files.

Changes/New Features in 5.3.8:

- ESS[JAGS]: more separation from ESS[BUGS] (as much as is currently planned); now `C-c C-c` on an empty `.jmd` creates a template as it should; symbolic links are created for CODA output so BOA is happy: from `index.txt` to `.ind` and `chain#.txt` to `#.out`
- ESS[SAS]: buffer-local `ess-sas-submit-command` and `ess-sas-submit-command-options` now recognized by `ess-sas-submit-region`
- ESS[S]: When trying to evaluate code in an S language buffer and there is no associated process, now start R automatically instead of signalling an error. Also, restart R if there is an associated process which is not running. However, do not start R just via the “electric” (`ess-r-args-auto-show`).
- ESS[S]: For (one-line) functions withOUT `{ .. }` bodys, the end of function is now correctly found more often. This notably improves `C-c C-c` (`ess-eval-function-or-paragraph-and-step`).
- ESS[JAGS]: cleanup/re-organization of elisp code; symbolic links for CODA output are now only created by the new JAGS `system` command in version 1.0.3; specify whether this command is available via `ess-jags-system`; if not present, then no links are created so that the `*shell*` buffer does not become unresponsive during the batch run

Changes/New Features in 5.3.7:

- ESS: `ess-default-style` now `*is*` customizable, i.e., changing its value in `~/ .emacs` now does have the desired effect.
- ESS: `ess-font-lock-mode` is a new variable (default: `t`) which controls whether font-locking is enabled in ESS buffers.
- ESS[R]: for XEmacs on Windows; another tweak to find R versions
- ESS[SAS]: font-locking updated for ODS and SAS Bayesian Procedures; a more consistent handling of SAS options by `ess-sas-submit-command-options` which is buffer-local; portable snooze for MS Windows via customizable `ess-sleep-for` (floats welcome); Xpdf now supported as a PDF viewer
- ESS[Rnw]: now also works with “emacs -nw” and Emacs 22.
- ESS[JAGS]: now requires JAGS 1.0 (see the new ESS for JAGS help section for more information): both need work; suggestions welcome
- ESS[R]: [TAB] completion now uses the R-internal completion mechanism (for R $\geq 2.5.0$).
- ESS[R], [S]: interpretation of “`_`” as assignment has been removed in `ess-continued-statement-p` for R and S.

- several internal code cleanups.
- ESS[R]: An experimental version of a new command `Rgui` on MS Windows to send lines directly from emacs to `Rgui` is available in file `lisp/essd-rgui.el`. Preliminary documentation is in file `doc/rgui-doc.txt`.

Changes/New Features in 5.3.6:

- ESS: for XEmacs, using “gnuclient” (without a “-q”) works for things like `fix()` after M-x `gnuserv-start` has been done.
- ESS[R]: M-x `R-newest` should now work in more situations on MS Windows, e.g., when R has been installed in a non-default “ProgramFiles” directory tree. In these cases, there’s no need to specify the name (and full path) of the R program anymore.
- ESS[R]: For XEmacs, startup (with new tooltip code) works again.

Changes/New Features in 5.3.5:

- ESS[R] a new defun is available, *M-x R-newest*, which will start the newest version of R that it can find on your system.
- ESS[R] add Sven Hartenstein’s “R function arguments tips” functionality, via new file ‘`../lisp/essd-r-args.el`’. Note that this includes an “electric “(” behavior inside R-mode which is *active by default* and can be customized via `ess-r-args-electric-paren`; i.e., use `(setq ess-r-args-electric-paren nil)` to turn it off. Further, `ess-r-args-show-as` allows to switch to the “tooltip” mode.
- ESS: functions `ess-get-pdf-viewer` and `*-ps-viewer`; built on new customizable variables `ess-pdf-viewer-pref` and `ess-ps-viewer-pref`; currently used in `ess-swv-PDF` and `*-PS`.
- ESS[R] Improved `ess-swv-PDF` to run pdf viewer only if `pdflatex` was ok
- ESS[R] Improved `ess-swv-weave` to start R automatically if none is running.
- ESS: Do no longer ask *which* ESS process to use if there is only one.

Changes/New Features in 5.3.4:

- ESS[R] now better work with `options(error=recover)`; and the new default of CHM help files on windows.
- ESS[R] some more cleanup in the “sweave” functions
- miscellaneous fixes

Changes/New Features in 5.3.3:

- ESS[S] fix buglet (5.3.2 only) which left command prompt in “execute buffer” and hence help files.
- new customizable variable `ess-display-buffer-reuse-frames` set to true (which changes default behavior) such that execution or help `*frames*` are reused.

Changes/New Features in 5.3.2:

- Classic BUGS now supported by `(require 'essd-bugs)` with ESS[BUGS] and JAGS by `(require 'essd-jags)` with ESS[JAGS]. But, only one of them can be used at a time since they don’t play nice together. Also, `C-c C-c` is now bound to the function `ess-bugs-next-action` (`F12` has been retired). And finally, note that ‘`ess1-bug.el`’ is deprecated and the replacement is ‘`ess1-bugs.el`’.

- ESS[R] Improved some of the “Sweave-make” functions (yet scarcely documented) in ‘`ess-swv.el`’.
- ESS[S] No longer mess with `.Last.value` (nor in other “languages”).

Changes/New Features in 5.3.1:

- See the docs for 2 ways to install ESS for XEmacs
 1. by uncommenting the XEmacs part of Section 1 of ‘`Makeconf`’ and performing `make install`
 2. by unpacking either ‘`ess-5.3.1.tgz`’ or ‘`ess-5.3.1.zip`’ into ‘`PREFIX/lib/xemacs/site-packages`’ on unix or ‘`PREFIX\XEmacs\site-packages`’ on windows
- ESS[R]: fixed bugs so that `Rterm.exe` can be found by XEmacs
- ESS[S]: `ess-toggle-S-assign-key` is slightly changed; in particular, the default `ess-assign-key` is now `C-x =`.
- ESS[R]: `M-x R-site-search` is a new (slightly experimental) utility similar to R’s `RSiteSearch(...)` but with the advantage of using Emacs’ preferred browser, see `browse-url-browser-function`

Changes/New Features in 5.3.0:

- ESS[BUGS]: sanely re-format statistical output, ‘`.bog`’, from scientific notation to numbers rounded with 4 decimal places with `M-x ess-bugs-sci-round-to-4-dp`.
- The keys for navigating among section headings in help buffers worked, but only for one language per session; they should now work for multiple languages. (They were also broken on Windows machines.)
- ESS[S] long standing buglets in the internal logic for loading lisp code on Windows. Particularly fixed behavior in help mode with S-plus GUI.
- New variable, `ess-use-inferior-program-name-in-buffer-name`, which enables using the executable name instead of the dialect name for R. Feature request.
- ESS[S] `ess-execute-screen-options` now also works correctly when there is more than one window **side-by-side** in the same frame and runs in the correct buffer also when there is more than one S buffer.
- iESS[S] new functions `ess-eval-paragraph-and-step` and `ess-eval-function-or-paragraph-and-step` are bound to keys `C-c C-p` and `C-c C-c` respectively and to the menu in ESS-mode; also bound in the help mode (for evaluating examples).
- ESS[S] new function `ess-toggle-S-assign-key` allows to assign the “`<-`” insertion to an arbitrary key.

1.11 Reporting Bugs

Please send bug reports, suggestions etc. to ESS-bugs@stat.math.ethz.ch

The easiest way to do this 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.

1.12 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@stat.math.ethz.ch; contributions to the list may be mailed to ess-help@stat.math.ethz.ch. 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 on Emacs and XEmacs.
- suggestions for improvements.
- announcements of new releases of ESS.
- posting small patches to ESS.

1.13 Authors

- [A.J. Rossini](#)
- [Richard M. Heiberger](#)
- [Kurt Hornik](#)
- [Martin Maechler](#)
- [Rodney A. Sparapani](#)
- [Stephen Eglen](#)
- [Sebastian P. Luque](#)
- [Henning Redestig](#)
- [Vitalie Spinu](#)