# Package 'RInno'

January 20, 2025

**Type** Package

**OS_type** windows

**Title** An Installation Framework for Shiny Apps

**Version** 1.0.1

**Maintainer** Jon Hill <jon.mark.hill@gmail.com>

**URL** www.ficonsulting.com

**BugReports** https://github.com/ficonsulting/RInno/issues

**Description**
Installs shiny apps packaged as stand-alone Electron apps using Inno Setup, an open source software that builds installers for Windows programs <http://www.jrsoftware.org/ishelp/>.

**License** GPL-3 | file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.3.0)

**Imports** curl, glue (>= 1.2.0), httr, installr, jsonlite, magrittr, methods, pkgbuild, remotes, rmarkdown, shiny, stringr , utils

**Suggests** knitr, stringi, covr, testthat

**VignetteBuilder** knitr

**RoxygenNote** 6.1.0

**NeedsCompilation** no

**Author** Jon Hill [aut, cre, cph],
W. Lee Pang [aut, cph] (DesktopDeployR project at https://github.com/wleepang/DesktopDeployR),
Hanjo Odendaal [ctb],
William Bradley [ctb],
Brent (Tom) Bailey [ctb],
Mikolaj Rybinski [ctb],
Chase Clark [ctb],
Damien Soukhavong [ctb],
Jonathan Godfrey [ctb] (https://github.com/ajrgodfrey),

Gábor Csárdi [aut],
Hadley Wickham [aut],
Winston Chang [aut],
Jim Hester [aut],
RStudio [cph],
Martin Morgan [aut],
Dan Tenenbaum [aut],
Mango Solutions [cph]

**Repository** CRAN

**Date/Publication** 2018-09-21 16:00:12 UTC

# Contents



---

`code_section` *Pascal script to check registry for R*

---

## Description

Modern Delphi-like Pascal adds a lot of customization possibilities to the installer. For examples, please visit Pascal Scripting Introduction.

## Usage

```
code_section(iss, R_version = paste0(">=", R.version$major, ".",
  R.version$minor))
```

## Arguments

| | |
|---|---|
| iss | Character vector which cumulatively becomes an Inno Setup Script (ISS). |
| R_version | R version to use. Supports inequalities. Defaults to: paste0(">=", R.version$major, '.', R.version$minor). |

## Details

This script checks the registry for R, so that R will only be installed if necessary.

## Value

Chainable character vector, which can be used as the text argument of writeLines to generate an ISS.

## Author(s)

Jonathan M. Hill

## See Also

get_R, copy_installation, create_config, create_bat, directives_section, setup_section, languages_section, tasks_section, files_section, icons_section, run_section, and code_section.

## Examples

```
## Not run:
readLines(system.file('installation/code.iss', package = 'RInno'))

## End(Not run)
```

---

# compile_iss *Compile ISS*

---

## Description

After running create_app and editing the content of the installer and app, call compile_iss.

## Usage

```
compile_iss()
```

### Value

Installer in `dir_out`.

### Author(s)

Jonathan M. Hill

---

`copy_installation` *Default installation files*

---

### Description

This function moves files stored in `system.file('installation', package = 'RInno')` to `app_dir`:

- Icons for installer and app, *setup.ico*, *default.ico* and *default.png*.
- Files that manage app start up, *utils/package_manager.R* and *utils/launch_app.R*.
- First/last page of the installation wizard, *infobefore.txt* and *infoafter.txt*.
- Batch support files, *utils/wsf/run.wsf*, *utils/wsf/js/run.js*, *utils/wsf/js/json2.js*, and *utils/wsf/js/JSON.minify.js*.

### Usage

```
copy_installation(app_dir = getwd(), overwrite = TRUE)
```

### Arguments

| | |
|---|---|
| `app_dir` | Development app's directory, defaults to `getwd()`. |
| `overwrite` | Logical. Should existing installation files be overwritten? See `copy_installation` for details. |

### Author(s)

Jonathan M. Hill

### See Also

`create_app`

create_app *Creates installation files and Inno Setup Script (ISS), "app_name.iss"*

## Description

This function manages installation and app start up. To accept all defaults, just provide app_name. After calling create_app, call compile_iss to create an installer in dir_out.

## Usage

```
create_app(app_name = "myapp", app_dir = getwd(),
  dir_out = "RInno_installer", pkgs = c("jsonlite", "shiny",
  "magrittr"), pkgs_path = "bin", repo = "https://cran.rstudio.com",
  remotes = "none", locals = NULL, app_repo_url = "none",
  auth_user = "none", auth_pw = "none", auth_token = github_pat(),
  user_browser = "electron", include_R = FALSE,
  include_Pandoc = FALSE, include_Chrome = FALSE,
  include_Rtools = FALSE, R_version = paste0(">=", R.version$major,
  ".", R.version$minor), Pandoc_version = rmarkdown::pandoc_version(),
  Rtools_version = "3.5", overwrite = TRUE, force_nativefier = TRUE,
  nativefier_opts = c(), ...)
```

## Arguments

| | |
|---|---|
| app_name | The name of the app. It will be displayed throughout the installer's window titles, wizard pages, and dialog boxes. See [Setup]:AppName for details. For continuous installations, app_name is used to check for an R package of the same name, and update it. The Continuous Installation vignette has more details. |
| app_dir | Development app's directory, defaults to getwd(). |
| dir_out | Installer's directory. A sub-directory of app_dir, which will be created if it does not exist. Defaults to 'RInno_installer'. |
| pkgs | Character vector of package dependencies. Remote development versions are supported via remotes. pkgs are downloaded into file.path(app_dir, pkgs_path) as Windows binary packages (.zip). If you build binary packages and store them there before calling create_app, they will be included as well. |
| pkgs_path | Default location inside the app working directory to install package dependencies This defaults to pkgs_path = "bin" |
| repo | Default repository to install CRAN package dependencies. Defaults to repo = "https://cran.rstudio.com". |
| remotes | Character vector of GitHub repository addresses in the format username/repo[/subdir][\@ref\|#pull] for GitHub package dependencies. |
| locals | Character vector of local package dependencies. Deprecated as of v1.0.0. Use pkgs instead. |
| app_repo_url | Repository address for continuous installations in the format "https://bitbucket.org/username/repo (repo = app_name). Only Bitbucket and GitHub repositories are supported. |

| | |
|---|---|
| auth_user | Bitbucket username. It is recommended to create a read-only account for each app. Support for OAuth 2 and tokens is in the works. |
| auth_pw | Bitbucket password matching the above username. |
| auth_token | To install from a private Github repo, generate a personal access token (PAT) in https://github.com/settings/tokens and supply to this argument. This is safer than using a password because you can easily delete a PAT without affecting any others. |
| user_browser | Character for the default browser. Options include "chrome", "firefox", and "ie." |
| include_R | To include R in the installer, include_R = TRUE. The version of R specified by R_version is used. The installer will check each user's registry and only install R if necessary. |
| include_Pandoc | To include Pandoc in the installer, include_Pandoc = TRUE. If installing a flex-dashboard app, some users may need a copy of Pandoc. The installer will check the user's registry for the version of Pandoc specified in Pandoc_version and only install it if necessary. |
| include_Chrome | To include Chrome in the installer, include_Chrome = TRUE. If you would like to use Chrome's app mode, it is no longer supported by Google :(. |
| include_Rtools | To include Rtools in the installer, include_Rtools = TRUE. For some packages to build properly, you may need to include Rtools. |
| R_version | R version to use. Supports inequalities. Defaults to: paste0(">=", R.version$major, '.', R.version$minor). |
| Pandoc_version | Pandoc version to use, defaults to: pandoc_available. |
| Rtools_version | Rtools version to include. For more information, see Building R for Windows. |
| overwrite | Logical. Should existing installation files be overwritten? See copy_installation for details. |
| force_nativefier | Boolean. Defaults to true and re-builds UI. If false, the UI is not rebuilt. |
| nativefier_opts | Character vector. Extra options to pass to nativefier when user_browser = "electron". Each string in the vector should be a valid nativefier command. For example, c('--no-overwrite', '--conceal', '--show-menu-bar'). For more information, system("nativefier --help"). |
| ... | Arguments passed on to setup_section, files_section, directives_section, icons_section, languages_section, code_section, tasks_section, and run_section. |

## Details

Creates the following files in app_dir:

- Icons for installer and app, *setup.ico* and *default.ico* respectively.
- Files that manage app start up, *utils/package_manager.R*, *utils/ensure.R*, and *utils/launch_app.R*.
- First/last page of the installer, *infobefore.txt* and *infoafter.txt*.
- Batch support files, *utils/wsf/run.wsf*, *utils/wsf/js/run.js*, *utils/wsf/js/json2.js*, *utils/wsf/js/JSON.minify.js*.

- A configuration file, *config.cfg*. See create_config for details.
- A batch file, *app_name.bat*. See create_bat for details.
- An Inno Setup Script, *app_name.iss*.

## Author(s)

Jonathan M. Hill and Hanjo Odendaal

## See Also

get_R, copy_installation, create_config, create_bat, directives_section, setup_section, languages_section, tasks_section, files_section, icons_section, run_section, and code_section.

## Examples

```
## Not run:

create_app('myapp')

create_app(
  app_name     = 'My AppName',
  app_dir      = 'My/app/path',
  dir_out      = 'wizard',
  pkgs         = c('jsonlite', 'shiny', 'magrittr', 'xkcd'),
  include_R    = TRUE,   # Download R and install it with the app
  R_version    = "2.2.1",  # Old version of R
  privilege    = 'high', # Admin only installation
  default_dir  = 'pf') # Program Files

## End(Not run)
```

---

# create_bat *Creates app's batch file, "app_name.bat"*

---

## Description

This procedure creates a batch file that starts a shiny app using `wsf/run.wsf`.

## Usage

```
create_bat(app_name, app_dir)
```

## Arguments

| | |
|---|---|
| app_name | The name of the app. It will be displayed throughout the installer's window titles, wizard pages, and dialog boxes. See [Setup]:AppName for details. For continuous installations, app_name is used to check for an R package of the same name, and update it. The Continuous Installation vignette has more details. |
| app_dir | Development app's directory, defaults to getwd(). |

## Value

BATCH file in app_dir

## Author(s)

Jonathan M. Hill

## See Also

create_app

---

create_config *Creates an app config file, "utils/config.cfg"*

---

## Description

Creates an app config file, "utils/config.cfg"

## Usage

```
create_config(app_name, app_dir = getwd(), pkgs = c("jsonlite",
  "remotes", "magrittr"), pkgs_path = "library", remotes = "none",
  repo = "https://cran.rstudio.com", error_log = "error.log",
  app_repo_url = "none", auth_user = "none", auth_pw = "none",
  auth_token = "none", user_browser = "electron")
```

## Arguments

| | |
|---|---|
| app_name | The name of the app. It will be displayed throughout the installer's window titles, wizard pages, and dialog boxes. See [Setup]:AppName for details. For continuous installations, app_name is used to check for an R package of the same name, and update it. The Continuous Installation vignette has more details. |
| app_dir | Development app's directory, defaults to getwd(). |
| pkgs | Character vector of package dependencies. Remote development versions are supported via remotes. pkgs are downloaded into file.path(app_dir, pkgs_path) as Windows binary packages (.zip). If you build binary packages and store them there before calling create_app, they will be included as well. |
| pkgs_path | Default location inside the app working directory to install package dependencies This defaults to pkgs_path = "bin" |
| remotes | Character vector of GitHub repository addresses in the format username/repo[/subdir][\@ref\|#pull] for GitHub package dependencies. |
| repo | Default repository to install CRAN package dependencies. Defaults to repo = "https://cran.rstudio.com". |
| error_log | Name of error logging file. Contains start up errors from *run.R*. |

| | |
|---|---|
| app_repo_url | Repository address for continuous installations in the format "https://bitbucket.org/username/repo (repo = app_name). Only Bitbucket and GitHub repositories are supported. |
| auth_user | Bitbucket username. It is recommended to create a read-only account for each app. Support for OAuth 2 and tokens is in the works. |
| auth_pw | Bitbucket password matching the above username. |
| auth_token | To install from a private Github repo, generate a personal access token (PAT) in https://github.com/settings/tokens and supply to this argument. This is safer than using a password because you can easily delete a PAT without affecting any others. |
| user_browser | Character for the default browser. Options include "chrome", "firefox", and "ie." |

## Value

A json file, *config.cfg*, in app_dir/utils.

## Author(s)

Jonathan M. Hill

## See Also

create_app.

---

# directives_section *Inno Setup Preprocessor (ISPP) Directives*

---

## Description

Sets ISPP directives at the top of an ISS.

## Usage

```
directives_section(app_name, include_R = FALSE,
  R_version = paste0(R.version$major, ".", R.version$minor),
  include_Pandoc = FALSE, Pandoc_version = rmarkdown::pandoc_version(),
  include_Chrome = FALSE, include_Rtools = FALSE,
  Rtools_version = "3.5", app_version = "0.0.0", publisher = "",
  main_url = "", custom_vars = "", custom_values = "")
```

## Arguments

| | |
|---|---|
| app_name | The name of the app. It will be displayed throughout the installer's window titles, wizard pages, and dialog boxes. See [Setup]:AppName for details. For continuous installations, app_name is used to check for an R package of the same name, and update it. The Continuous Installation vignette has more details. |

| | |
|---|---|
| include_R | To include R in the installer, include_R = TRUE. The version of R specified by R_version is used. The installer will check each user's registry and only install R if necessary. |
| R_version | R version to use. Supports inequalities. Defaults to: paste0(">=", R.version$major, '.', R.version$minor). |
| include_Pandoc | To include Pandoc in the installer, include_Pandoc = TRUE. If installing a flexdashboard app, some users may need a copy of Pandoc. The installer will check the user's registry for the version of Pandoc specified in Pandoc_version and only install it if necessary. |
| Pandoc_version | Pandoc version to use, defaults to: pandoc_available. |
| include_Chrome | To include Chrome in the installer, include_Chrome = TRUE. If you would like to use Chrome's app mode, it is no longer supported by Google :(. |
| include_Rtools | To include Rtools in the installer, include_Rtools = TRUE. For some packages to build properly, you may need to include Rtools. |
| Rtools_version | Rtools version to include. For more information, see Building R for Windows. |
| app_version | Version number of the app being installed, defaults to '0.0.0'. It is displayed in the Version field of the app's *Add/Remove Programs* entry. See [Setup]:AppVersion for details. |
| publisher | String displayed on the "Support" dialogue of the *Add/Remove Programs* Control Panel applet, defaults to " ". See [Setup]:AppPublisher for details. |
| main_url | String. Defaults to "". |
| custom_vars | String vector. Defaults to "", and must be the same length as custom_values. |
| custom_values | String vector of values for custom_vars. Defaults to "", and must be the same length as custom_vars. |

## Details

ISPP directives automate compile-time tasks and decrease the probability of typos. When referring to an ISPP directive, use '{#var_name}'. For more information, call ispp_doc() or visit ISPP Help.

custom_vars and custom_values utilize the #define directive.

## Value

Chainable character vector, which can be used as the text argument of writeLines to generate an ISS.

## Author(s)

Jonathan M. Hill

## See Also

get_R, copy_installation, create_config, create_bat, directives_section, setup_section, languages_section, tasks_section, files_section, icons_section, run_section, and code_section.

## Examples

```
## Not run:
start_iss('myapp') %>%
  directives_section(
    include_R = FALSE, R_version = '3.3.2',
    custom_vars = 'helpers',
    custom_values = 'path\\to\\helpers') %>%
  files_section(
    app_dir = getwd(),
    file_list = '{#helpers}')

## End(Not run)
```

---

# download_packages *Download packages*

---

## Description

Places package dependencies in pkgs_path.

## Usage

```
download_packages(app_dir, pkgs_path, pkgs, repo, remotes, auth_user,
  auth_token)
```

## Arguments

| | |
|---|---|
| app_dir | Development app's directory, defaults to getwd(). |
| pkgs_path | Default location inside the app working directory to install package dependencies This defaults to pkgs_path = "bin" |
| pkgs | Character vector of package dependencies. Remote development versions are supported via remotes. pkgs are downloaded into file.path(app_dir, pkgs_path) as Windows binary packages (.zip). If you build binary packages and store them there before calling create_app, they will be included as well. |
| repo | Default repository to install CRAN package dependencies. Defaults to repo = "https://cran.rstudio.com". |
| remotes | Character vector of GitHub repository addresses in the format username/repo[/subdir][\@ref\|#pull] for GitHub package dependencies. |
| auth_user | Bitbucket username. It is recommended to create a read-only account for each app. Support for OAuth 2 and tokens is in the works. |
| auth_token | To install from a private Github repo, generate a personal access token (PAT) in https://github.com/settings/tokens and supply to this argument. This is safer than using a password because you can easily delete a PAT without affecting any others. |

example_app *Example app*

## Description

Creates a basic app to test in wd/app_dir.

## Usage

```
example_app(app_dir = "app", wd = getwd(), type = "Shiny")
```

## Arguments

| | |
|---|---|
| app_dir | Shiny app's directory. Defaults to "app". |
| wd | Path to working directory. Defaults to getwd(). |
| type | "Shiny" or "flexdashboard". Defaults to "Shiny". |

## Value

Shiny app example.

## Author(s)

Jonathan M. Hill

## Examples

```
## Not run:
# Shiny example
example_app()
create_app("myapp", "app")

# Flexdashboard example
example_app(type = "flexdashboard")
create_app("myapp", "app")

## End(Not run)
```

## files_section *Files Section of ISS*

### Description

Files to be installed on user's computer. Everything in `app_dir` plus `file_list`. For more information, visit [Files] section.

### Usage

```
files_section(iss, app_name, app_dir, user_browser,
  file_list = character())
```

### Arguments

| | |
|---|---|
| `iss` | Character vector which cumulatively becomes an Inno Setup Script (ISS). |
| `app_name` | The name of the app. It will be displayed throughout the installer's window titles, wizard pages, and dialog boxes. See [Setup]:AppName for details. For continuous installations, `app_name` is used to check for an R package of the same name, and update it. The Continuous Installation vignette has more details. |
| `app_dir` | Development app's directory, defaults to `getwd()`. |
| `user_browser` | Character for the default browser. Options include "chrome", "firefox", and "ie." |
| `file_list` | Character vector. Extra files to be installed with the app. |

### Value

Chainable character vector, which can be used as the `text` argument of `writeLines` to generate an ISS.

### Author(s)

Jonathan M. Hill

## get_Chrome *Downloads Chrome*

### Description

Downloads Chrome in `app_dir`. If Chrome has already been downloaded, `get_Chrome` will use that file. If the download fails it will stop.

### Usage

```
get_Chrome(app_dir)
```

### Arguments

| | |
|---|---|
| app_dir | Development app's directory, defaults to getwd(). |

### Details

If create_app(include_Chrome = TRUE), then get_Chrome.

### Value

chrome_installer.exe in app_dir.

### Author(s)

Jonathan M. Hill

### See Also

get_R, copy_installation, create_config, create_bat, directives_section, setup_section, languages_section, tasks_section, files_section, icons_section, run_section, and code_section.

---

## get_Pandoc *Downloads Pandoc*

---

### Description

Downloads Pandoc in app_dir. If Pandoc has already been downloaded, get_Pandoc will use that file. If the download fails it will stop.

### Usage

```
get_Pandoc(app_dir, Pandoc_version = rmarkdown::pandoc_version())
```

### Arguments

| | |
|---|---|
| app_dir | Development app's directory, defaults to getwd(). |
| Pandoc_version | Pandoc version to use, defaults to: pandoc_available. This ensures that the same version of Pandoc used during development is installed on users' computers. |

### Details

If create_app(include_Pandoc = TRUE), then get_Pandoc.

### Value

sprintf("pandoc-%s-windows.msi", Pandoc_version) in app_dir.

### Author(s)

Jonathan M. Hill and Hanjo Odendaal

### See Also

get_R, copy_installation, create_config, create_bat, directives_section, setup_section, languages_section, tasks_section, files_section, icons_section, run_section, and code_section.

---

## get_R *Downloads R*

---

### Description

Downloads R in app_dir. If it has already been downloaded, get_R will use that file. If the download fails it will stop.

### Usage

```
get_R(app_dir = getwd(), R_version = paste0(">=", R.version$major, ".",
  R.version$minor))
```

### Arguments

| | |
|---|---|
| app_dir | Development app's directory, defaults to getwd(). |
| R_version | R version to use. Supports inequalities. Defaults to: paste0(">=", R.version$major, '.', R.version$minor). |

### Details

If create_app(include_R = TRUE), then get_R.

### Value

sprintf('R-%s-win.exe', R_version) in app_dir.

### Author(s)

Jonathan M. Hill

### See Also

get_R, copy_installation, create_config, create_bat, directives_section, setup_section, languages_section, tasks_section, files_section, icons_section, run_section, and code_section.

get_Rtools *Downloads Rtools*

## Description

Downloads Rtools in app_dir. If it has already been downloaded, get_Rtools will use that file. If the download fails it will stop.

## Usage

```
get_Rtools(app_dir, Rtools_version, R_version)
```

## Arguments

| | |
|---|---|
| app_dir | Development app's directory, defaults to getwd(). |
| Rtools_version | Rtools version to include. For more information, see Building R for Windows. |
| R_version | R version to use. Supports inequalities. Defaults to: paste0(">=", R.version$major, '.', R.version$minor). |

## Details

If create_app(include_Rtools = TRUE), then get_Rtools.

## Value

sprintf('Rtools%s.exe', gsub("\.", "", Rtools_version)) in app_dir.

## Author(s)

Jonathan M. Hill

## See Also

get_R, copy_installation, create_config, create_bat, directives_section, setup_section, languages_section, tasks_section, files_section, icons_section, run_section, and code_section.

`icons_section` *Icons Section of ISS*

## Description

Shortcuts Inno Setup creates in the Start Menu and/or other locations, such as the desktop. For more information, see [Icons] section, or call `inno_doc()`.

## Usage

```
icons_section(iss, app_dir, app_desc = "", app_icon = "default.ico",
  prog_menu_icon = TRUE, desktop_icon = TRUE)
```

## Arguments

| | |
|---|---|
| `iss` | Character vector which cumulatively becomes an Inno Setup Script (ISS). |
| `app_dir` | Development app's directory, defaults to `getwd()`. |
| `app_desc` | Description of Shiny app, appears on mouse-over of icons. |
| `app_icon` | Filename of icon in `app_dir`, used for desktop and program menu shortcuts. |
| `prog_menu_icon` | Logical. If TRUE, create a program menu shortcut. |
| `desktop_icon` | Logical. If TRUE, create a desktop shortcut. |

## Value

Chainable character vector, which can be used as the `text` argument of `writeLines` to generate an ISS.

## Author(s)

Jonathan M. Hill

## See Also

`get_R`, `copy_installation`, `create_config`, `create_bat`, `directives_section`, `setup_section`, `languages_section`, `tasks_section`, `files_section`, `icons_section`, `run_section`, and `code_section`.

## Examples

```
## Not run:
start_iss('myapp') %>%
  icons_section(app_desc = 'This Shiny app is awesome!')

## End(Not run)
```

`install_inno` *Downloads and installs Inno Setup*

## Description

Downloads and installs Inno Setup's stable release

## Usage

```
install_inno(quick_start_pack = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `quick_start_pack` | The Inno Setup QuickStart Pack includes Inno Setup and Inno Script Studio script editor. See Third-Party Files page for more information. |
| `...` | extra parameters to pass to `install.URL` |

## Details

Inno Setup is a free installer for Windows programs. First introduced in 1997, it currently rivals many commercial installers in feature set and stability.

See Features for more information.

## Value

TRUE/FALSE - was the installation successful or not.

## Author(s)

Tal Galili and Jonathan M. Hill

## Examples

```
## Not run:
install_inno()
install_inno(quick_start_pack = T)

## End(Not run)
```

`install_nodejs` *Downloads and installs nodejs*

## Description

Suports Nodejs's "current" and "lts" versions - LTS - Current

## Usage

```
install_nodejs(page_with_download_url = "https://nodejs.org/en/download/",
version = "LTS", ...)
```

## Arguments

| | |
|---|---|
| `page_with_download_url` | nodejs download url. |
| `version` | character. "current" or "lts". Defaults to "lts" |
| `...` | extra parameters to pass to `install.URL` |

## Details

As an asynchronous event driven JavaScript runtime, Node is designed to build scalable network applications.

See About for more information.

## Value

TRUE/FALSE - was the installation successful or not.

## Author(s)

Tal Galili, A. Jonathan R. Godfrey, and Jonathan M. Hill

## Examples

```
## Not run:
install_nodejs()
install_nodejs(version = "current")

## End(Not run)
```

## languages_section *Languages Section of ISS*

### Description

RInno currently supports 25 languages. Check the `languages` directory of Inno Setup for a complete list, and see [Languages] section for details.

### Usage

```
languages_section(iss, language = "english")
```

### Arguments

| | |
|---|---|
| `iss` | Character vector which cumulatively becomes an Inno Setup Script (ISS). |
| `language` | Character vector of lower case languages to include. |

### Value

Chainable character vector, which can be used as the `text` argument of `writeLines` to generate an ISS.

### Author(s)

Jonathan M. Hill

## nativefy_app *Package app into electron with nativefier*

### Description

Package app into electron with nativefier

### Usage

```
nativefy_app(app_name, app_dir, nativefier_opts,
  app_icon = "default.ico")
```

### Arguments

| | |
|---|---|
| app_name | The name of the app. It will be displayed throughout the installer's window titles, wizard pages, and dialog boxes. See [Setup]:AppName for details. For continuous installations, app_name is used to check for an R package of the same name, and update it. The Continuous Installation vignette has more details. |
| app_dir | Development app's directory, defaults to getwd(). |
| nativefier_opts | Character vector. Extra options to pass to nativefier when user_browser = "electron". Each string in the vector should be a valid nativefier command. For example, c('--no-overwrite', '--conceal', '--show-menu-bar'). For more information, system("nativefier --help"). |
| app_icon | Filename of icon in app_dir, used for desktop and program menu shortcuts. |

---

## run_section *Run Section of ISS*

---

### Description

Specifies any number of programs to execute after the program has been successfully installed, but before the installer displays the final dialog. See [Run] for details.

### Usage

```
run_section(iss, R_flags = "/SILENT")
```

### Arguments

| | |
|---|---|
| iss | Character vector which cumulatively becomes an Inno Setup Script (ISS). |
| R_flags | String of flags to customize R's installation. Defaults to "/SILENT". For other options, visit Section 2.4 of the R FAQ. If using the '/DIR=""C:\myapp""' flag, use double backslashes and double quotes. For more information on valid Inno Setup constants, see the Constants section. |

### Value

Chainable character vector, which can be used as the text argument of writeLines to generate an ISS.

### Author(s)

Jonathan M. Hill

### See Also

get_R, copy_installation, create_config, create_bat, directives_section, setup_section, languages_section, tasks_section, files_section, icons_section, run_section, and code_section.

## Examples

```
## Not run:
# You can combine custom R installation flags with Inno Setup constants
create_app("myapp", "app", R_flags = '/SILENT /DIR=""{userdocs}""')

# Or directly
run_section(iss, R_flags = '/SILENT /DIR=""{userdocs}""')


## End(Not run)
```

---

`setup_section` *Setup Section of ISS*

---

## Description

This section contains global settings used by the installer and uninstaller. See [Setup] for details.

## Usage

```
setup_section(iss, app_dir, dir_out, app_version = "{#MyAppVersion}",
  name = "{#MyAppName}", publisher = "{#MyAppPublisher}",
  default_dir = "userdocs", privilege = "lowest",
  info_before = "infobefore.txt", info_after = "infoafter.txt",
  license_file = "none", setup_icon = "setup.ico", inst_pw = "none",
  pub_url = "{#MyAppURL}", sup_url = "{#MyAppURL}",
  upd_url = "{#MyAppURL}", compression = "lzma2/ultra64")
```

## Arguments

| | |
|---|---|
| `iss` | Character vector which cumulatively becomes an Inno Setup Script (ISS). |
| `app_dir` | Development app's directory, defaults to `getwd()`. |
| `dir_out` | Installer's directory. A sub-directory of `app_dir`, which will be created if it does not exist. Defaults to 'RInno_installer'. |
| `app_version` | Version number of the app being installed, defaults to `'0.0.0'`. It is displayed in the Version field of the app's *Add/Remove Programs* entry. See [Setup]:AppVersion for details. |
| `name` | Defaults to ISPP directive, `'{#MyAppName}'` set by `directives(app_name)`. |
| `publisher` | String displayed on the "Support" dialogue of the *Add/Remove Programs* Control Panel applet, defaults to " ". See [Setup]:AppPublisher for details. |
| `default_dir` | The default directory name used by the *Select Destination Page* of the installer. See [Setup]:DefaultDirName and Constants for details. |
| `privilege` | Valid options: `'poweruser'`, `'admin'`, `'lowest'`. Defaults to `'lowest'`. This directive affects whether elevated rights are requested when an installation is started. See [Setup]:PrivilegesRequired for details. |

`info_before` File, in .txt or .rtf format, which is displayed on the first page of the installer. It must be located in `app_dir`. See [Setup]:InfoBeforeFile for details.

`info_after` File, in .txt or .rtf format, which is displayed on the last page of the installer. It must be located in `app_dir`. See [Setup]:InfoAfterFile for details.

`license_file` File, in .txt or .rtf format, which is displayed before the *Select Destination Page* of the wizard. See [Setup]:LicenseFile for details.

`setup_icon` File name of the icon used for installer/uninstaller. The file must be located in `app_dir`. See [Setup]:SetupIconFile for details.

`inst_pw` Installer password, string. Visit the Inno Setup Downloads page and place *ISCrypt.dll* in your Inno Setup directory. Afterwards, if a `inst_pw` is supplied, then the contents of the installer will be encrypted using a 160-bit key derived from the password string. See [Setup]:Password and [Setup]:Encryption for details.

`pub_url` String. Defaults to `'{#MyAppURL}'`, which is the ISPP directive for `main_url`. Therefore, `main_url` will be used, unless otherwise specified. See [Setup]:AppPublisherURL for details.

`sup_url` String. Defaults to `'{#MyAppURL}'`, which is the ISPP directive for `main_url`. Therefore, `main_url` will be used, unless otherwise specified. See [Setup]:AppSupportURL for details.

`upd_url` String. Defaults to `'{#MyAppURL}'`, which is the ISPP directive for `main_url`. Therefore, `main_url` will be used, unless otherwise specified. See [Setup]:AppUpdatesURL for details.

`compression` Defaults to `'lzma2/ultra64'`, which has the best compression ratio available. Other valid options include: `'zip'`, `'bzip'`, `'lzma'`, and `'none'`. See [Setup]:Compression for details.

## Value

Chainable character vector, which can be used as the `text` argument of `writeLines` to generate an ISS.

## Author(s)

Jonathan M. Hill

## See Also

`get_R`, `copy_installation`, `create_config`, `create_bat`, `directives_section`, `setup_section`, `languages_section`, `tasks_section`, `files_section`, `icons_section`, `run_section`, and `code_section`.

## Examples

```
## Not run:
start_iss('myapp') %>%
  directives_section(
    include_R = FALSE, R_version = '3.3.2') %>%
  setup_section(
```

```
    dir_out = 'installer', default_dir = 'pf')

## End(Not run)
```

---

`start_iss` *Start ISS*

---

## Description

Chain `directives_section` against this function to start building custom installers.

## Usage

```
start_iss(app_name)
```

## Arguments

| | |
|---|---|
| `app_name` | The name of the app. It will be displayed throughout the installer's window titles, wizard pages, and dialog boxes. See [Setup]:AppName for details. For continuous installations, `app_name` is used to check for an R package of the same name, and update it. The Continuous Installation vignette has more details. |

## Value

`app_name` and set `options('RInno.app_name' = app_name)`

## Author(s)

Jonathan M. Hill

## See Also

`directives_section`.

## Examples

```
## Not run:
start_iss('myapp') %>%
  directives_section(
    include_R = FALSE, R_version = '3.3.2')

## End(Not run)
```

## tasks_section *Tasks Section of ISS*

### Description

Defines all of the user-customizable tasks during installation. These tasks appear as check boxes and radio buttons on the *Select Additional Tasks* installer page. See [Tasks] section for details.

### Usage

```
tasks_section(iss, desktop_icon = TRUE)
```

### Arguments

| | |
|---|---|
| iss | Character vector which cumulatively becomes an Inno Setup Script (ISS). |
| desktop_icon | Logical. If TRUE, create a desktop shortcut. |

### Value

Chainable character vector, which can be used as the text argument of writeLines to generate an ISS.

### Author(s)

Jonathan M. Hill

### See Also

get_R, copy_installation, create_config, create_bat, directives_section, setup_section, languages_section, tasks_section, files_section, icons_section, run_section, and code_section.

## %>% *magrittr Pipes*

### Description

magrittr Pipes

### Usage

```
lhs %>% rhs
```

### Arguments

| | |
|---|---|
| lhs | A value or the magrittr placeholder. |
| rhs | A function call using the magrittr semantics. |

## See Also

magrittr

# Index