


```

" Add an UnBundle command {
function! UnBundle(arg, ...)
    let bundle = vundle#config#init_bundle(a:arg, a:000)
    call filter(g:vundle#bundles, 'v:val["name_spec"] != "" . a:arg . "')
endfunction

com! -nargs=+          UnBundle
\ call UnBundle(<args>)
" }

" }

" Bundles {
Bundle 'supertab'
" Deps {
    Bundle 'gmarik/vundle'
    Bundle 'MarcWeber/vim-addon-mw-utils'
    Bundle 'tomtom/tlib_vim'
    if executable('ag')
        Bundle 'mileszs/ack.vim'
        let g:ackprg = 'ag --nogroup --nocolor --column --smart-case'
    elseif executable('ack-grep')
        let g:ackprg="ack-grep -H --nocolor --nogroup --column"
        Bundle 'mileszs/ack.vim'
    elseif executable('ack')
        Bundle 'mileszs/ack.vim'
    endif
" }

" In your .vimrc.before.local file
" list only the plugin groups you will use
if !exists('g:spf13_bundle_groups')
    let g:spf13_bundle_groups=['general', 'writing', 'neocomplete',
'programming', 'php', 'ruby', 'python', 'javascript', 'html', 'misc',]
endif

" To override all the included bundles, add the following to your
" .vimrc.bundles.local file:
" let g:override_spf13_bundles = 1
if !exists("g:override_spf13_bundles")

" General {
    if count(g:spf13_bundle_groups, 'general')
        Bundle 'scrooloose/nerdtree'
        Bundle 'altercation/vim-colors-solarized'
        Bundle 'spf13/vim-colors'
        Bundle 'tpope/vim-surround'
        Bundle 'tpope/vim-repeat'
        Bundle 'rhysd/conflict-marker.vim'
        Bundle 'jiangmiao/auto-pairs'
        Bundle 'ctrlpvim/ctrlp.vim'
        Bundle 'tacahiroy/ctrlp-funky'
        Bundle 'terryma/vim-multiple-cursors'
        Bundle 'vim-scripts/sessionman.vim'
        Bundle 'matchit.zip'
        if (has("python") || has("python3")) &&
exists('g:spf13_use_powerline') && !exists('g:spf13_use_old_powerline')
            Bundle 'Lokaltog/powerline', {'rtp': '/powerline/bindings/vim'}
        elseif exists('g:spf13_use_powerline') &&
exists('g:spf13_use_old_powerline')
            Bundle 'Lokaltog/vim-powerline'
        else
            Bundle 'vim-airline/vim-airline'
            Bundle 'vim-airline/vim-airline-themes'

```

```

endif
Bundle 'powerline/fonts'
Bundle 'bling/vim-bufferline'
Bundle 'easymotion/vim-easymotion'
Bundle 'jistr/vim-nerdtree-tabs'
Bundle 'flazz/vim-colorschemes'
Bundle 'mbbill/undotree'
Bundle 'nathanaelkane/vim-indent-guides'
if !exists('g:spf13_no_views')
    Bundle 'vim-scripts/restore_view.vim'
endif
Bundle 'mhinz/vim-signify'
Bundle 'tpope/vim-abolish.git'
Bundle 'osyo-manga/vim-over'
Bundle 'kana/vim-textobj-user'
Bundle 'kana/vim-textobj-indent'
Bundle 'gcmt/wildfire.vim'
endif
" }

" Writing {
    if count(g:spf13_bundle_groups, 'writing')
        Bundle 'reedes/vim-litecorrect'
        Bundle 'reedes/vim-textobj-sentence'
        Bundle 'reedes/vim-textobj-quote'
        Bundle 'reedes/vim-wordy'
        Bundle 'xolox/vim-notes'
    endif
" }

" General Programming {
    if count(g:spf13_bundle_groups, 'programming')
        " Pick one of the checksyntax, jslint, or syntastic
        Bundle 'scrooloose/syntastic'
        Bundle 'tpope/vim-fugitive'
        Bundle 'mattn/webapi-vim'
        Bundle 'mattn/gist-vim'
        Bundle 'scrooloose/nerdcommenter'
        Bundle 'tpope/vim-commentary'
        Bundle 'godlygeek/tabular'
        Bundle 'luochen1990/rainbow'
        if executable('ctags')
            Bundle 'majutsushi/tagbar'
        endif
    endif
" }

" Snippets & AutoComplete {
    if count(g:spf13_bundle_groups, 'snipmate')
        Bundle 'garbas/vim-snipmate'
        Bundle 'honza/vim-snippets'
        " Source support_function.vim to support vim-snippets.
        if filereadable(expand("~/vim/bundle/vim-
snippets/snippets/support_functions.vim"))
            source ~/.vim/bundle/vim-snippets/snippets/support_functions.vim
        endif
    elseif count(g:spf13_bundle_groups, 'youcompleteme')
        Bundle 'Valloric/YouCompleteMe'
        Bundle 'SirVer/ultisnips'
        Bundle 'honza/vim-snippets'
    elseif count(g:spf13_bundle_groups, 'neocomplcache')
        Bundle 'Shougo/neocomplcache'
        Bundle 'Shougo/neosnippet'
        Bundle 'Shougo/neosnippet-snippets'
    endif
" }

```

```

        Bundle 'honza/vim-snippets'
elseif count(g:spf13_bundle_groups, 'neocomplete')
    Bundle 'Shougo/neocomplete.vim.git'
    Bundle 'Shougo/neosnippet'
    Bundle 'Shougo/neosnippet-snippets'
    Bundle 'honza/vim-snippets'
endif
" }

" PHP {
    if count(g:spf13_bundle_groups, 'php')
        Bundle 'spf13/PIV'
        Bundle 'arnaud-lb/vim-php-namespace'
        Bundle 'beyondwords/vim-twig'
    endif
" }

" Python {
    if count(g:spf13_bundle_groups, 'python')
        " Pick either python-mode or pyflakes & pydoc
        Bundle 'klen/python-mode'
        Bundle 'yssource/python.vim'
        Bundle 'python_match.vim'
        Bundle 'pythoncomplete'
    endif
" }

" Javascript {
    if count(g:spf13_bundle_groups, 'javascript')
        Bundle 'elzr/vim-json'
        Bundle 'groenewege/vim-less'
        Bundle 'pangloss/vim-javascript'
        Bundle 'briancollins/vim-jst'
        Bundle 'kchmck/vim-coffee-script'
    endif
" }

" Scala {
    if count(g:spf13_bundle_groups, 'scala')
        Bundle 'derekwyatt/vim-scala'
        Bundle 'derekwyatt/vim-sbt'
        Bundle 'xptemplate'
    endif
" }

" Haskell {
    if count(g:spf13_bundle_groups, 'haskell')
        Bundle 'travitch/hasksyn'
        Bundle 'dag/vim2hs'
        Bundle 'Twinside/vim-haskellConceal'
        Bundle 'Twinside/vim-haskellFold'
        Bundle 'lukerandall/haskellmode-vim'
        Bundle 'eagletmt/neco-ghc'
        Bundle 'eagletmt/ghcmod-vim'
        Bundle 'Shougo/vimproc.vim'
        Bundle 'adinapoli/cumino'
        Bundle 'bitc/vim-hdevtools'
    endif
" }

" HTML {
    if count(g:spf13_bundle_groups, 'html')
        Bundle 'amirh/HTML-AutoCloseTag'
        Bundle 'hail2u/vim-css3-syntax'
    endif
" }

```

```

        Bundle 'gorodinskiy/vim-coloresque'
        Bundle 'tpope/vim-haml'
        Bundle 'mattn/emmet-vim'
    endif
" }

" Ruby {
    if count(g:spf13_bundle_groups, 'ruby')
        Bundle 'tpope/vim-rails'
        let g:rubycomplete_buffer_loading = 1
        "let g:rubycomplete_classes_in_global = 1
        "let g:rubycomplete_rails = 1
    endif
" }

" Puppet {
    if count(g:spf13_bundle_groups, 'puppet')
        Bundle 'rodjek/vim-puppet'
    endif
" }

" Go Lang {
    if count(g:spf13_bundle_groups, 'go')
        "Bundle 'Blackrush/vim-gocode'
        Bundle 'fatih/vim-go'
    endif
" }

" Elixir {
    if count(g:spf13_bundle_groups, 'elixir')
        Bundle 'elixir-lang/vim-elixir'
        Bundle 'carlosgaldino/elixir-snippets'
        Bundle 'mattreduce/vim-mix'
    endif
" }

" Misc {
    if count(g:spf13_bundle_groups, 'misc')
        Bundle 'rust-lang/rust.vim'
        Bundle 'tpope/vim-markdown'
        Bundle 'spf13/vim-preview'
        Bundle 'tpope/vim-cucumber'
        Bundle 'cespare/vim-toml'
        Bundle 'quentindecock/vim-cucumber-align-pipes'
        Bundle 'saltstack/salt-vim'
        Bundle 'scriptnames'
    endif
" }

endif

" }

" Use fork bundles config if available {
    if filereadable(expand("~/vimrc.bundles.fork"))
        source ~/.vimrc.bundles.fork
    endif
" }

" Use local bundles config if available {
    if filereadable(expand("~/vimrc.bundles.local"))
        source ~/.vimrc.bundles.local
    endif
" }

```