# main.cpp 主函数

* bool CheckArgs(const QStringList &args)  
  检测传入字符串是否有-h或者-v指令
* void ShowUsage()  
  显示帮助信息
* void ShowVersion()  
  显示版本信息

# mainwindow.cpp 主窗体

* MainWindow::MainWindow(TranslationHandler\* th, QSettings\* settings)  
  按传入的设置布置主窗体
* void MainWindow::HandleCLIParams(const QStringList &params)  
  对传入的命令行程序进行处理
* void MainWindow::LoadSettings()  
  加载设置
* void MainWindow::SaveSettings()  
  保存设置
* void MainWindow::DoCheckFiles(const QStringList &files)  
  检测文件是否存在
* void MainWindow::CheckCode(const QString& code, const QString& filename)  
  利用cppcheck检测代码错误以及返回信息
* QStringList MainWindow::SelectFilesToCheck(QFileDialog::FileMode mode)  
  查询所以存在的文件并返回代表文件路径的字符串数字并显示错误信息
* void MainWindow::CheckFiles()  
  对SelectFilesToCheck检测出的文件进行多线程检测
* void MainWindow::CheckDirectory()  
  扫描.cppcheck文件中的目录是否存在
* void MainWindow::AddIncludeDirs(const QStringList &includeDirs, Settings &result)  
  为result增加绝对路径
* Library::Error MainWindow::LoadLibrary(Library \*library, QString filename)  
  加载库文件
* bool MainWindow::TryLoadLibrary(Library \*library, QString filename)  
  尝试加载库文件，成功返回true，失败显示错误信息后返回false
* Settings MainWindow::GetCppcheckSettings()  
  获取cppcheck的设置信息
* void MainWindow::CheckDone()  
  检测结束后对主界面做的操作
* void MainWindow::CheckLockDownUI()  
  锁定check的UI
* void MainWindow::ProgramSettings()  
  程序设置(执行设置)
* void MainWindow::ReCheck()  
  再次检测
* void MainWindow::ClearResults()  
  清除检测结果
* void MainWindow::OpenResults()  
  打开结果(XML)
* void MainWindow::LoadResults(const QString selectedFile)  
  加载结果信息
* void MainWindow::LoadResults(const QString selectedFile, const QString sourceDirectory)  
  加载结果信息(包含源文件目录参数)
* void MainWindow::EnableCheckButtons(bool enable)  
  点亮测试按钮(使之可以点击)
* void MainWindow::ShowStyle(bool checked)  
  结果视图增加风格警告
* void MainWindow::ShowErrors(bool checked)  
  结果视图增加错误信息
* void MainWindow::ShowWarnings(bool checked)  
  结果视图增加警告信息
* void MainWindow::ShowPortability(bool checked)  
  结果视图增加可移植性警告
* void MainWindow::ShowPerformance(bool checked)  
  结果视图增加性能警告
* void MainWindow::ShowInformation(bool checked)  
  结束视图显示信息
* void MainWindow::CheckAll()  
  选择全部
* void MainWindow::UncheckAll()  
  取消全部
* void MainWindow::closeEvent(QCloseEvent \*event)  
  关闭事件
* void MainWindow::ToggleAllChecked(bool checked)  
  切换到所有选择项
* void MainWindow::About()  
  程序相关信息
* void MainWindow::ShowLicense()  
  显示证书
* void MainWindow::ShowAuthors()  
  显示作者
* void MainWindow::Save()  
  保存
* void MainWindow::ResultsAdded()  
  结果集增加
* void MainWindow::ToggleMainToolBar()  
  切换主工具栏
* void MainWindow::ToggleViewToolBar()  
  切换视图工具栏
* void MainWindow::ToggleFilterToolBar()  
  切换过滤器工具栏
* void MainWindow::FormatAndSetTitle(const QString &text)  
  规范化并设置标题
* void MainWindow::SetLanguage(const QString &code)  
  设置语言
* void MainWindow::AboutToShowViewMenu()  
  显示视图菜单
* void MainWindow::StopChecking()  
  停止检测
* void MainWindow::OpenHelpContents()  
  打开在线帮助窗口
* void MainWindow::OpenOnlineHelp()  
  打开在线帮助网页
* void MainWindow::OpenProjectFile()  
  打开项目文件
* void MainWindow::ShowScratchpad()  
  显示暂时存储器
* void MainWindow::LoadProjectFile(const QString &filePath)  
  加载项目文件
* void MainWindow::CheckProject(Project \*project)  
  对整个项目进行检查
* void MainWindow::NewProjectFile()  
  新建项目文件
* void MainWindow::CloseProjectFile()  
  关闭项目文件
* void MainWindow::EditProjectFile()  
  编辑项目文件
* void MainWindow::ShowLogView()  
  显示日志视图
* void MainWindow::ShowStatistics()  
  显示统计信息
* void MainWindow::ShowLibraryEditor()  
  显示库文件(设置)编辑器
* void MainWindow::Log(const QString &logline)  
  显示日志栏目
* void MainWindow::DebugError(const ErrorItem &item)  
  显示错误信息栏目
* void MainWindow::FilterResults()  
  显示过滤结果
* void MainWindow::EnableProjectActions(bool enable)  
  允许进行工程操作
* void MainWindow::EnableProjectOpenActions(bool enable)  
  允许进行打开工程操作
* void MainWindow::OpenRecentProject()  
  打开最近的工程
* void MainWindow::UpdateMRUMenuItems()  
  更新最大接收单元(MRU)设置
* void MainWindow::AddProjectMRU(const QString &project)  
  增加最大接收单元(MRU)设置
* void MainWindow::RemoveProjectMRU(const QString &project)  
  删除最大接收单元(MRU)设置
* void MainWindow::SelectPlatform()  
  选择使用平台(Unix,windows)

# aboutdialog.cpp

获取对话框相关信息

# application.cpp

调用软件

# applicationdialog.cpp

程序对话框(打开文件窗口，信息回显策划公开)

# applicationlist.cpp

根据设置加载程序列表

# checkstatistics.cpp

检测得到的各项统计量

# checkthread.cpp

检测的线程（多线程进行检测）

# common.cpp

设置和获取指定字符串路径

# cppchecklibrarydata.cpp

cppcheck设置数据

# csvreport.cpp

CSV型文件报告

# erroritem.cpp

错误条目信息

# filelist.cpp

得到指定文件格式的文件列表

# fileviewdialog.cpp

文件显示窗体

# libraryaddfunctiondialog.cpp

增加设置的功能

# librarydialog.cpp

设置窗口

# libraryeditargdialog.cpp

编辑设置窗口

# logview.cpp

日志查看

# platforms.cpp

本机使用平台查看及设置

# printablereport.cpp

打印报告信息

# project.cpp

工程状态以及操作

# projectfile.cpp

工程文件信息

# projectfiledialog.cpp

工程文件窗口(include,exclude，library)

# report.cpp

操作报告（创建，打开，关闭等操作的返回信息）

# resultstree.cpp

检测后的结果树

# resultsview.cpp

结果显示试图(显示结果树)

# scratchpad.cpp

暂时存储器

# settingsdialog.cpp

设置窗口

# showtypes.cpp

显示类型(严重级显示,加载，保存操作)

# statsdialog.cpp

统计窗口(计算扫描时间，保存信息到剪切板等功能)

# threadhandler.cpp

线程处理程序

# threadresult.cpp

线程的结果报告

# translationhandler.cpp

翻译处理

# txtreport.cpp

保存错误报告到txt文件

# xmlreport.cpp

结果保存到xml文件

# xmlreportv1.cpp

结果保存到xml版本1

# xmlreportv2.cpp

结果保存到xml版本2