Code Atlas 使用手册

欧阳耀斌

目录

介绍

很多程序员，尤其是初学编程的程序员，往往会觉得看代码是一件困难的事。原因是看代码时往往需要搞清函数的调用关系。有时一个功能往往有几十个函数，记忆这几十个函数的调用关系本身就是一件吃力的事。我自己经常需要用笔在纸上画出调用图，于是我就想，能不能把这个工作用程序自动完成呢？于是就有了做代码可视化工具的念头。

最初的想法是用地图的形式把代码的关系表现出来。为此当时还看了一些论文，做了一些 demo，也认识了 Understand 这个代码分析工具，接触了一些数据挖掘和可视化知识。但是一开始目标定得太大，要求太高，导致进展缓慢，做出来的 demo 离实用很远。

进了一个游戏公司工作以后时间大大减少，于是不得不调低目标，先解决工作中面对的看代码问题，再逐步完善系统。公司游戏逻辑用的是 Python 语言，有时任务紧，需要快速看懂一份代码然后在其基础上实现新功能，例如新的玩法。而同事们特别喜欢用 Sublime Text 作为代码编辑工具，因为它能够很方便地在代码中跳转，编辑功能也很强大。因此我的目标便定在编写一个快速浏览 Python 调用图的 Sublime Text 插件。后来去了另一间公司，用的语言变成 C++，于是又对插件做了一些修改，逐步变成现在的样子。

这个插件能做什么呢？主要是帮助程序员浏览代码。在刚刚接触一份代码时，由于对其内部逻辑不熟悉，程序员往往需要花费很多时间看代码，以便搞清楚函数的调用关系，找到重要的调用流程，重要功能的实现代码。这个工具可以快速在调用图中切换，从而加速这个过程。而且程序员可以保存过往浏览过的调用关系，在重要函数上写说明，从而避免看过又忘的情形，减少重复劳动。

目前这个插件支持 C/C++ 和 Python 两种语言。对于 Java，C#等语言，原则上也是支持的，但是我没有测试过。

我在自己的博客上写了这个插件简化版教程，可以参考这个链接：

http://www.cnblogs.com/dydx/p/5393802.html

这是优酷上的视频教程：

http://list.youku.com/albumlist/show?id=28202266&ascending=1&page=1

非常欢迎大家给我提建议，我的邮箱是 549088764@qq.com，github 主页是 https://github.com/league1991。

2016 年 9 月 16 日

安装方法

安装之前，需要准备以下“原材料”：

名称	版本	作用	网址
Python	3.4	插件用 python 编写	https://www.python.org/ftp/python/3.4.0/python-3.4.0.msi
PyQt	对应 Qt 版本 4.8 对应 Python 版本 3.4	插件使用的界面库	https://sourceforge.net/projects/pyqt/files/PyQt4/PyQt-4.11.4/PyQt4-4.11.4-gpl-Py3.4-Qt4.8.7-x32.exe/download
Sublime Text	3.x	代码编辑器	https://download.sublimetext.com/Sublime%20Text%20Build%203114%20Setup.exe
Understand	3.x	代码分析工具	https://scitools.com/ （官方网站，需要先注册才能下载）
CodeAtlas	无	本插件	https://github.com/league1991/CodeAtlasSublime （直接点 Clone or download 下载 zip 压缩包即可）

注意上述所有程序均为 32 位。以上软件中，必须保证 PyQt 对应的 Python 版本为 3.4，这是因为 Understand 软件提供的 Python API 版本为 3.4。

下面介绍 Code Atlas 的安装步骤。

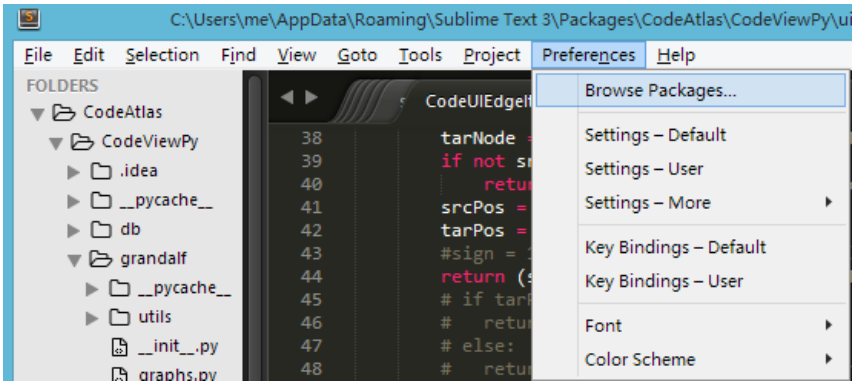
（1）安装 Understand。直接双击安装包安装即可。

（2）安装 Sublime Text。直接双击安装包安装即可。

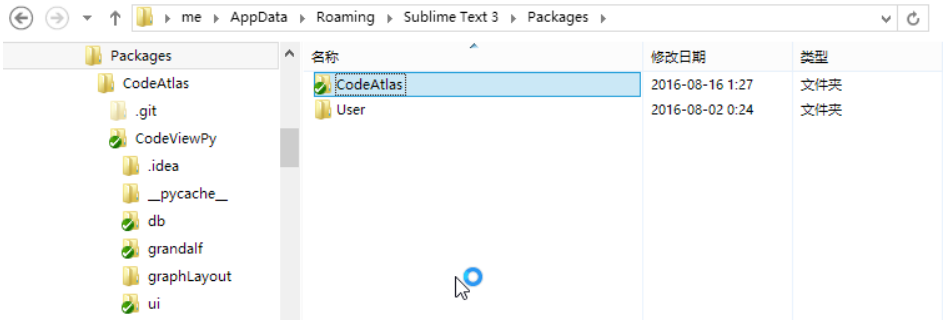
（3）安装 Python。直接双击安装包安装即可。

（4）安装 PyQt。直接双击安装包安装即可。

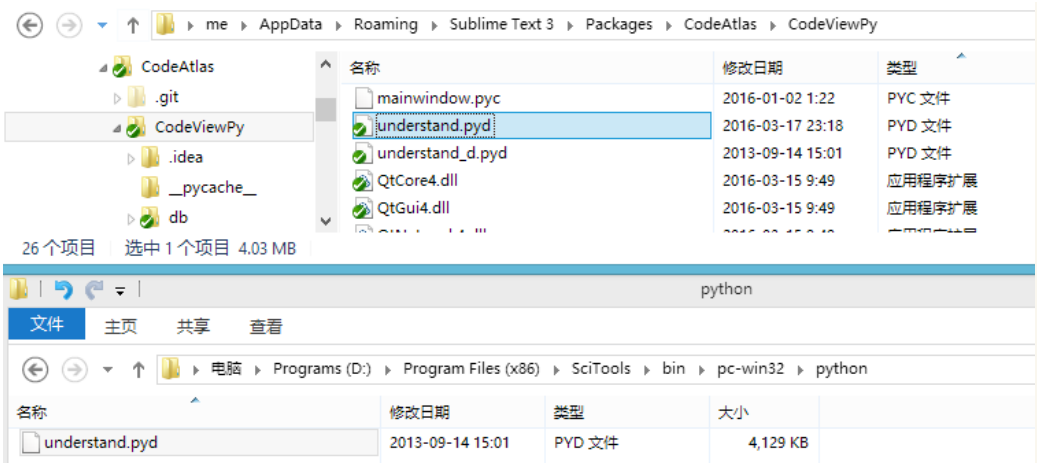
（5）打开 Sublime Text，点 Preferences -> Browse Packages，打开 Packages 文件夹。



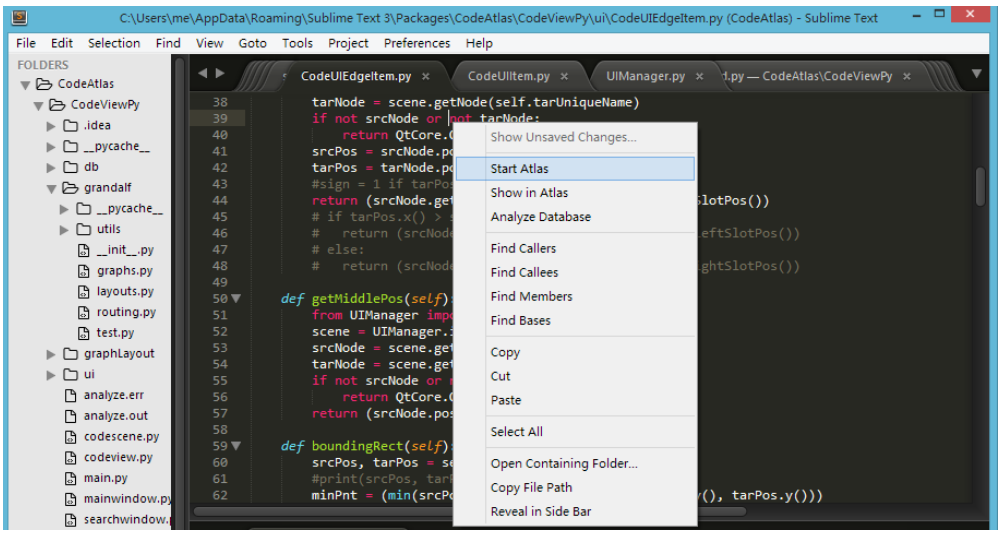
（6）将下载的 Code Atlas 压缩包内容解压到该文件夹，如图。



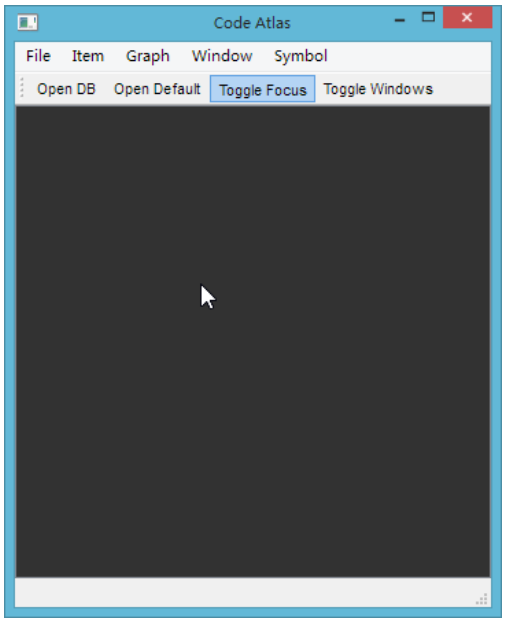
（7）把文件 Packages/CodeAtlas/CodeViewPy/understand.pyd 用 Understand 安装目录下面的同名文件（通常在 SciTools\bin\pc-win32\python 目录下）替换。此 pyd 为 Understand 的 API 库，替换是为了保证本插件能够使用新版本的 Understand。两个 pyd 的位置如图。



（8）至此插件已经安装完毕。为了检查是否安装成功，可以打开 Sublime Text，在代码编辑区域按鼠标右键，选择“Start Atlas”。



如果一切正常，会弹出下面的窗口。



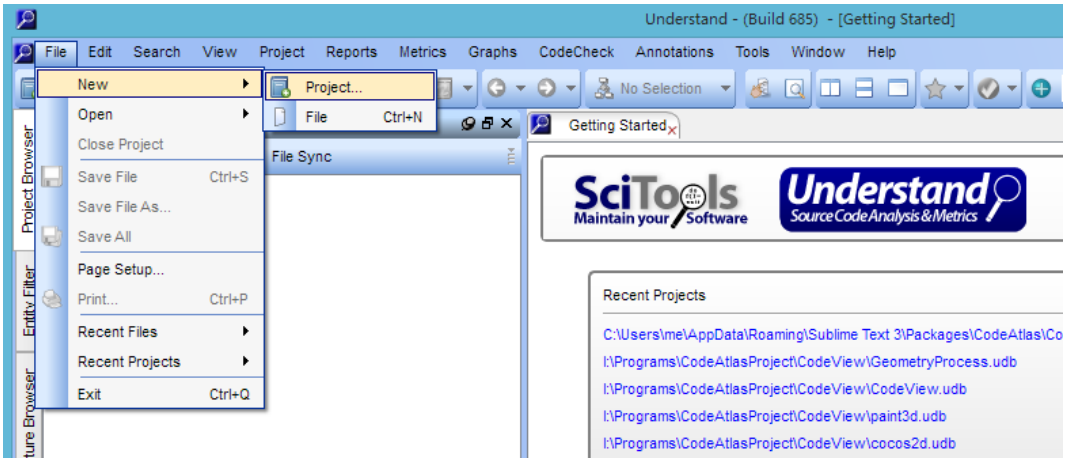
下面我们可以开始使用这个插件啦。

快速入门

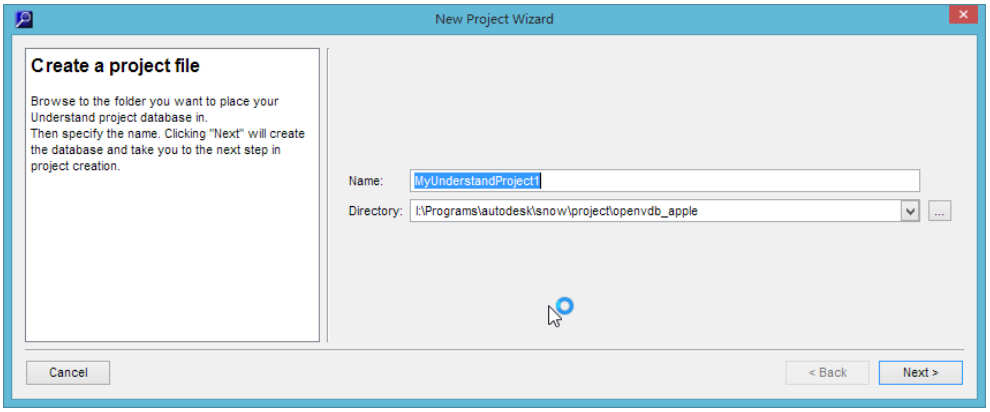
本插件通过查询 Understand 数据库，将查询结果用节点连接图的形式显示出来。因此，首先用户需要建立 Understand 数据库，然后在 Sublime Text 打开插件，用插件打开数据库。

现假设用户有一份 C++代码，代码可以是 Visual Studio 的解决方案，也可以是单纯的 C/C++代码文件。使用本插件的具体步骤如下：

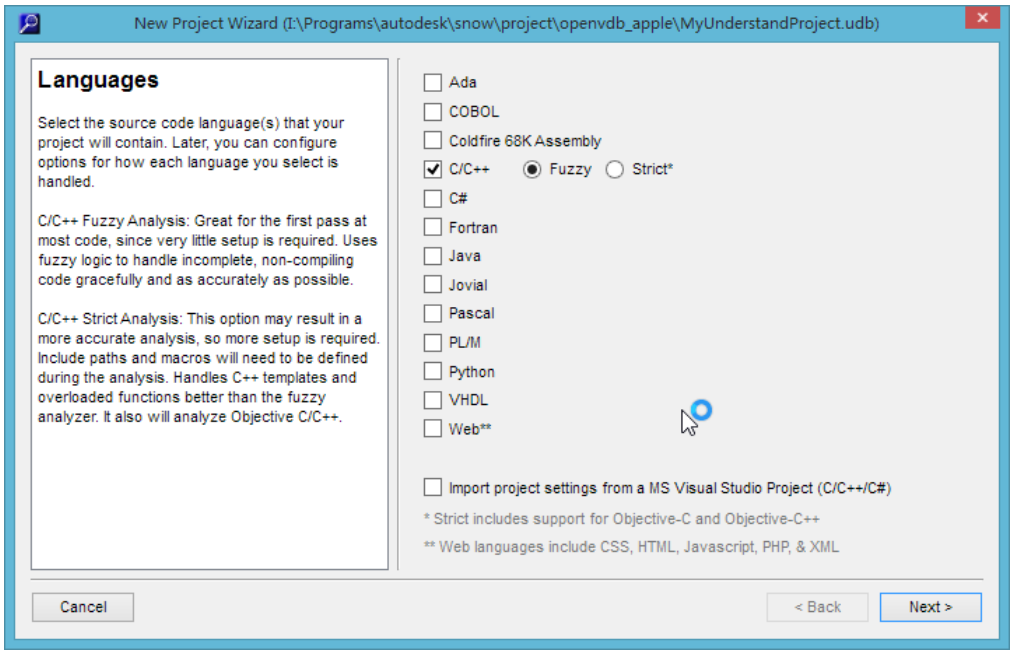
（1）打开 Understand 软件，选择 File -> New -> Project。



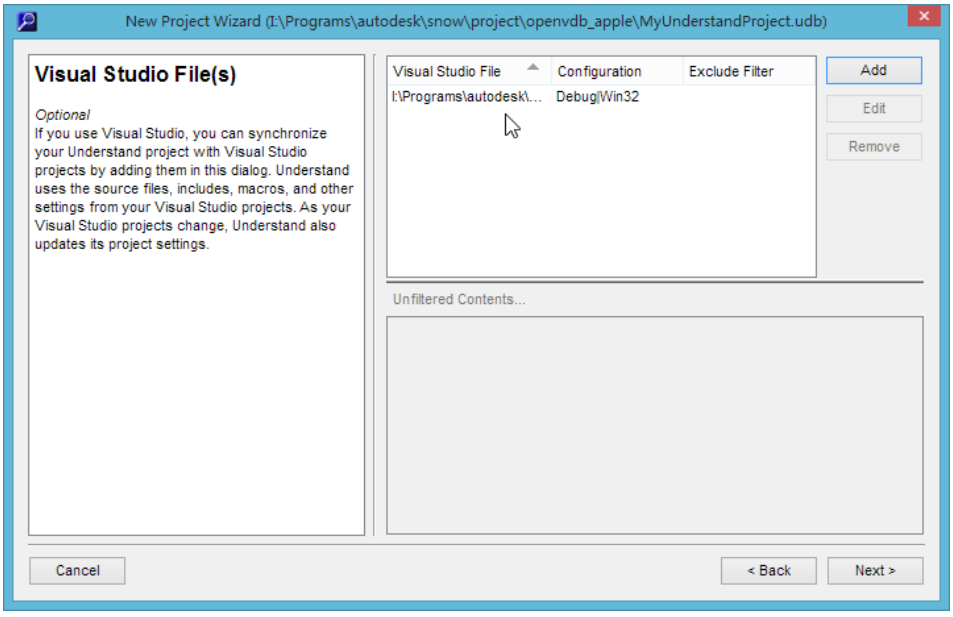
（2）弹出一个窗口。Name 可以随便起，Directory 请指向你要分析的代码所在文件夹。



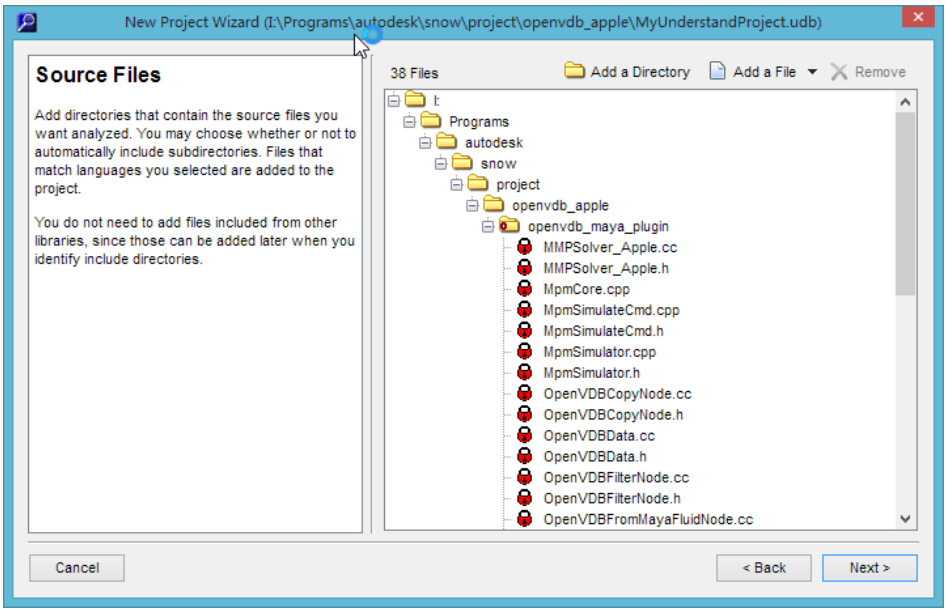
（3）指定语言，请选择 C/C++。注意窗口下面有个选择，问你是不是要导入 Visual Studio 工程。如果有的话，最好选导入。这样分析结果会更加精准。



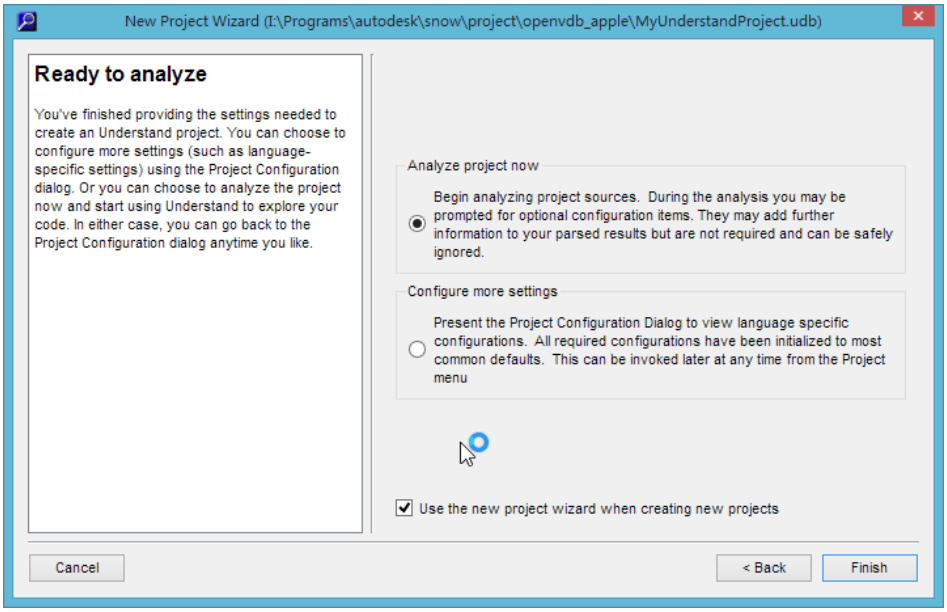
（4）如果上一步指定了导入 Visual Studio 工程，会进一步要求你指定 VS 解决方案文件。



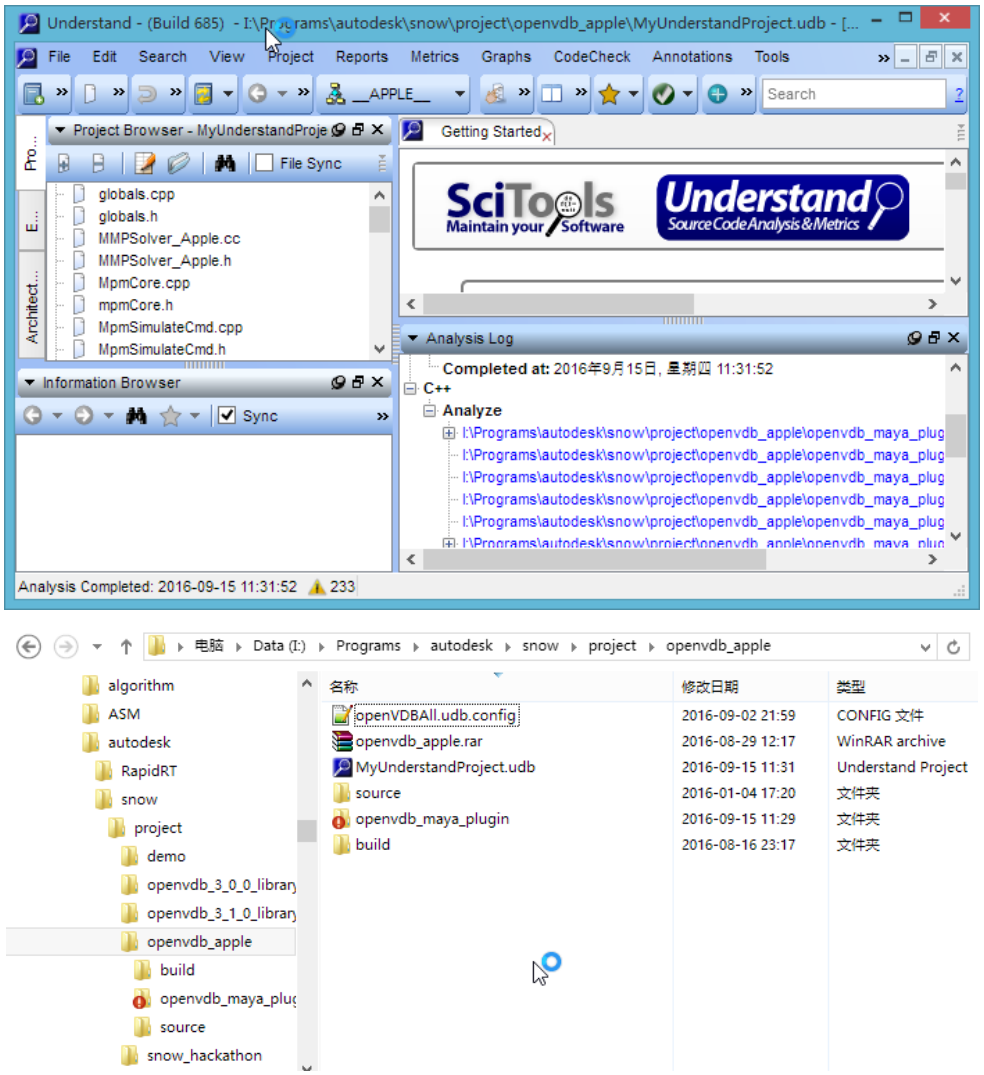
（5）指定要分析的代码文件夹。



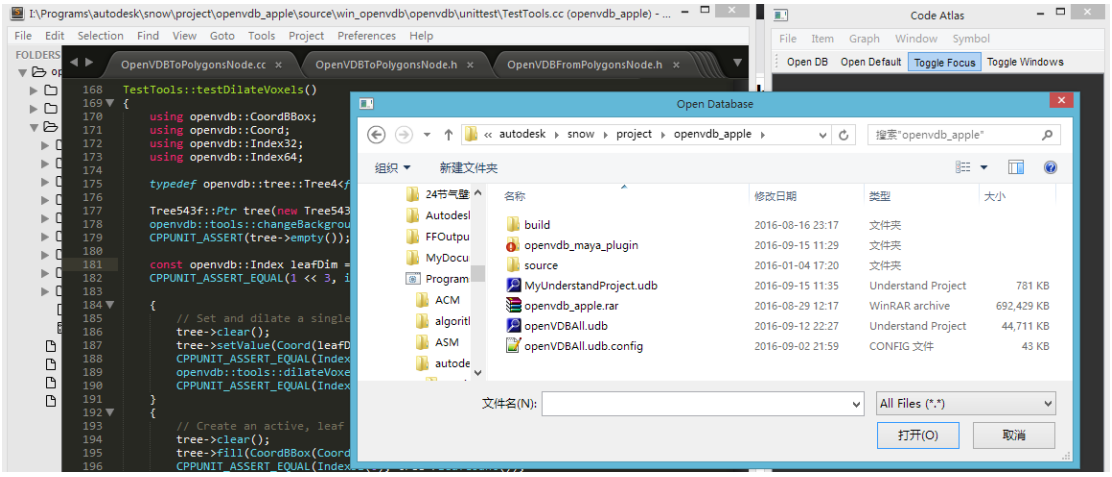
（6）确认并开始分析。



（7）分析结束后，会提示 “Analysis Complete”。此时可以直接关闭 Understand 窗口。在最初指定的文件夹位置会发现一个.udb 文件，这就是要生成的代码数据库。

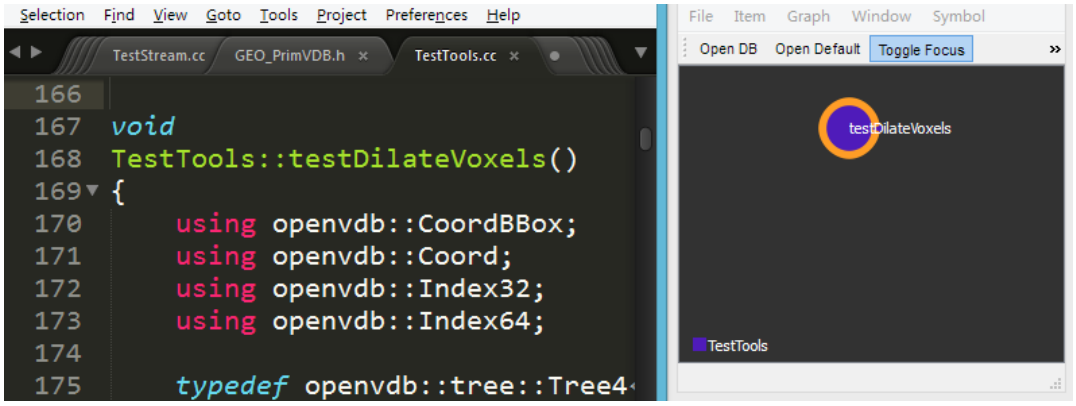


（8）至此数据库生成完毕，打开 Sublime Text，在代码编辑区按右键点“Start Atlas”，弹出插件界面，再点工具栏中的“Open DB”，选择之前生成的 udb 数据库文件即可。



（9）至此数据库已经被打开。为了查看一个函数的调用关系，我们在 Sublime Text 上令光标指向一个我们感兴趣的函数名，按下 Alt+G，在插件界面上即会弹出一个圆点，代表

我们的函数。圆点周围有橙色边，表示当前函数被选中。左下角有个与函数一样是紫色的正方形，正方形右边写着“TestTools”，这是表示函数属于一个叫做“TestTools”的类。

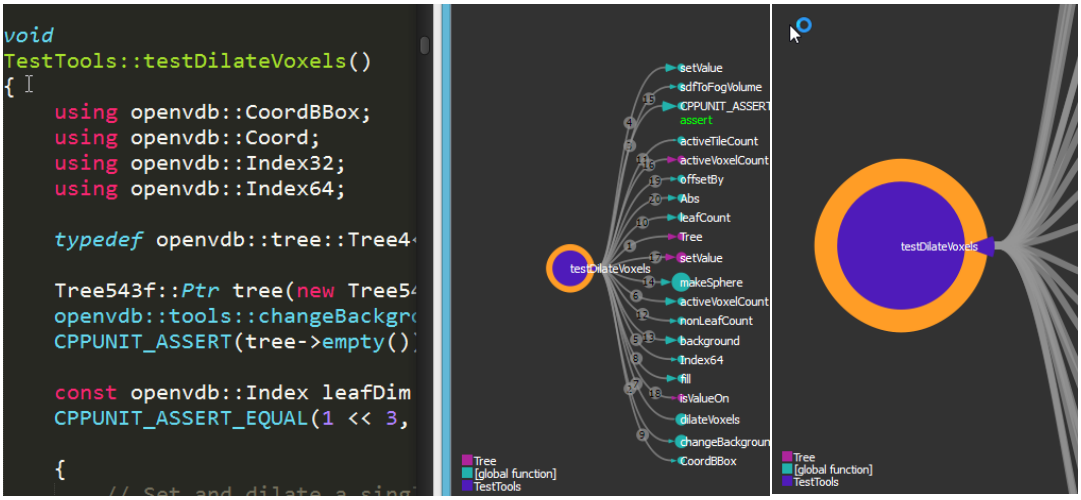


（10）现在我们希望查看这个函数调用了哪些函数，只需要在 Sublime Text 界面按下 Alt+V，右边的插件界面即会显示出其调用的函数。

可以发现代表函数的圆形有大有小，这是代表函数自身代码行数的多少。例如：函数 makeSphere()代码行数多于函数 Abs()。代码多的函数往往包含较重要的流程、算法，值得进一步检查。

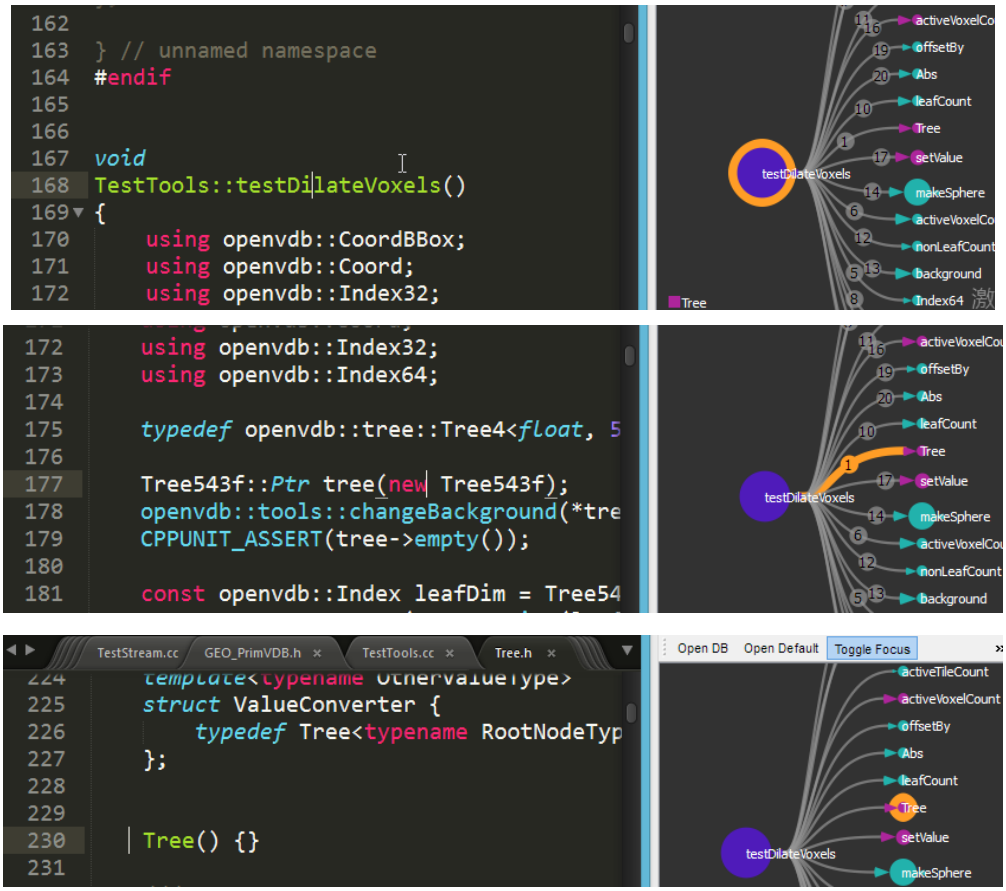
代表函数的圆形左侧会有一个三角形，三角形的大小代表调用此函数的调用者多少。通常一些工具类，例如标准库的 std::vector，数学库的 Matrix，会在整个程序中到处被使用，导致其成员函数有特别多的调用者。根据此信息，可以判断一个函数是否属于这种工具类。通常这类函数功能平平无奇，没有必要展开来看。

与左侧的情况类似，代表函数的圆形右侧也有一个三角形，如下面最右边的放大图。右侧的三角形大小表示一个函数调用其他函数的数目。通常来说，调用的数目越多，函数越可能包含复杂的处理步骤或算法。这种函数值得跟进去看看。

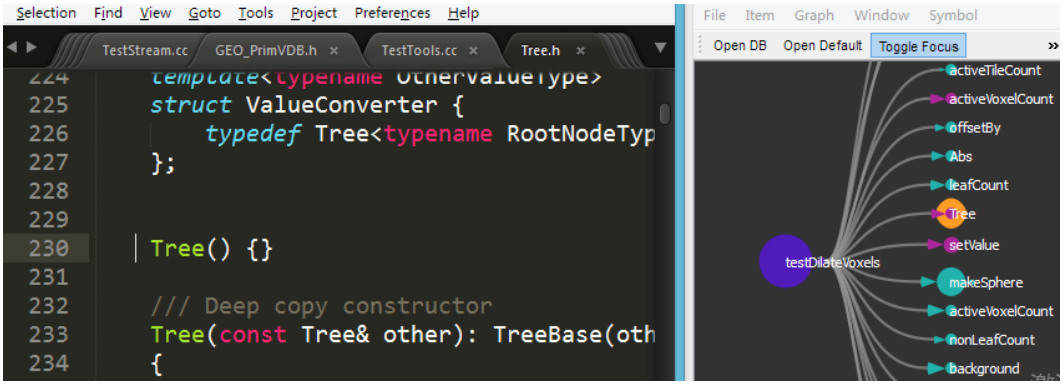


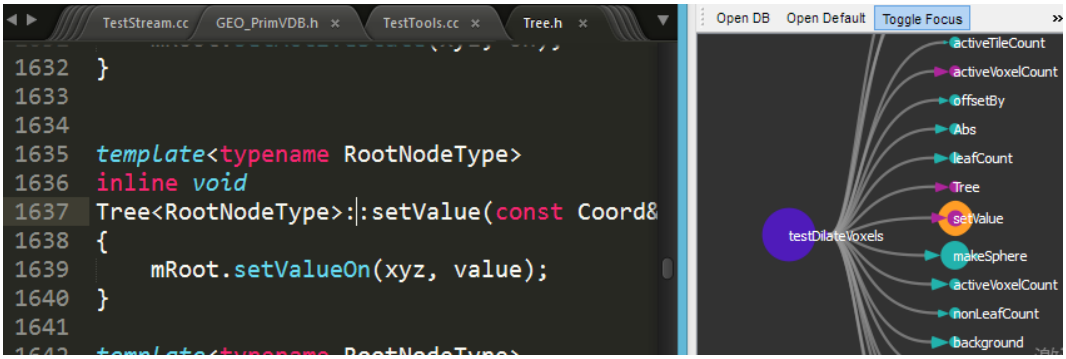
（11）查看了调用的函数之后，怎样跳转到这些函数当中呢？很简单，在 Sublime Text

编辑器中按下 Alt+右箭头，调用图会自动选中调用边，同时 Sublime Text 会跳转到原有函数调用第一个函数的代码，再按一下 Alt+右箭头后，调用图会选中调用的函数，同时 Sublime Text 也跳转到那个函数中了。整个流程如下面三幅图所示，注意光标的位置：

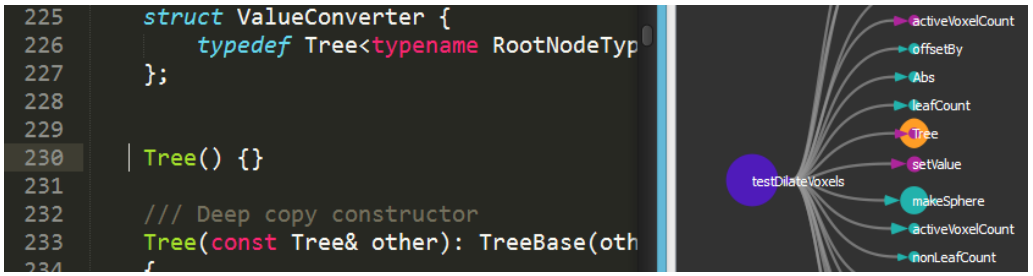


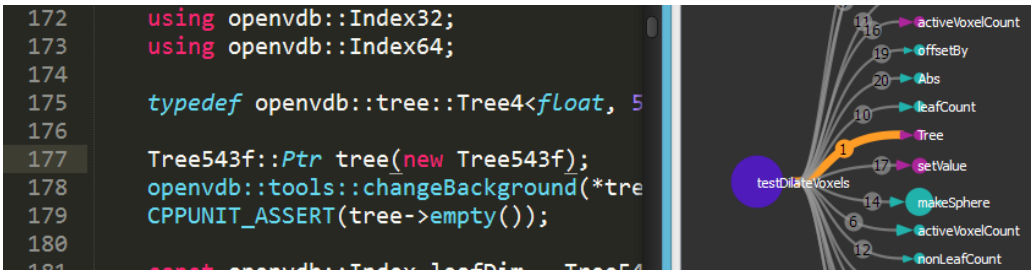
（12）当前跳转到的是第一个函数，如果我想看看其他函数呢？很简单，继续在 Sublime Text 编辑器中，按下 Alt +下箭头，即可跳转到插件视图中位于当前选中的函数下面的函数。如果按 Alt+上箭头，则是跳转到上面的函数。如果按 Alt+左箭头，则是跳回到调用者。下面的图展示了跳到其他函数的经过。

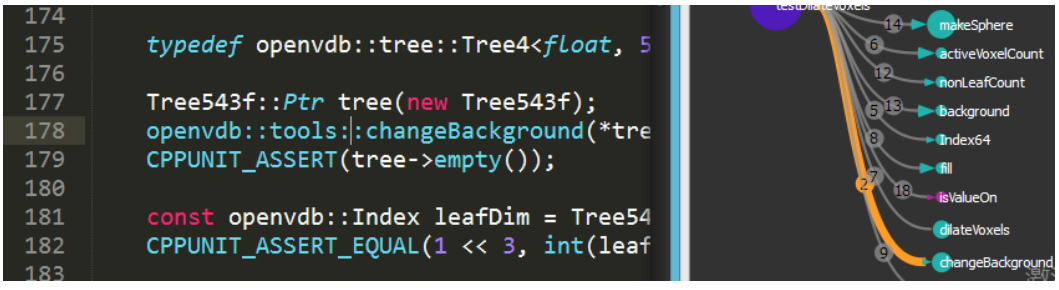


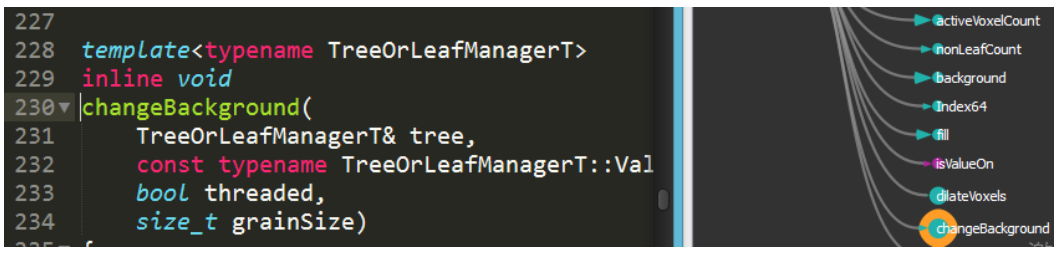


（13）但是如果想按调用的顺序浏览呢？也很简单，只需要先按 Alt+左箭头，退回到调用边，再按 Alt+上/下箭头，此时不会跳转到当前调用边相邻的调用边，而是按照调用的出现顺序，跳转到下一条调用边。如果想看看下一个调用的函数的详情，按 Alt+右箭头进去看即可。整个操作过程如下面四幅图，注意调用边上的数字表示调用顺序：





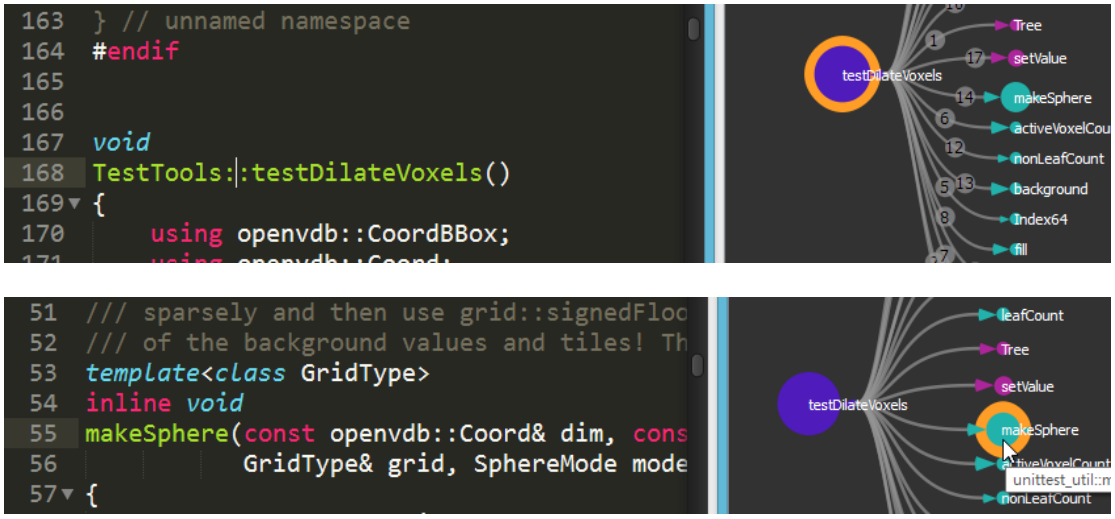




（14）以上的操作都是通过在 Sublime Text 上按 Alt+快捷键完成，这也是这个插件的推荐使用方法。为什么推荐使用快捷键，而不是用鼠标点点点呢？因为在实际浏览代码、编写

程序的过程中，如果操作需要交替使用鼠标和键盘，则操作时间会大大延长，降低使用效率。因此，一些 Linux 平台上的编辑器，以及近年一些较新的编辑器如 Sublime Text，其交互设计的思路，都是力求用户能用键盘完成大部分操作。

但如果你是一个新手，对快捷键尚未熟练，往往会更喜欢使用鼠标。我们的插件当然也会照顾你的需求。在界面上，用鼠标双击任何一个函数，Sublime Text 会自动跳转到对应的代码；双击任何一条调用边，Sublime Text 也会自动跳转到调用的语句。同时，只要鼠标一离开视图范围，新选中的函数或调用边会慢慢移动到视图中心。下面两幅图展示了这个过程：



（15）这么辛苦才看到的调用图，一关闭程序就没有了，是不是太不值得了呢？不会的，关闭插件时，插件会自动将当前你在看的调用图保存。保存的地方在 udb 文件的目录下，文件的名字与 udb 文件同名，但扩展名为.config。

以上就是这个插件的基本使用方法啦。除了上面介绍的方法，这个插件还能够查看类的继承关系，搜索函数到另一个函数是否存在调用链，保存看过的调用图，在函数上随时写注释等等功能。这些功能会在下面详细介绍。

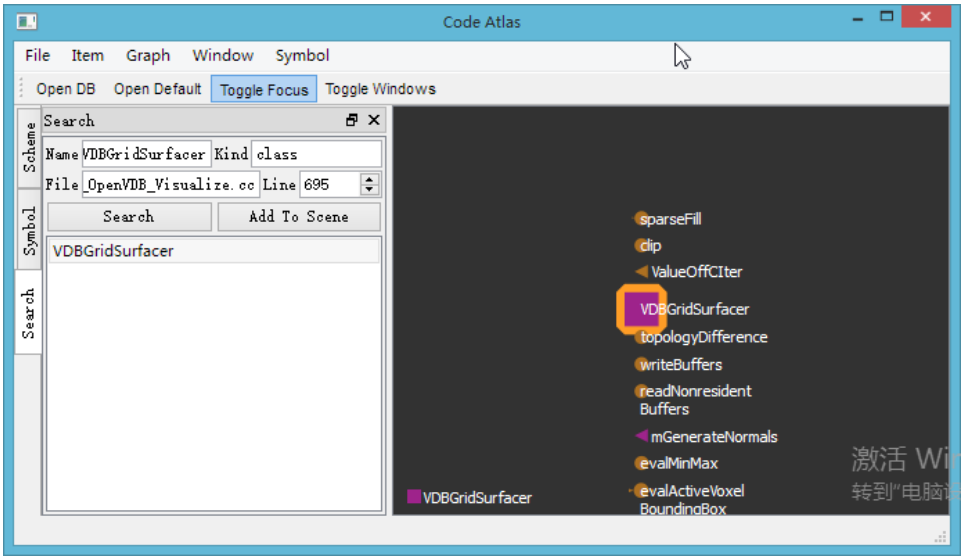
功能介绍

下面完整介绍这个插件的所有功能。各位读者最好先读完上面的快速教程，对插件的基本使用方法有一定了解之后，再来了解这里列出的其他用法。

在视图中显示符号

> *在 Sublime Text 中，令光标至于类名/函数名/变量名上，按下 Alt+G*

除了在 Sublime Text 按快捷键之外，还可以在插件界面上直接搜索。方法是点击界面上 Toggle Windows 按钮，在弹出的选项卡中选择“Search”，即可看到搜索界面。输入相应的名称、类型（包括 class/variable/function）、文件路径（可选）、行号（可选）后，点 Search，下方的列表即显示出符合要求的结果，选中结果后点 Add To Scene，即可添加到右方的视图中。在一个函数被大量重载、或是使用虚函数的情形，Understand 分析结果可能有误，导致直接按 Alt+G 可能并不能找到正确的函数。这个时候就需要利用搜索栏慢慢筛选，找到正确的函数。



查找函数调用

> *1. 在 Sublime Text 中（实际上在调用图中也可以）按下 Alt+C，查找当前选中函数的调用者*
>
> *2. 按下 Alt+V 查找当前函数的被调用者*

为了避免视图中的节点越积越多，拖慢显示速度，令用户迷失方向，视图中会自动删除最旧的节点。在有些情况下，由于函数的调用者或被调用者太多，这种机制会导致按下快捷键之后原有的函数反而不见了。因此，需要通过函数两侧的三角形大小判断调用者、被调用者的多少。如果调用者或被调用者太多，最好直接在代码中用 Alt+G 查看，再反过来用 Alt+V/C 补上调用边。

查找类的成员

> *1. 在 Sublime Text 中，令光标至于类名上，按 Alt+G，视图中显示类的节点*
>
> *2. 按 Alt+M，视图中显示类的成员*

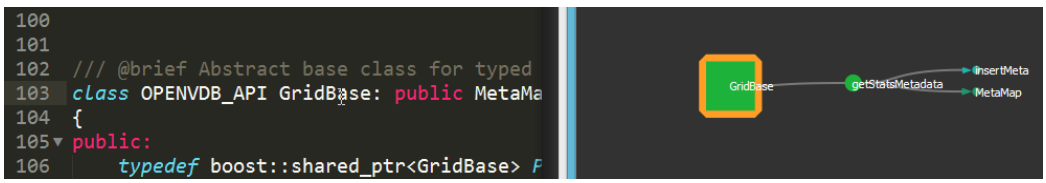
上述步骤如图所示：

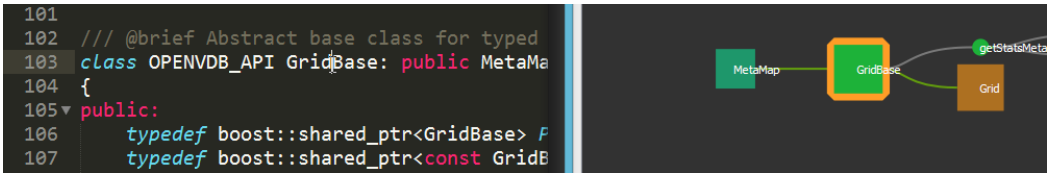
图中正方形的节点代表类，三角形的节点代表变量。类与函数的情形类似，正方形的大小表示类成员函数和变量的多少。变量的三角形的大小恒定。

查找类的继承关系

> *在视图中选中一个类，在 Sublime Text 按下 Alt+B，即可显示这个类的基类和派生类。*

下面显示了操作前后的变化。规定一个类的基类在这个类的左边，派生类在右边。

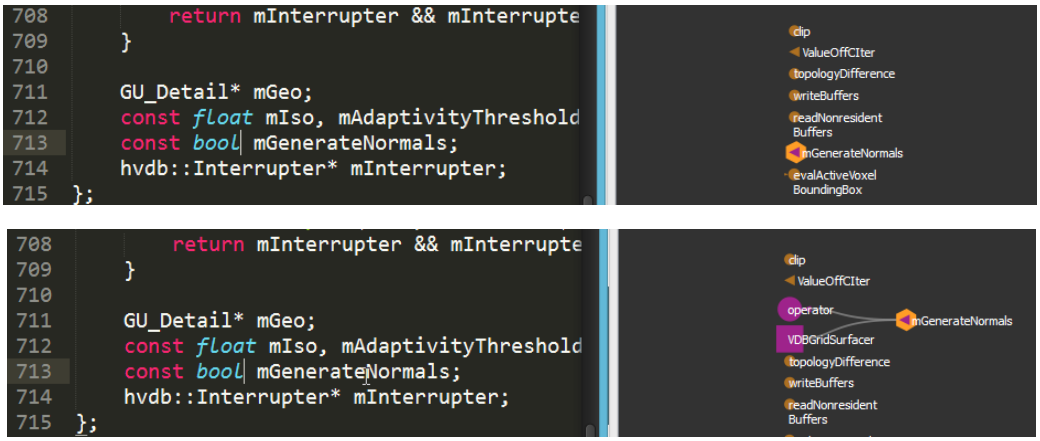




查找变量被引用的情况

> *在视图中选中一个变量，按 Alt+U，即可找到定义这个变量的类和使用这个变量的函数。*

下图显示了操作前后的变化。



跳到其他函数、变量和类

> *在 Sublime Text 中，使用 Alt+上下左右箭头跳转。*

具体的操作已经在快速教程详细讲过啦。

删除不需要的节点和边

有时候只对一部分类、函数、变量等感兴趣，这时可以把一些不需要的部分删除。

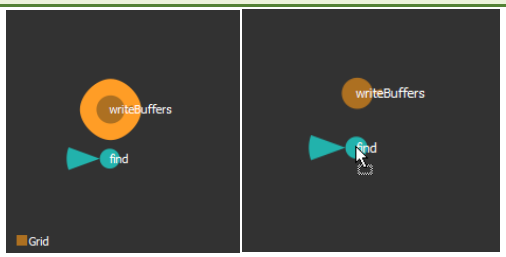
> *1. 在视图中选中要删除的节点或边*
>
> *2. 在 Sublime Text 中，按 Alt+Delete，即可删除之*

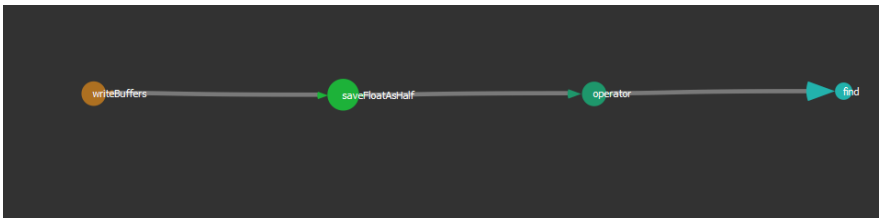
删除后会自动选中邻近的任一节点或边，因此此时不必用鼠标手动选择新的函数、变量等，而是可以用 Alt+上下左右箭头键跳到你想看的下一个函数上。

查找调用路径

有时候需要检查从一个函数到另一个函数是否存在函数调用链。

1. *在视图中显示函数 A 和 B。*
2. *若要检查是否存在从 A 到 B 的调用链，在视图中，用鼠标中键把 A 拖到 B。*



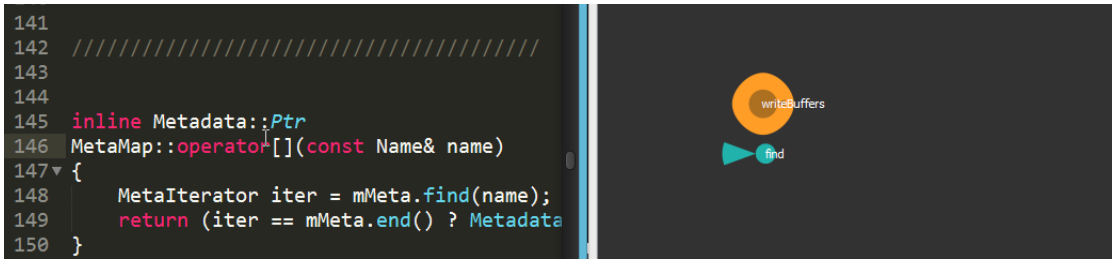


善用这个功能，可以快速找到一个功能的主要调用流程。方法是选定两个有代表性的函数，通常是有明确功能的函数，检查其中一个是否存在到另一个的调用链。若存在，再仔细检查调用链的每一个节点，找出更多重要的函数。

把特定函数加入黑名单

有些函数，例如 std::vector::size()这样的函数，在程序中大量使用，但是由于大家都知道它的意思，因此完全没有必要在调用图中显示出来。因此，可以将这些函数拉入黑名单，令其不显示。

1. *选中不想显示的函数。*
2. *按 Alt+I，将函数加入黑名单。*

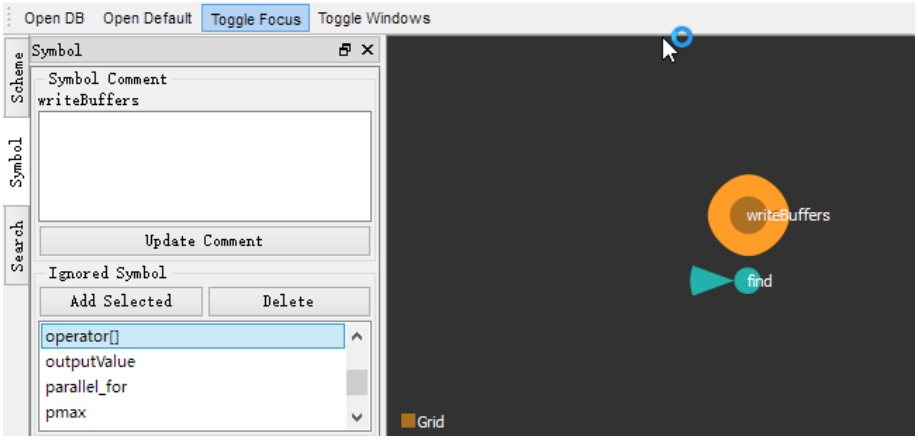


加入黑名单的函数会立即被删除，下次若对其调用者按 Alt+V，或对其被调用者按 Alt+C，这个函数都不会再显示出来。

怎样把一个函数拉出黑名单呢？步骤如下：

> *1. 点一下 Toggle Windows 按钮，在弹出的窗口中选择 Symbol 选项卡。其中的 Ignored Symbol 即为被加入黑名单的函数/类/变量。*
>
> *2. 选中要拉出黑名单的项，点 Delete，即可将其拉出黑名单。*

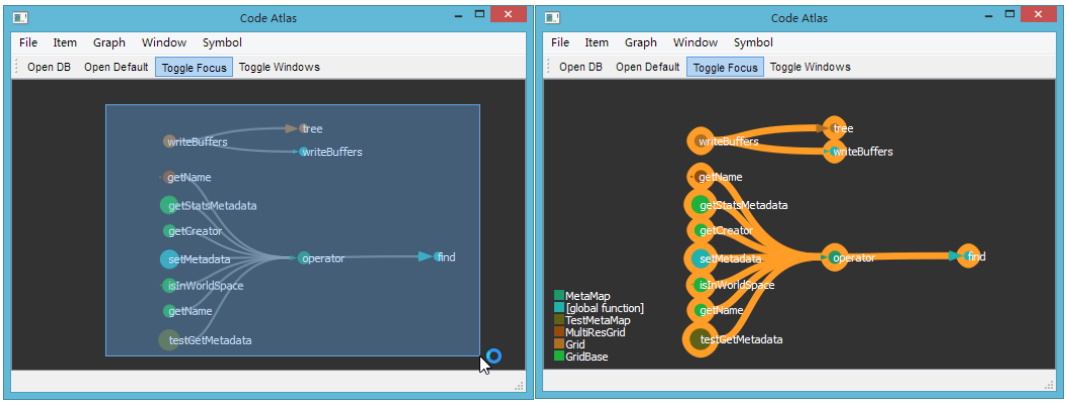
图中可以看到刚才被拉黑的 operator[]()函数。



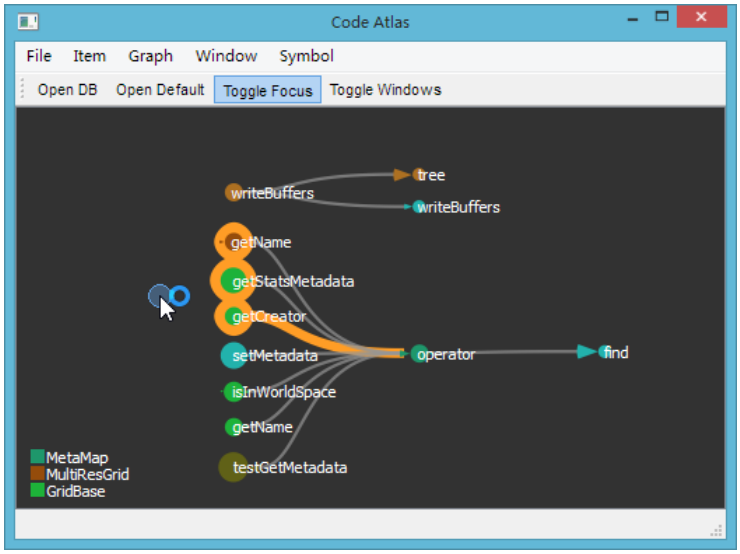
选择的技巧

选中节点和边有两种方法：一种是框选，另一种是用笔刷去刷。

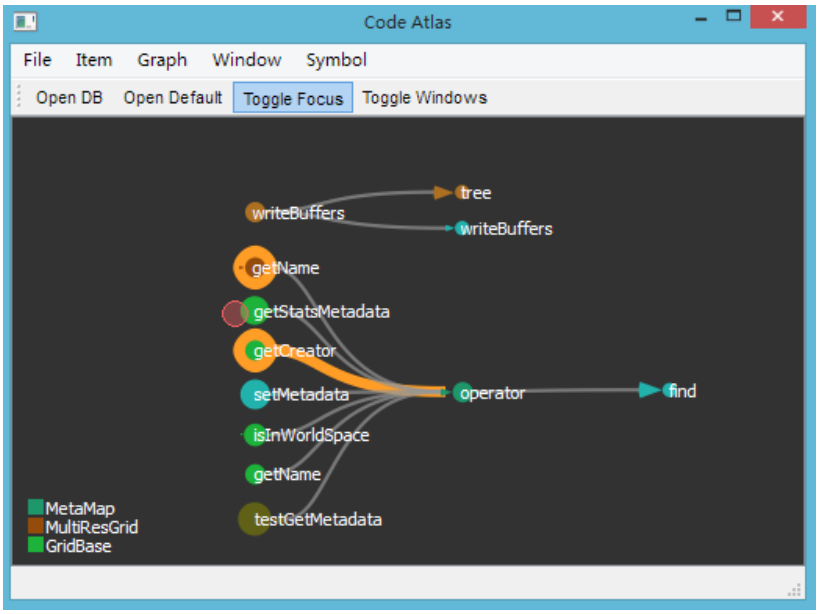
框选的方法没有什么好说的，如图。选中以后，点击视图空白处会清空选择。



如果你觉得框选常常选到不想选的东西，可以尝试笔刷选择模式。方法是按住 Ctrl 键，此时鼠标指针会变成一个蓝色的圆圈。按下左键，令鼠标扫过节点或者边，此节点或边就会被选中，如下图。



如果不小心选多了呢？可以按住 Shift 键，鼠标按住左键，扫过不想选中的节点或边，于是它就不被选中了，如下图。

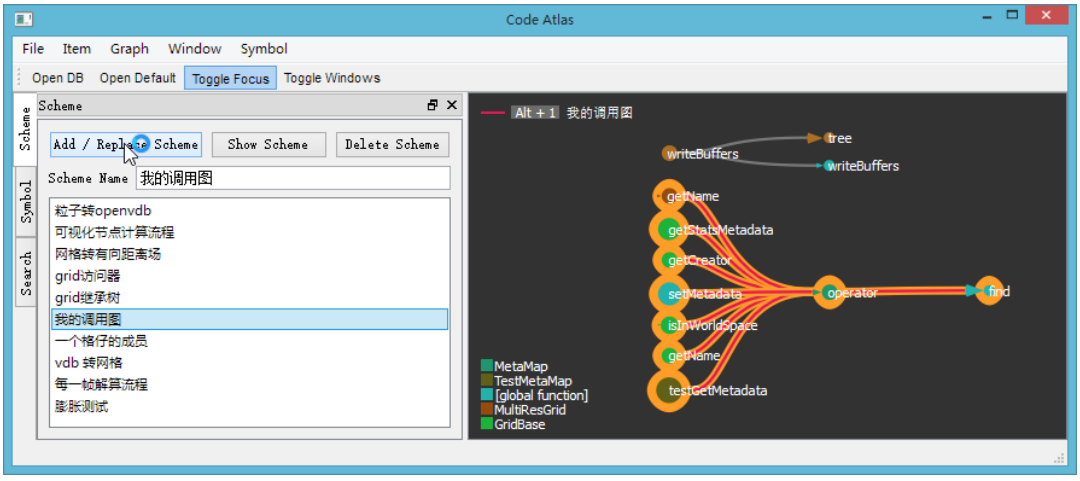


保存调用图

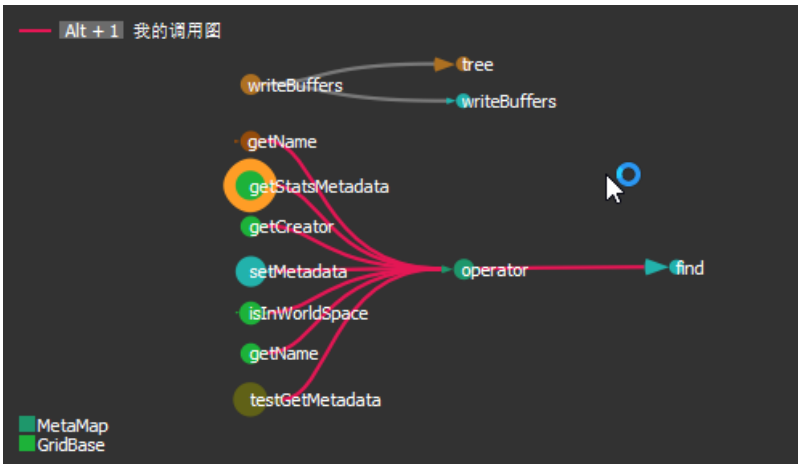
辛辛苦苦挖出来的调用图，当然舍不得就这样冲掉了。怎样保存呢？步骤如下：

1. *选中要保存的节点和边。*
2. *按下 Toggle Windows，调出 Scheme 选项卡。*
3. *在 Scheme Name 输入调用图名字，按下 Add/Replace Scheme。*

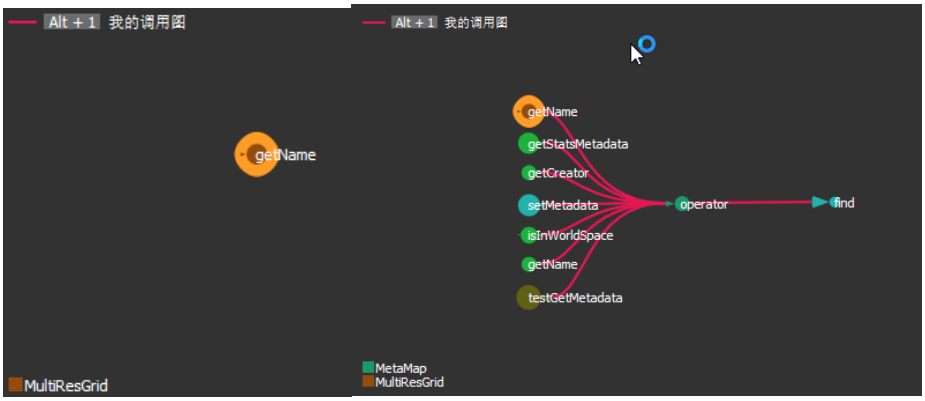
保存了之后，可以见到下方列表多了刚才输入的名字“我的调用图”，如下图。



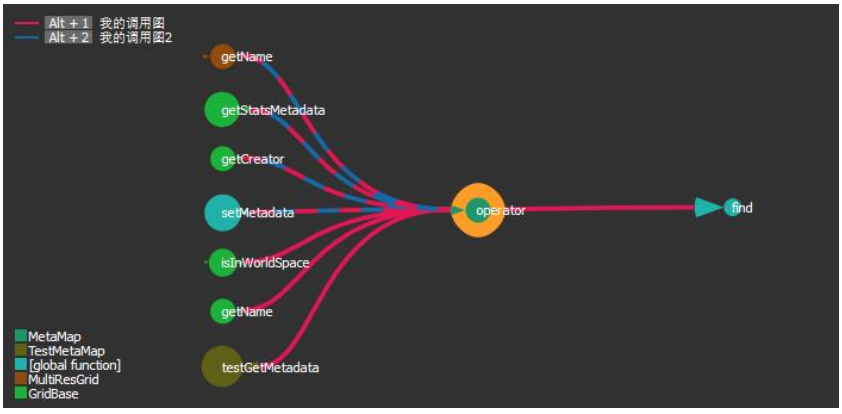
另一个变化是，当我们选中刚才“我的调用图”的其中一个节点或边时，属于该调用图的其余所有边的颜色会从灰色变成彩色，如图。



同时，视图左上角会出现一个图例，写着“Alt+1 我的调用图”。这个图例的意思是，当我们选中调用图的其中一个节点或边时，在 Sublime Text 或直接在视图上按下 Alt+1，会把整个调用图显示出来。如下图，剩下一个函数的时候，按下 Alt+1，整个调用图又出来了：



如果一个函数同时属于多个调用图，则会存在多个图例。同一条边如果属于多个调用图，则会用不同颜色的虚线展示，如下图。按下图例指示的快捷键，即可调出对应的调用图。



Scheme 选项卡中的 Show Scheme 和 Delete Scheme 按钮，分别是显示和删除列表中选中的调用图。

善用调用图这个功能，可以加快看代码的效率。例如将一个功能主要的函数调用保存为

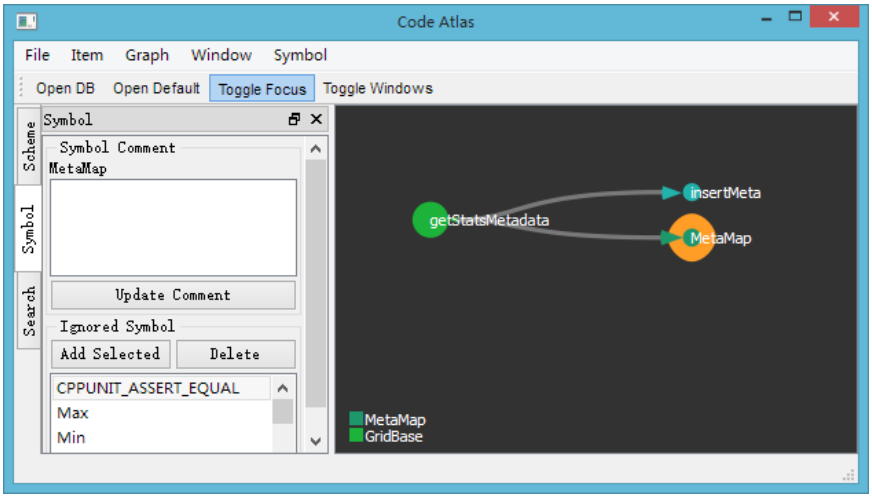
一个调用图，以后浏览这个功能的相关代码，或是对这个功能做修改时，即可通过 Alt+1/2/3/4/5... 快速调出先前保存的整个调用图，再利用这个调用图在函数之间快速切换。

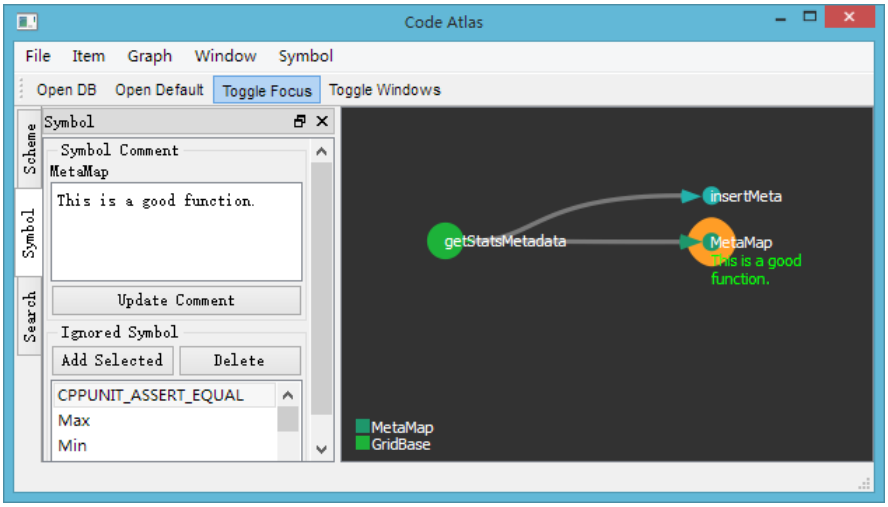
为函数添加说明

有些函数看完就忘了，怎么办呢？可以把你的说明写在调用图中，下次看到说明就不会忘记啦。

> *1. 选中要添加说明的函数*
>
> *2. 点击 Toggle Windows，选中 Symbol 选项卡*
>
> *3. 在 Symbol Comment 一栏中输入你的说明，点击 Update Comment*

可以发现函数名下面多了一排绿色的字，这排字就是我们刚才打的说明了。



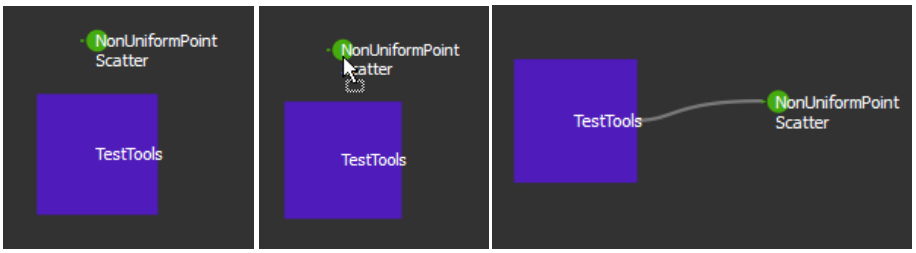


手动加边

有些时候，Understand 的分析结果会不准确，例如虚函数往往存在多个实现版本，此时可能需要手动为函数指定合适的实现版本。边的具体含义并没有规定，用户可以按照自己的使用习惯，用边来指明一些自定义的关系。

> *1. 在视图中显示函数/变量/类 A 和 B。*
>
> *2. 在视图中，用鼠标右键把 A 拖到 B，即可添加一条从 A 到 B 的边。*

下图演示了操作流程。



至此所有功能就介绍完了。