# 第一章 欢迎学习VTK

欢迎阅读《VTK用户指南》。VTK是一个开源的、面向对象的计算机图形、可视化和图像处理的开源的软件系统。虽然VTK比较庞大、复杂，但是当你了解了它基本的面向对象的设计和实现的方法以后，你会发现VTK被设计得如此的简单且易于学习。这本《VTK用户指南》的目的就是帮助你学习这种方法论，让你熟悉各种重要的VTK类。

如果你读过这本书的其他版本，你会发现我们已经开始使用Edition Number而不是VTK的Version Number来区分更新这本书了。这本书是《VTK用户指南》的11th Edition。《VTK用户指南》的出版已经超过十一个年头，而这个版本的用户指南的出版正值VTK面世以来的第十六个年头。这本书是在VTK 5.6版本发布前出版的，我们可以很有信心地说，这本书中所列举的所有素材对VTK将来要发布的任何版本都会是有效的。VTK把向后兼容这个特性当作重中之重，一些新的特性在将来的VTK发布版本中会加进来，不会在这里罗列出来，但这些相对于目前存在的特性来说还是比较少的。

VTK是一个庞大的系统。正因为如此，不可能完整的将所有的VTK对象和他们的方法都放在这个指南中。不过，这个指南将向你介绍重要的系统概念，带领你沿着学习曲线尽可能快速有效地学习。一旦你掌握这个基础，我们建议你从诸多可用的资源中获取知识点，这些资源包括VTK Doxygen文档(<http://www.vtk.org/doc/nightly/html/>)以及VTK用户社区。

VTK是一个开源的软件系统。这意味着成千上万的像你一样的慷慨的开发人员和用户对这个系统做出了贡献。如果你发现VTK是一种有用的工具，我们鼓励你贡献出Bug 修正代码、算法、想法或者是应用程序给VTK社区。你也可以支持像Kitware这样的商业公司来开发或者加入新的特性和工具。

## 1.1 用户指南组织结构

这本用户指南划分为三个部分，每一部分都细分为若干独立的章节。第一部分是VTK的一般介绍，包括（下一章）如何在计算机上安装VTK，主要是安装预编译库和可执行程序，以及直接从源代码编译。第一部分也介绍了基本的系统概念，包括系统架构概览、如何使用C++、TCL、Java和Python这些编程语言建立应用程序。从某种意义上讲，第二部分是用户指南的核心，因为在这一部分里用到了很多例子来说明重要的系统特性。第三部分是针对VTK高级用户的。如果你是一位开发者，第三部分解释了如何创建你自己的类库、扩展系统以及与各种视窗和GUI系统进行整合。第19章包含了简单的VTK对象关系概览的对象图表、过滤器（Filters）的概要列表和VTK文件格式的描述。最后是本书的索引部分，它可以方便的检索该用户指南的内容。

## 1.2 如何学习VTK

VTK 的用户主要分为两类，第一类是使用C++创建类的开发人员，第二类是使用C++类库来建立完整的应用程序的开发人员。类开发人员必须精通C++，如果你正在扩展或修改VTK，你必须同时熟悉VTK的内部结构和设计（第三部分所涵盖的内容）。应用程序开发人员可以使用C++或者不用C++，因为编译好的VTK C++类库已经“封装”了TCL、Python、VB和Java等解释性语言。不过，作为一名应用程序开发人员，你必须了解VTK对象的外部接口和他们之间的关系。

学习VTK的关键是熟悉它的对象并知道如何组合他们。如果你是一位VTK初学者，先从安装VTK软件系统开始学习。如果你是一位类开发人员，你可能会下载源代码并编译它们。应用程序开发人员可能仅仅需要预编译库和可执行程序。我们建议你通过研究例子程序（如果你是一位应用程序开发人员）和源代码（如果你是一位类开发人员）来学习这个软件系统。第三章中会对VTK软件系统中的一些关键概念作一个简要概览，然后在第二部分的例子中对这些内容进行回顾。你也可以运行随源代码发布的各种各样的例子，可以在源代码的VTK/Examples目录中找到（请查看文件VTK/Examples/README.txt来获取各个子目录下示例的描述）。在源代码发布目录中也有很多测试用例，如在VTK/Graphics/Testing/Tcl以及VTK/Graphics/Testing/Cxx，虽然大多数测试例子没有被载入文档的测试脚本，但是他们对了解VTK中的类如何组合使用是相当有用的。

## 1.3 VTK软件系统组织结构

接下来简要描述一下各个源码目录中的内容，列举各个目录的软件功能、文档内容以及数据。

**获取软件**

可以通过以下两种方式访问VTK的源代码。

1. 从VTK网站(http://www.vtk.org)下载官方发布的源码。

2. 通过Git访问VTK的源码(git clone http://vtk.org/VTK.git VTK)。

这本用户指南假定你选择的VTK源码是官网发布的版本。本书的编写时间是2009年9月份。在编写本书时我们考虑了VTK 5.4以及即将发布的VTK 5.6里的一些新特性。注意，本书的内容对VTK的后续版本也是同样适用。还有一点要注意的是，在过去的VTK发布的版本号中，我们使用了一个主识别号来代表版本的更新(如，VTK 4.4到VTK 5.0)，某种程度上也表示版本的向后兼容性。但是随着版本的频繁发布，我们可能会面临着发布VTK 5.10版本(5.10可能会混淆某些用户，让人误会是5.2的前一个版本，实际上它是在5.8之后发布的)，或者会发布VTK 6.0，而6.0会让人更加清楚版本的向后兼容性，所以将来我们会选择发布VTK 6.0而不是VTK 5.10。当更新版本的VTK发布时，你还是可以阅读本书的内容，尽管某些最新的特性没有在本书中提到，但本书中的资源对于将来VTK发布的版本仍会是适用的。如果想了解将来要发布的VTK的新特性，可以关注VTK的邮件列表(<http://www.vtk.org/VTK/help/mailing.html>)或是Kitware内部发行的免费刊物《Source》 (<http://www.kitware.com/products/thesource.html>)。

我们强烈建议你使用VTK 5.4或者更高的VTK官方发布版本。官方发布版本比Git版本稳定性、一致性要更好以及经过严格的测试。当然，如果你一定要用VTK的最新版本，就先查看一下VTK测试公告栏里的信息。VTK使用了Kitware的软件开发流程(Software Process)(http://www.kitware.com/solutions/softwareprocess.html)。在你更新Git仓库时，先确认测试公告栏是否为“绿色”，如果不是绿色，有可能所更新的版本就会不稳定。(见”Kitware’s Quality Software Process”一节，了解更多VTK代码质量控制公告栏等信息。)

**目录结构**

开始学习VTK之前，首先有必要对VTK的目录结构做一个整体的认识。即使你是选择预编译二进制文件的安装方式，了解这部分内容也有助你在VTK源码中更方便快捷地查找例子、代码以及文档文件。下面是VTK文档的组织结构：

* InfoVis – 包含了用于信息可视化的类。
* Views – 包含了对数据可视化的类，包括：过滤器(Filter)、可视化(Visualization)、交互(Interaction)和选择(Selection)。
* VTK/CMake – 用于跨平台编译的配置文件。
* VTK/Common – 核心的类
* VTK/Examples – 包含按主题归档的详细注释的例子。
* VTK/Filtering – 可视化管道中与数据处理有关的类。
* VTK/GenericFiltering – VTK与外部模拟包的接口适配框架。
* VTK/Geovis – 用于地形可视化的视图、数据源和其他对象。
* VTK/Graphics – 处理3D数据的过滤器(Filter)。
* VTK/GUISupport – VTK与MFC和Qt等用户图形界面开发工具的接口类。
* VTK/Hybrid – 依赖于其他多个目录的综合类(complex classes that depend on classes in multiple other directories)。
* VTK/Imaging – 图像处理过滤器。
* VTK/IO – 用于读写数据的类。
* VTK/Parallel – 支持并行处理类，如MPI。
* VTK/Rendering – 用于渲染的类。
* VTK/Utilities – 支持像expat, png, jpeg, tiff和zlib等软件库。Doxygen目录包含了从源代码里生成Doxygen文档的脚本和配置文件。
* VTK/VolumeRendering – 用于体绘制的类。
* VTK/Widgets – 3D Widget类。
* VTK/Wrapping – 支持对Tcl, Python和Java的封装。

**文档**

除了这本用户指南以及《The Visualization Toolkit An Object-Oriented Approach to 3D Graphics》，你还可以获取到其他的文档资源。

**Doxygen文档**。Doxygen文档是学习VTK非常重要的文档工具，每个web页上面都详细地描述了VTK里每个类的用法、数据成员、成员函数等。文档里也列出了类的继承图以及与该类协同工作的其他类的关系图表。每个文档都有链接到其他类以及源代码的超链接。Doxygen文档可通过http://www.vtk.org/doc/nightly/html/在线访问。注意你所用的VTK版本应该与Doxygen文档版本一致。

**头文件**。每一个VTK类都是由一个.h头文件以及.cxx的实现文件构成。所有在头文件中记载的每一个成员函数都是对该类所提供的方法的一种快速检索。(事实上，Doxygen文档也是通过这些头文件生成的)。

**数据**

VTK的程序用例以及测试例子里所用到的数据都可以在http://www.vtk.org上下载到，或者通过Git下载。如何使用Git可以访问VTK官方网页。

## 1.4 其他资源

这本用户指南仅仅是学习VTK的可用资源之一，下面列出了一些在线资源、服务、应用程序以及其他的出版物，相信这些资源对你学习VTK会有很大的帮助。

* 与这本指南配套的教科书《The Visualization Toolkit An Object-Oriented Approach To 3D Graphics》深入讲解了许多VTK里使用到的算法、数据结构等。这本书是Kitware公司出版的，你可以在Kitware网站或者亚马逊网站上购买到。
* 《Source》是Kitware公司按季度发行的内部刊物，里面涵盖了Kitware公司的所有开源项目。一旦有新的功能加进VTK时，会有相关的文章发表在《Source》上。与VTK有关的其他有用的资源、入门等东西也会在季刊上发表。你可以在线访问《Source》(<http://www.kitware.com/products/thesource.html>)，或者通过邮箱注册请求发送每期的《Source》季刊。
* VTK官网上含有大量的资源，如在线帮助文档、Wiki、常见问题解答、dashboard、bug跟踪以及vtkusers邮件列表的搜索引擎。不管是初学者还是经验丰富的开发者，Doxygen文档都是不可多得的好资源。
* vtkusers邮件列表可以让用户和开发者提问题以及接收别人的解答、发布更新、bug修复和改进的内容，以及提出改进系统的建议。请访问VTK官网查询如何加入vtkusers邮件列表(注册地址：<http://public.kitware.com/mailman/listinfo/vtkusers>)。
* Kitware公司开设了专业的培训课程。涵盖了Kitware公司的所有开源项目，包括VTK，ITK，CMake和ParaView等，每年在纽约北部地区举办两次。另外Kitware公司也可以根据你的开发团队提供一些定制的培训课程。更多信息请访问Kitware官网或者直接发邮件到courses@kitware.com咨询。
* Kitware公司也提供了商业支持和顾问。主要包括VTK专家辅助项目开发、根据你的项目详细说明书Kitware为你提供大规模的顾问等。你可以访问Kitware官网或者发送邮件到sales@kitware.com获取更多信息。
* ParaView是用VTK实现的针对科学可视化的应用程序。可从paraview.org上下载。使用ParaView来学习VTK是比较不错的选择，因为你可以通过用户图形界面来熟悉VTK的大部分常用的功能。这也是非常有参考价值的，你可以加载自己的数据，看看主要有哪些可视化技术是可以使用以及你自己希望具有什么样的性能等等。
* CMake是用于跨平台构建编译环境的开源工具。对于VTK初学者来说，只要掌握非常少的关于CMake的知识就可以在标准的Windows, Linux或Mac OSX平台上成功编译VTK；对于一些高级用户来说，他们会觉得CMake在开发过程中起到的作用是相当明显的，要把VTK移植到非标准的平台上时就需要掌握更多的关于CMake的知识。可以访问CMake官网cmake.org了解更多的信息。
* CDash是VTK采用的用于源码测试的开源平台。你可以在VTK官网上找到VTK的测试公告栏(由CDash提供技术支持)。公告栏显示了在不同的平台上经过测试的一些结果。对于一些在非标准的平台上进行开发的人员可以贡献他们的测试结果。10.8节中会有详细内容介绍Kitware的软件开发流程。

翻译者：罗火灵

QQ: 348774226

Email: shuiling119@hotmail.com