



10年口碑积累，成功培养50000多名研发工程师，铸就专业品牌形象
华清远见的企业理念是不仅要良心教育、做专业教育，更要做受人尊敬的职业教育。

《VXWORKS 内核、设备驱动与 BSP 开发详解》

作者：华清远见

专业始于专注 卓识源于远见

第 1 章 使用 Tornado

本章简介

Tornado 是 WindRiver 公司为开发 VxWorks 提供的集成开发环境。在经历了最初的几个版本后，WindRiver 公司推出了 Tornado 2.2。相对于之前的版本，Tornado 2.2 的易用性和功能等得到了大幅的增强。由于目前 Tornado 2.2 的普及率非常高，因此本书以 Tornado 2.2 为基础介绍 Tornado 开发环境的使用。

专业始于专注 卓识源于远见

1.1 概述

虽然 Tornado 2.2 相对于 Tornado 2.0 在版本号上只有微小的提升，却更加易用，功能上更为丰富，界面也更加友好。这种改进体现在很多细节上。

Tornado 2.2 提供了编译器的选择。Tornado 2.0 中使用的编译器是 GNU 的编译器，如果需要更换其他的编译器，必须使用命令行进行编译，也就无法使用集成开发环境所带来的便利。Tornado 2.2 在这方面有了很大的改进，第 2 章中会给出将默认的 GNU 编译器换成 Diab 编译器的例子。

更为人性化的编辑器。虽然 Tornado 2.2 所带有的编辑器依然很不方便，但是比起 Tornado 2.0 来说已经有了长足的进步。Tornado 2.2 支持使用“Ctrl+F”组合键弹出搜索框的方式，而在 Tornado 2.0 中，只能使用“Alt+F3”组合键开始搜索。更为重要的是，Tornado 2.2 支持自定义的编辑器，用户可以设定任意一种自己喜欢的编辑器来替代 Tornado 自带的编辑器，并从集成开发环境中调用它。

Tornado 2.2 的变化还有很多，这里没有必要一一列举。总而言之，在条件允许的情况下，请尽量放弃使用 Tornado 2.0 和 VxWorks 5.4，以减少不必要的麻烦。

在 VxWorks 6.0 诞生之际，WindRiver 公司提供了另一种开发环境“WorkBench”。这种开发环境基于开源工程“Eclipse”，习惯 Java 开发的朋友们对这个开发环境应该非常熟悉。WorkBench 的使用极为方便，然而由于软件升级等问题，国内还没有大量普及 VxWorks 6.0 和 WorkBench。

1.2 Tornado 的安装目录

Tornado 2.2 既可以运行在 Windows 平台上，也可以运行在 UNIX 平台上。对于 Windows 平台，它支持 Windows NT 4.0、Windows 2000 和 Windows XP；对于 UNIX 平台，它支持 Solaris 2.7、Solaris 2.8、Solaris 2.9 等。下面将以 Windows 平台为例，介绍 Tornado 2.2 的目录结构。

Tornado 2.2 在 Windows 平台上的安装过程和常见软件的安装过程几乎完全相同，通过逐步提示就能顺利完成。Tornado 2.2 可以被安装到系统中的任何目录下，但它不允许路径中包含空格，因此不能安装到诸如“C:\Program Files\”之类的位置。

安装完成后，Tornado 的主安装目录下除包含 license.dat、readme.txt 等文件外，还包含如图 1.1 所示的几个子目录。

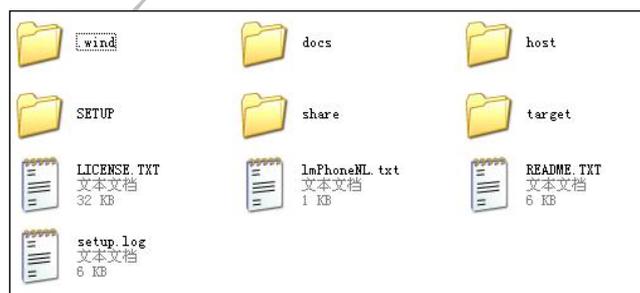


图 1.1 Tornado 的目录结构

- .wind 目录——用于保存 Tornado 软件的注册信息以及卸载软件所必须的信息。在 UNIX 平台下，该目录是隐藏的。
- docs 目录——用于存放用户手册，其中包含了非常实用的帮助文件。这些文件都可以通过 Tornado 的菜单“Help→Manuel Index”访问。
- host 目录——用于存放在主机端（即安装和运行 Tornado 的计算机）运行的程序（如编译器等）、库文件、源代码、脚本文件等。
- SETUP 目录——保存了安装盘的原始文件，用于修复可能损坏的任何安装后的信息。
- share 目录——保存一些系统工具会使用到的文件及源程序。

- **target** 目录——用于存放将要在目标机（正在开发或者将要开发的、将要运行 VxWorks 的系统，通常是一个嵌入式计算机）上运行的二进制文件，以及要生成这些二进制文件所必需的库文件、源文件、脚本文件、工程项目文件等。Tornado 所包含的 VxWorks 操作系统就是以库文件的形式保存在该目录下的。在上述目录中，最常用到的是 **host** 目录和 **target** 目录中的文件。如果需要修改开发环境文件或者为开发环境添加功能，都需要将文件放置在 **host** 目录中；而 **target** 目录则包含了目标机 **BSP**、目标机驱动程序等重要内容。主机部分的文件修改可能性较小，而目标机部分的文件则需要时常与其打交道，图 1.2 所示就是 **target** 目录中所包含的内容。

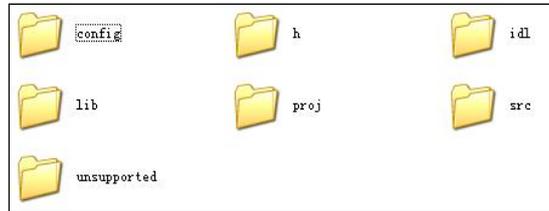


图 1.2 Target 目录结构

- **config** 目录——所有 **BSP** 所在的目录，Tornado 环境会自动找到该目录下所包含的每一个 **BSP**。
- **h** 目录——包含了目标机系统中所需要的所有头文件，与 **src** 目录相对应。
- **lib** 目录——包含 VxWorks 核心文件编译而成的库文件。
- **proj** 目录——包含一些例子工程，用户在生成工程时的默认路径也指向这里。
- **src** 目录——包含了各种源程序。这个目录可以说是 VxWorks 的学习宝藏，很多 API 的实现都可以在这里找到源程序。
- **unsupported** 目录——包含一些 VxWorks 暂时不支持的网卡驱动程序。

1.3 创建一个工程

本节所讲述的内容非常基本，任何使用过 VxWorks 的朋友对此都非常熟悉。如果没有疑问可以跳过本节继续阅读后面的内容。

1.3.1 可加载工程和可引导工程

在创建一个新的工程之前，首先需要分清 VxWorks 中两种工程的异同点。打个比方，Tornado 中的可引导工程相当于 PC 机上的 Windows 操作系统，而且这是一个可剪裁的操作系统。用户可以根据需要进行剪裁和配置，然后重新编译。当然，重新编译一个操作系统可能需要较长的时间，而且，要想重新加载新编译的操作系统，就必须重新引导计算机。可加载工程就相当于 Windows 中用户自己编写的应用程序或第三方应用程序。应用程序必须依赖操作系统才能运行，但是由于它只包含应用代码，因此代码量通常比较小，编译时间短。在操作系统已经启动的情况下，可以随时启动应用程序。

由上面的比喻可以看出，VxWorks 将工程分为两种的目的有很多，其中之一就是开发上的方便。在开发过程中，没有必要花费大量的时间一遍一遍地编译、重新下载可引导的 VxWorks 映像，将可加载映像编译下载即可。使用可加载映像大大节省了编译内核所需要的时间。

可引导工程是带有 VxWorks 内核的工程。将这样的工程编译完成之后，即可通过引导程序下载到目标机上执行 VxWorks。在生成这样的工程之后，Tornado 会将 **BSP** 中的部分文件加入到工程中，同时通过 **usrAppInit.c** 文件提供用户程序入口。编译可引导工程能生成 VxWorks 操作系统映像，该映像包含了 VxWorks 内核以及用户的应用程序。通过 Tornado 2.2 所提供的工具，可以对可引导工程进行裁剪。每次加载可引导的 VxWorks 映像都需要重新引导。

相对应的，可加载工程仅包含用户程序。可加载工程需要可引导工程的支撑，只有在运行有可引导工程且连接了主机开发工具的目标机上，可加载工程才可以通过调试工具下载、运行。每次修改可加载工程后，

不需要重新引导目标机，将编译后的目标文件重新下载即可。

这样，我们就知道可引导工程带有 VxWorks 内核，它需要较长的编译时间，可以提供裁剪等功能；而可加载工程仅包含用户文件，它需要可引导工程的支撑，编译较快但无法裁剪。那么，何时使用可加载工程，何时使用可引导工程呢？这需要将开发过程分为三个阶段。

第一个阶段是在硬件已经准备好的情况下加载操作系统。在这一阶段，软件刚刚开始开发，需要在硬件平台上将 VxWorks 系统运行起来。软件开发的目標就是可以通过配置好的引导程序在目标机上运行 VxWorks 操作系统。进行这一阶段时，不涉及到应用软件开发，却需要一个 VxWorks 系统，要建立一个配置好的、稳定的可引导工程，为下一阶段的开发做好准备。

第二阶段是软件开发过程。这一阶段需要对所开发的软件进行修改、调试，所以整个过程需要频繁地编译程序、下载程序。此时，需要创建可加载工程以减少编译、下载、重新引导目标机的时间，为调试提供方便。

第三阶段是在软件开发完成后开始的部署过程。在这一阶段中，开发所需要考虑的问题集中在如何减小映像尺寸和如何提高整机性能上。这时，需要将第二阶段所开发的应用程序与 VxWorks 系统进行整合，并且适当地裁剪系统，需要创建新的可引导工程并整合第二阶段的成果。

下面将举例说明如何生成不同的工程。创建工程的过程中需要注意两个问题：其一，不要在路径中包含中文名称，因为中文的路径可能会导致编译过程出错；其二，不要因为工程之间有联系就将其建立在相同目录，因为有些同名文件可能会相互覆盖。

1.3.2 实例：创建可加载工程

本小节将通过实例描述创建可加载工程的过程。

1. 材料准备

练习本小节的实例，需要准备下列材料。

- 安装有 Tornado 2.2 的主机一台。

2. 实例说明

在 Tornado 的主界面下，选择“File→New Project”，即弹出创建新工程的窗口，如图 1.3 所示。

3. 操作步骤

(1) 提供必要的信息。

在图 1.3 所示的窗口中双击“Create downloadable application modules for...”，弹出创建可加载工程的参数搜集窗口，如图 1.4 所示。窗口中的 3 个编辑区分别用来输入新建工程的名称、存储位置和文字描述，单选按钮用来选择将新建工程添加到当前 workspace 或者为新建工程创建新的 workspace。

这里推荐将工程目录设置到自定义的目录下，否则，在 Tornado 出现问题时可能导致工程丢失。

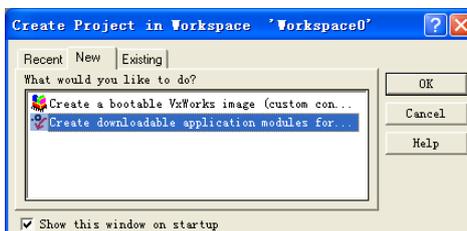


图 1.3 创建工程窗口

(2) 为工程选择工具链。

设置好新建工程的基本信息后，单击“Next >”按钮即进入下一步，为工程选择工具链，如图 1.5 所示。



图 1.4 设置工程的基本信息



图 1.5 选择工具链

在 VxWorks 系统中，可加载工程相当于一个应用程序，它不包括操作系统映像，因而不需要指定 BSP。但是该工程对应的产品、可下载的模块，仍然是一个二进制模块，它需要运行在特定的目标机上，而且在不同体系结构的目标机之间是不可移植的，因而必须为工程指定目标机和编译器。

针对某一特定目标机体系结构的一整套编译工具统称为工具链。工具链通常包括编译器、汇编器、连接器、二进制库、反汇编工具、二进制目标文件格式转换工具等。

Tornado 提供两种选择工具链的方式：一是“A toolchain”，直接选择相应的工具链；二是“An existing project”，选择与某个已有工程相同的工具链。

常见的工具链有 GNU 工具链和 DIAB 工具链两个系列，均包括针对多种不同体系结构的工具链。另外，SIMNTgnu 是一个特殊的工具链，它所针对的目标机是一个仿真器，它是运行在开发主机上的一个应用程序。

这里我们选择 SIMNTgnu 作为示例。

(3) 生成工程。

选择完工具链之后，单击“Next >”按钮进入下一步，即弹出一个信息汇总的窗口，如图 1.6 所示。确认所有信息无误后，单击“Finish”按钮，完成一个可加载工程的创建。

新建的可加载工程如图 1.7 所示。这是一个空的工程，暂时不包括任何程序，其中的 Project1.out 是该工程的最终产品。

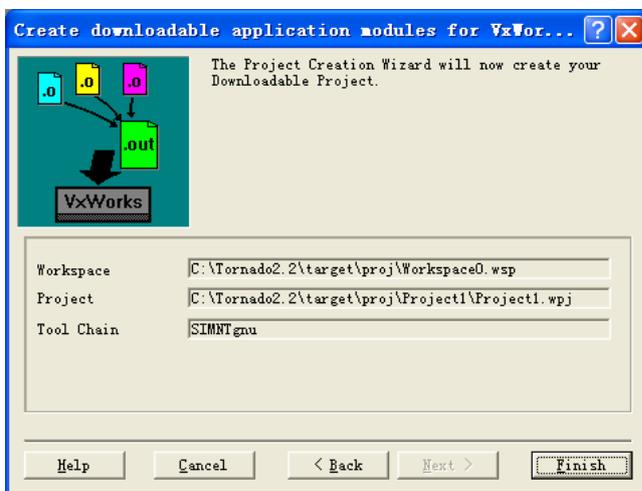


图 1.6 信息汇总和确认窗口

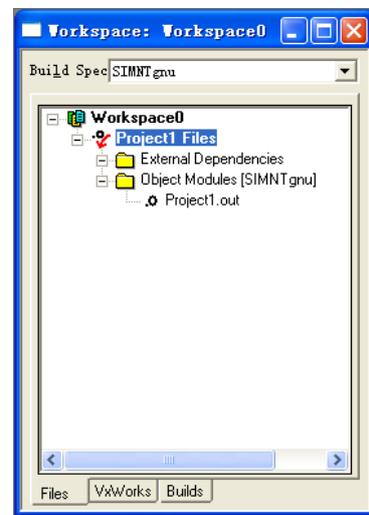


图 1.7 空的可加载工程

1.3.3 实例：创建可引导工程

创建可引导工程的步骤与创建可加载工程的步骤类似。

第一步中，在选择新建工程类型时选择“Create a bootable VxWorks image ...”，然后提供工程的基本信息，与可加载工程完全相同。

第二步中，可引导工程要提供的不仅仅是一个工具链，而是一个 BSP 和相应的工具链，如图 1.8 所示。与可加载工程类似，创建向导也提供了两种方法来选择 BSP：“An existing project”（基于已有工程）和“A BSP”（基于某个 BSP）。这两个选择分别应用于开发的不同阶段。

“An existing project”的选项更适合于开发的第三阶段。在这个阶段，用户已经有了第一阶段调试好的可引导工程，只需根据这个工程生成新的工程就可以继承原有工程的那些自定义内容，而不必重新进行配置。当然，这一阶段也可以从某个 BSP 直接创建新的工程。

选择完 BSP 之后，就是对所输入信息的汇总和确认，与可加载工程完全相同。

新建的可引导工程如图 1.9 所示。可以看出，它已经包含了一些源文件和头文件，这些文件是从 BSP 中来的。根据不同的设置，可引导工程的最终产品可以是 VxWorks、vxworks_romResident、vxworks_rom 或 vxworks_romCompress，分别表示下载型、rom 驻留型、rom 型和 rom 压缩型的 VxWorks 映像。图 1.9 描述的是一个以仿真器作为目标机的可引导工程，因此 VxWorks 映像文件名有后缀.exe。

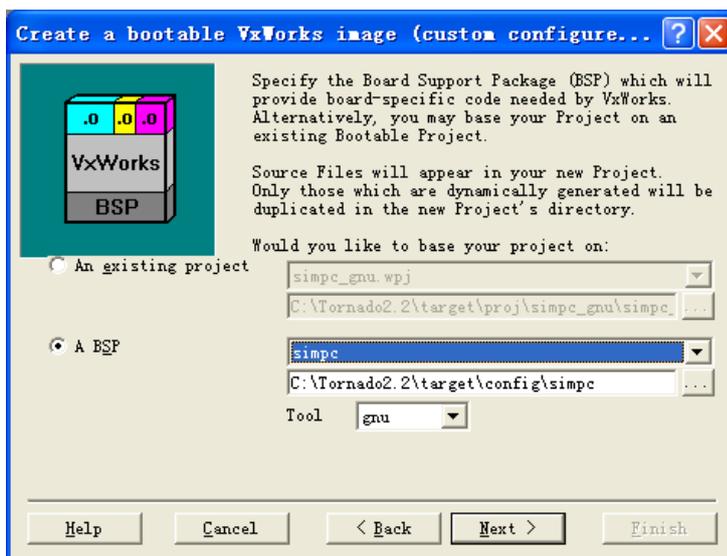


图 1.8 为可引导工程选择 BSP

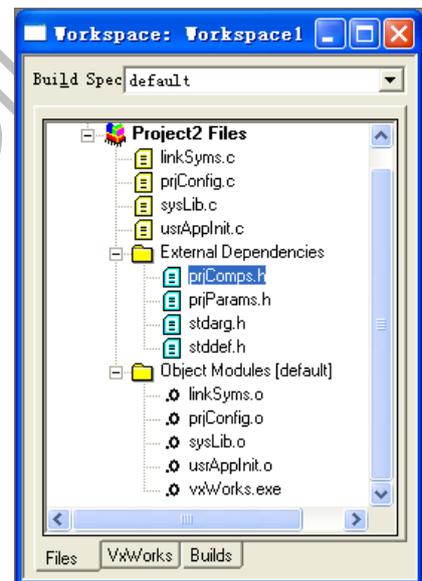


图 1.9 新建的可引导工程

1.4 更加有效率地工作

虽然有很多改进，Tornado 2.2 开发环境还是无法完成更多的事情。例如，在编辑过程中无法自动完成输入，在查看过程中无法跳转到函数定义，等等。为了最大效率地进行开发，需要对整个开发环境进行彻底的改造。

1.4.1 开发环境基本设置

通过 Tornado 的菜单“Tools→Options...”，可以打开整个开发环境的设置窗口，如图 1.10 所示。

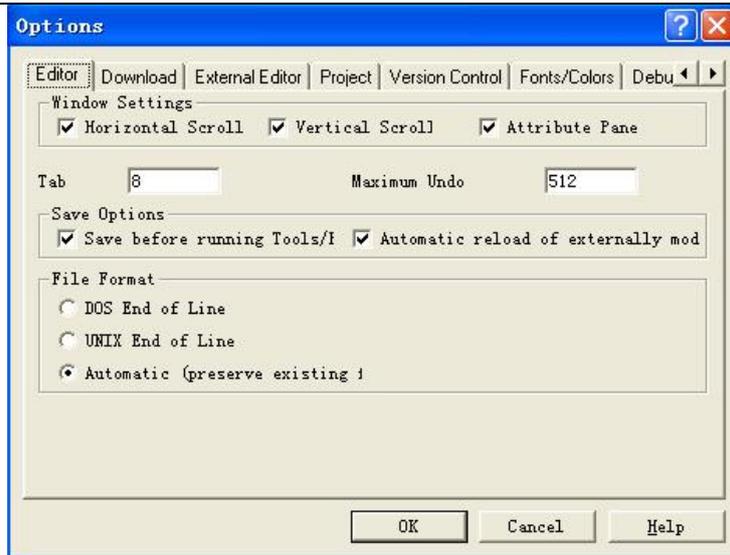


图 1.10 Tornado 设置窗口

该窗口有多个选项卡，从左至右开始分别是设置编辑器 Editor、设置下载内容 Download、设置外部编辑器 External Editor、设置工程 Project、设置版本控制 Version Control、设置字体颜色 Fonts/Colors、设置调试器 Debug、设置登录器、重设安装信息和许可。

事实上，在很多设置上使用默认值就足以很好地工作了。只是根据每个人的习惯使用不同的设置，可以提高程序开发的效率。下面即将讨论的内容并不包括全部的设置，只涉及一些能够提高开发效率或较难理解的设置。

- 设置编辑器。设置编辑器页面的内容仅局限于 Tornado 自带的编辑器，这些设置包括除了字体、颜色的其他内容。关键的两个选项是“Save Options”（保存设置）中的两项：“Save before running Tools/Builds”（在运行工具、创建之前保存）和“Automatic reload of externally modified files”（自动重载外部修改的文件）。使用这两个选项可以避免修改过的文件被编译器或其他工具忽略。
- 设置下载内容。该选项卡主要提供了下载时对符号表处理方法的选择。“System Symbol Table”用于选择系统符号表，窗口中列出了两个选项，实际上这两个选项隐含了另外两个选项，也就是共有 4 个选项：选择 LOAD_LOCAL_SYMBOLS 表示只将本地符号表添加到系统符号表中，选择 LOAD_GLOBAL_SYMBOLS 表示只将外部符号表添加到系统符号表中，两者都选表示添加所有符号表（LOAD_ALL_SYMBOLS），两者都不选表示不添加任何符号表（LOAD_NO_SYMBOLS）。“Common Symbol Resolution”用于设置通用符号表匹配关系，这里的 3 个选项是互斥的，分别表示是否匹配符号表。“C++ Constructors”用于开启或关闭 C++ 构造管理。“Miscellaneous”用于确定装载模块时是否对“WTX_MSG_MODULE_LIST”可见。
- 设置工程。选择“Enable extended Build Options”选项，将会在“Build”菜单中增加一个“Standard BSP Builds”选项。这个选项是 Tornado 2.0 遗留下来的内容，与“Build bootrom”菜单项功能相同。选择“Show advanced component properties”选项将会在元件的属性页上增加“Definition”页，该页提供了元件在描述文件中的相关定义。接下来的设置是默认的工程创建目录。工程目录和源文件目录可以根据创建工程的需要设置，默认的目录是 Tornado 安装目录下的 target\proj 目录。
- 设置调试器。该选项卡主要设置调试器的界面，包括以什么样的格式显示数据等内容。建议选择“Hexadecimal display”，以便于调试时查看。
- 设置登录器。登录器是 WindRiver 提供的，用于解析目标机名称和目标机地址的软件。这里可以设置使用本地登录器或远程某台机器上的登录器。

1.4.2 实例：设置一个更熟悉的编辑器

“Options”窗口中的“External Editor”选项卡允许用户设置一个更熟悉的编辑器，用来取代 Tornado 自带的编辑器，该选项卡如图 1.11 所示。本实例的目的就是将 Source Insight 3.5 设置为 Tornado 的外部编辑器。

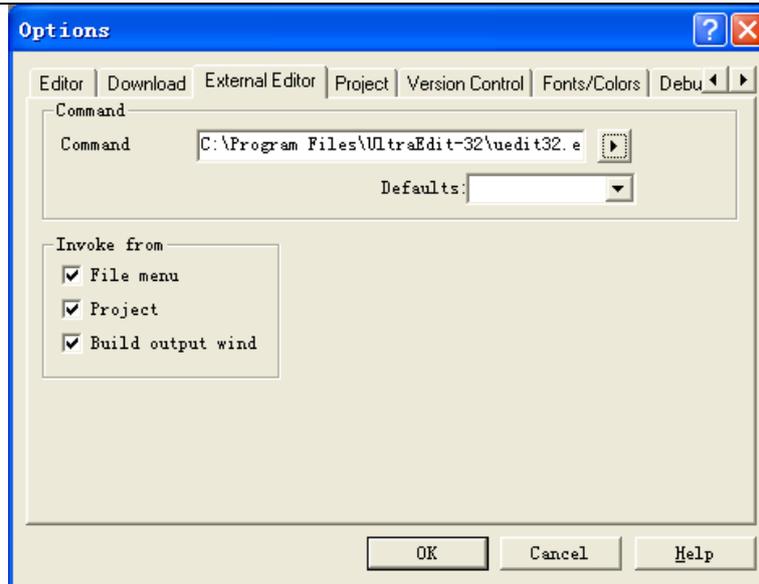


图 1.11 设置外部编辑器

1. 材料准备

练习本小节的实例，需要准备下面的材料。

- 安装有 Tornado 2.2 和 Source Insight 的主机一台。

2. 实例说明

“External Editor”选项卡包括以下功能。

- 命令文本输入框，用来输入编辑器的路径和命令行选项。输入框右边有一个按钮，单击按钮，出现一个弹出菜单，有 3 个选项。“Browse...”选项用来选择编辑器路径；“File name”选项用来在命令行中添加待编辑文件；“Line number”选项用来在命令行中添加当前要编辑的行号。
- 默认选项的下拉列表，用来选择常用的编辑器，包括 vi、CodeWright 和 Visual SlickEdit3 个选择。
- 复选列表，用来选择调入外部编辑器的时机。Tornado 中调用编辑器打开一个文件包括 3 种时机：通过菜单“File→Open...”打开文件；在 workspace 或者工作区窗口中选择打开相应的文件；在编译的输出窗口中双击错误打开相应的文件。该复选列表的“File menu”、“Project”和“Build output wind”三个选项分别用来设定在相应的时机是否调用外部编辑器。

3. 设置命令行

在命令文本输入框中输入下面的命令。

```
D:\Program Files\Insight3\insight3.exe -i +$lineno $filename
```

其中，“-i”、“+”都是由 Source Insight 规定的。“-i”表示在已存在的编辑窗口中打开文件，“+”表示紧跟其后的参数是需要跳转到的行号。命令行中“\$lineno”、“\$filename”是由输入框右边的按钮提供的，分别由单击“File name”和“Line number”完成。

经过这样的设置，每次打开文件的时候，Tornado 都会调用 Source Insight。

1.4.3 使用 Tcl 语言定义工具

除了使用菜单“Tools→Customize”的功能定义新的菜单项之外，Tornado 还提供了使用 TCL 语言为其增加功能的特性。使用这个特性必须先掌握的是 TCL 语言。事实上，关于 TCL 语言本身就可以写出一本书来，这里仅介绍一下 TCL 语言并粗略描述如何将新的工具增加到 Tornado 下。

TCL 最早被称为“工具命令语言”（Tool Command Language），它是基于字符串的解释型命令语言，经常被用于快速原型开发、脚本编程、GUI 和测试等方面。它在语法上的简单性和语义上的感官相通的方式使其易于学习。TCL 脚本由一个或多个命令组成，命令之间用换行或分号分隔。命令格式形如：

```
command arg1 arg2 arg3.....
```

命令是基本的执行单元。一个命令后跟随着一个或多个字，这些字是命令的参数，字之间用空格或 tab 分隔。其语法结构与 C 语言相似。

TCL 语言是 Tornado 为用户预留的工具接口，在很多 Tornado 工具上也可以看到它的影子，如在 Shell 下键入“?”命令可以进入 TCL 命令解释提示符。通过 TCL 语言编写的程序，用户可以将编写的程序添加到 Tornado 的工具条或菜单中。根据下面的步骤可以将一个应用程序（wtxapp.exe）添加到工具条中。

首先，建立如下文件：

```
installDir\host\resource\tcl\app-config\Tornado\01Wtxapp.win32.tcl
```

并在该文件中键入如下内容。

```
set wtxappBitmap [wtxPath host resource bitmaps Tornado]Wtxapp.bmp
controlCreate launch [list toolbarbutton -name wtxapp \
                        -callback launchWtxapp \
                        -tooltip "Launch Wtxapp" \
                        -bitmap $wtxappBitmap]

proc launchWtxapp {} {
    global env
    # 获取选定的目标机名称
    set tgtsvr_selected [selectedTargetGet]
    if {$tgtsvr_selected == ""} {
        messageBox -stopicon "Select a target first."
        return
    }
    # 试图连接工具 wtxapp（这个工具是用户自己编写的）
    if {[catch {wtxToolAttach $tgtsvr_selected wtxapp} attachRetVal]} {
        messageBox -stopicon "Tool attach failed. $attachRetVal"
        return
    }
    # 获取目标机服务器信息
    if [catch "wtxTsInfoGet" info] {
        if {[lindex $info 3] == "(SVR_TARGET_NOT_ATTACHED)"} {
            wtxToolDetach
            messageBox -stopicon "Target not attached"
            return
        }
        messageBox -stopicon "Couldn't obtain target server information"
        return
    }
    set tgtsvr_user [lindex $info 8]
```

```

set tgtsvr_lock      [lindex $info 1]
# 检查目标机服务器是否被锁
if {$tgtsvr_lock == "locked"} {
    messageBox -stopicon "Target server locked by $tgtsvr_user."
    return
}
# 检查是否有权限访问目标机服务器
if {$tgtsvr_lock == "locked"} {
    messageBox -stopicon "Access not authorized."
    return
}
#开始运行目标机服务器
if {[catch {toolLaunch "Wtxapp" "wtxapp $tgtsvr_selected" \
    $env(WIND_BASE)/host/x86-win32/bin 0 0 0 0 0} result]} {
    puts "\nError: wtxapp launch failed: '$result'\n"
}
}
    
```

其中，controlCreate 命令所使用的-bitmap 参数所指向的位图文件为（大小与图标相同）：*installDir*\host\resource\bitmaps\Launch\controls\wtxapp.bmp。这个位图用于工具条中的显示。整个程序中，实际起作用的语句是 controlCreate 命令，它创建了工具条、工具条上的图标、图标名称等，而过程 launchWtxapp 用于响应工具条中图标的动作。

面向 Tornado 的工具开发包含很多内容，Tornado 所带文档“Tornado API Guide”中有专门的描述，这里不再深入讨论，有兴趣的读者可以自行阅读。

联系方式

集团官网：www.hqyj.com

嵌入式学院：www.embedu.org

移动互联网学院：www.3g-edu.org

企业学院：www.farsight.com.cn

物联网学院：www.topsight.cn

研发中心：dev.hqyj.com

集团总部地址：北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址：北京市海淀区西三旗悦秀路北京明园大学校区，电话：010-82600386/5

上海地址：上海市徐汇区漕溪路 250 号银海大厦 11 层 B 区，电话：021-54485127

深圳地址：深圳市龙华新区人民北路美丽 AAA 大厦 15 层，电话：0755-22193762

成都地址：成都市武侯区科华北路 99 号科华大厦 6 层，电话：028-85405115

南京地址：南京市白下区汉中路 185 号鸿运大厦 10 层，电话：025-86551900

武汉地址：武汉市工程大学卓刀泉校区科技孵化器大楼 8 层，电话：027-87804688

西安地址：西安市高新区高新一路 12 号创业大厦 D3 楼 5 层，电话：029-68785218