



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做好良心教育、做专业教育，更要做好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 4 章 管理工程——Work Space

本章简介

工程管理器是 Tornado 集成开发环境的核心功能。通过工程管理器，开发者可以组织工作空间和工程，裁剪、配置 VxWorks 组件，以及定义自定义编译规则。

专业始于专注 卓识源于远见

4.1 管理文件

与 Windows 下的资源管理器相似，工程管理器也采用了树形结构。树形结构的根节点是工程所在的工作空间，接下来的一级是工程，再接下来就是源文件、相关的头文件以及编译后生成的目标文件。

4.1.1 文件的位置

工程管理器在组织文件方面为用户提供了很大方便。在建立工程的过程中，用户只需将源文件加入到工程中，其他的工作都会由工程管理器协助完成。

创建一个工程之后，首先要将源文件加入到工程之中。利用鼠标右键单击工程名称或工程中的任意文件名，都可以弹出快捷菜单，选择菜单中的“Add Files...”即可选择要加入工程的源文件。

在添加文件的时候要注意，工程管理器中需要添加的是源文件，而不是头文件。至于头文件，工程管理器可以通过依赖分析来自动添加。

可以看到，树形结构中工程的下一级还包含了两个节点，分别是“External Dependencies”和“Object Modules”。其中，前者包含源文件所必需的头文件，后者包含编译后生成的目标文件。将这些文件放置在节点中的目的是减少这些不常用文件出现的机会。

4.1.2 文件依赖关系

右键单击工程的任意位置，可以在快捷菜单中看到“Dependencies”选项。选中该选项，Tornado 会弹出依赖分析窗口，如图 4.1 所示。

依赖分析功能用于分析源文件与头文件的关系。经过分析之后，头文件会被自动加入到工程的“Dependencies”节点下。

依赖分析窗口中有一个“Advanced”选项，该选项的目的是在依赖分析的过程中免去对那些不会变化的头文件所在目录的分析工作。这样做可以避免大量重复的搜索工作。

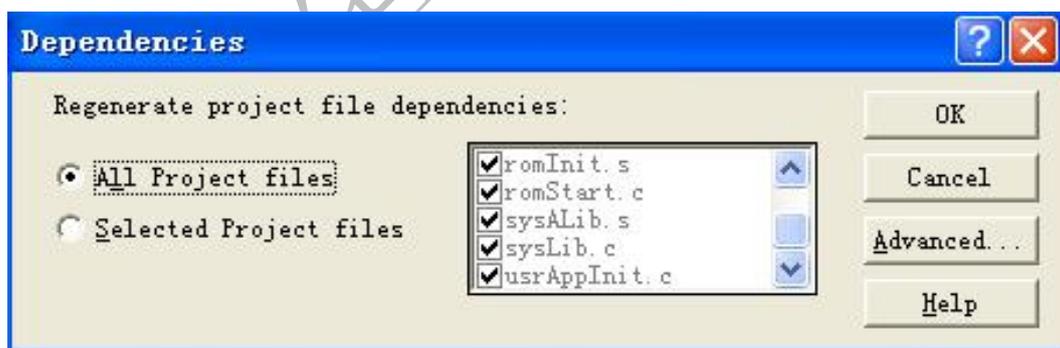


图 4.1 依赖分析窗口

4.2 管理组件

VxWorks 是一个微内核操作系统，除了一个极小的 Wind 内核以外，它的绝大部分功能都是以组件（components）的形式提供的。对 VxWorks 的定制和剪裁就是通过对组件的增删实现。

单击 Workspace 窗口下部的 VxWorks 标签，即出现 VxWorks 的组件管理窗口，如图 4.2 所示。它也是一个树形结构，可以很方便的查看工程中当前包含的组件，并进行组件的增加和删除操作。

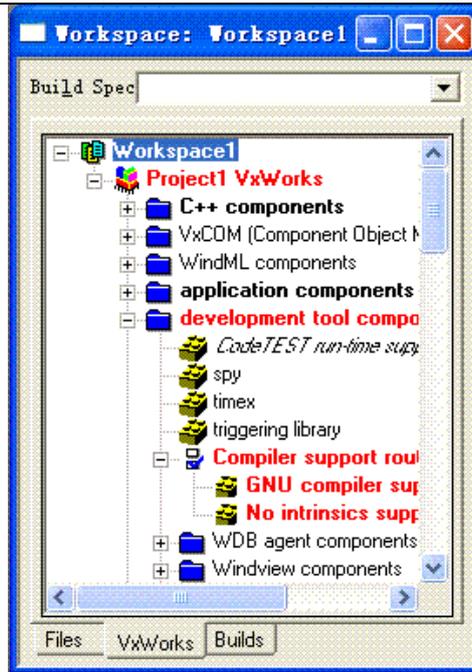


图 4.2 组件管理窗口

组件的管理工具为组件裁剪提供了方便。Tornado 组件管理工具可以用来在组件的裁剪过程中检查裁剪的错误，也可以通过分析程序自动裁剪组件。根据关联关系，组件管理工具可以在用户包含某个组件时自动指出其相关组件并将之包含。

组件管理器所列出的组件都是由工程所选定的 BSP 提供，并在工程文件中维护。打开 BSP 所在目录可以看到，目录下包含若干以 cdf 为后缀的文件。这些 cdf 文件描述了所有组件的所在位置、依赖关系等信息。关于 cdf 文件的详细情况，我们将会在 BSP 与开发环境的结合章节中讨论。

创建工程后，工程文件将会从 BSP 中获取组件信息并将其保存下来。在裁剪 VxWorks 映像之后，工程文件会记录下裁剪后的组件情况并将裁剪的情况应用到编译的过程中。每次应用工程管理器，它都会从工程文件中读取组件信息，再结合 BSP 提供的依赖关系，形成 VxWorks 映像的组件构成，即由用户可以看到的组件组成树型结构。

4.2.1 VxWorks 的各种组件

VxWorks 的组件树按照组件的功能来分类，包括 C++、VxCOM、WindML、application、development、hardware、network、obsolete 和 operating system 几个类别，用户也可以根据需要添加自定义分类。系统原有的分类情况说明如表 4.1 所示。

上表列出的只是系统默认会带有的一些组件，根据用户安装的组件不同，有些组件不会出现在工程管理器中。有些 BSP 经过修改之后也会提供一些独特的组件。

根据组件当前状态的不同，Tornado 采用不同的字体和颜色显示组件：黑体字表示已被包含到工程中的组件，正常字体表示未被包含到工程中的组件，斜体字表示尚未安装的 VxWorks 组件；红色表示相应位置的组件选择存在冲突，绿色字体表示工程管理器查找到的目标。

表 4.1 VxWorks 基本组件说明

分 类	分类说明
C++	与 C++ 相关的组件，其中包含 C++ 运行库、标准函数库等
VxCOM	与 COM 模型相关的组件，包含 COM 核心等
WindML	与 WindML 相关的组件，包含 WindML 的一些配置、驱动等

application	与用户应用程序相关的内容，仅包含用户入口函数的定义
development	开发工具组件，包含 WDB、WindView 等支持
hardware	硬件驱动配置组件，配置 VxWorks 支持哪些硬件
network	网络组件，配置 VxWorks 所支持的网络协议、网络支持等
obsolete	过期组件，这里包含早期的串行驱动程序
operating system	操作系统组件，包含了运行库的配置

4.2.2 查找组件并修改属性

除了从组件树中根据组件所属的分类找到某个组件以外，还可以通过组件对应的宏来查找组件。在组件树的任意位置单击鼠标右键，选择弹出菜单中的“Find Object...”，会弹出一个查找组件的窗口，如图 4.3 所示。在“Type”栏中选择“Component”，在“Object”栏中输入要查找组件对应的宏，单击“Find”按钮即可找到相应的组件。

在查找组件窗口中，不但可以通过选择“Type”为“Component”来选择组件，还可以通过选择“Type”为“Folder”、“Parameter”或“Selection”来选择组件文件夹、组件参数或互斥组件。

在组件树中双击某个组件，会弹出一个组件属性窗口，如图 4.4 所示。在该窗口中，可以查看和修改组件的属性。

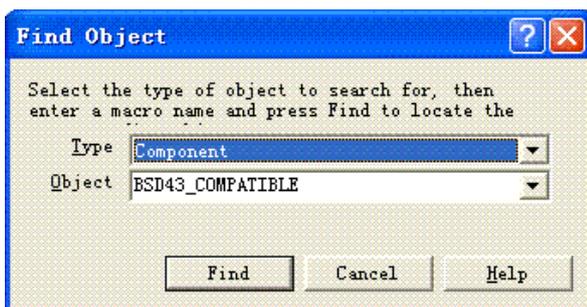


图 4.3 查找组件

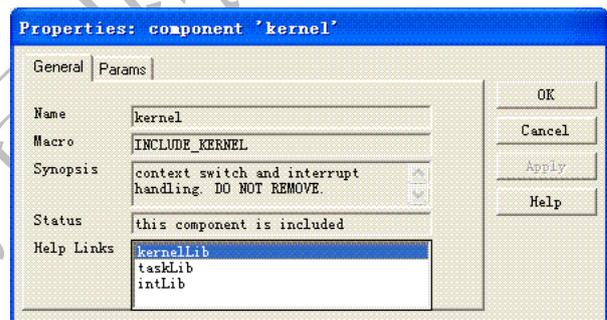


图 4.4 组件的属性窗口

组件属性窗口含有两个选项：“General”和“Params”。General 选项卡用来查看组件的基本信息，包括组件名称、对应的宏定义、描述信息、当前状态、相关的库文件等。在组件相关的库文件中双击某个库文件，即可通过浏览器打开相应库文件的用户手册。

VxWorks 中的某些组件包含参数，此时在组件属性窗口中会有一个 Params 标签。打开 Params 选项卡，可以查看该组件包含参数的宏定义和对应的值，并可修改这些值，如图 4.5 所示。用户从这里修改的参数将会自动代替 BSP 中原始的定义。这样的设计免去了修改 BSP 文件导致出错的可能，简化了 BSP 修改工作。

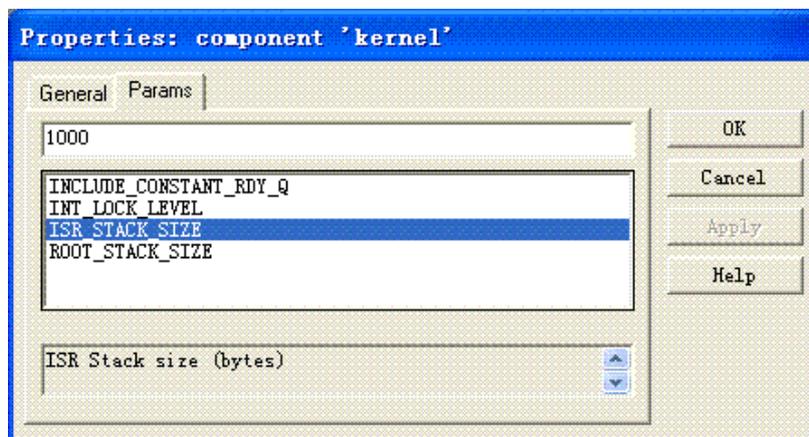


图 4.5 修改组件参数

4.2.3 裁剪组件

在 Tornado 下对 VxWorks 的组件进行剪裁非常方便。在工程管理器的相应组件上单击鼠标右键，在弹出菜单中选择“Include”即可将该组件添加到工程中，选择“Exclude”即从工程中删除该组件。通过对某个组件文件夹进行操作，也可以将整个文件夹下所有的组件添加或删除。

VxWorks 的组件之间通常会存在依赖关系，Tornado 能够根据声明自动处理这些依赖关系。在添加一个组件时，Tornado 能够自动将该组件所依赖的组件添加到工程中；在删除一个组件时，Tornado 自动将依赖于该组件的所有组件删除。例如，ANSI time 组件依赖于 POSIX clocks 组件，因此在添加 ANSI time 组件到工程中时，POSIX clocks 组件（如果还没有包含的话）会被自动添加到工程中；如果要从工程中删除 POSIX clocks 组件的话，ANSI time 组件（如果已经包含）也将被删除。

VxWorks 的有些组件之间存在互斥关系。这些互斥的组件不能同时被包含在同一个工程中。如果它们被同时包含，Tornado 会提示组件冲突，并将产生冲突的组件的完整路径用红色标记出来（见图 4.2）。

4.3 编译选项

工程管理器中的第三页提供了用于控制编译、链接的工具，使用这个工具可以部分地修改编译规则、编译参数、链接顺序等内容。如图 4.6 所示，编译选项中包含多种预知的编译方式选项，其中的粗体字表示当前正在使用的是 default 编译方式。

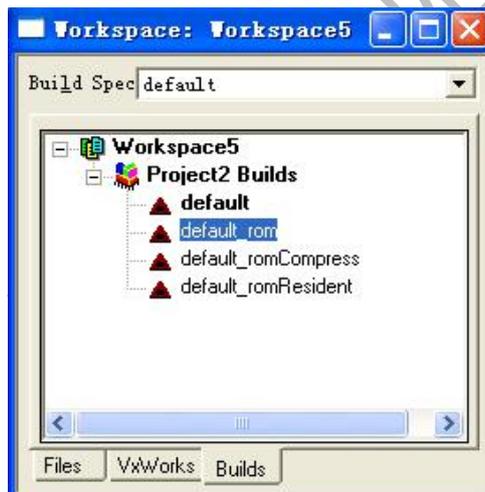


图 4.6 编译控制窗口

4.3.1 编译规则

要查看编译方式属性，可以打开编译方式设置窗口，如图 4.7 所示。编译方式窗口包括规则设置、宏定义设置、编译器设置、连接顺序设置、汇编器设置和链接器设置。

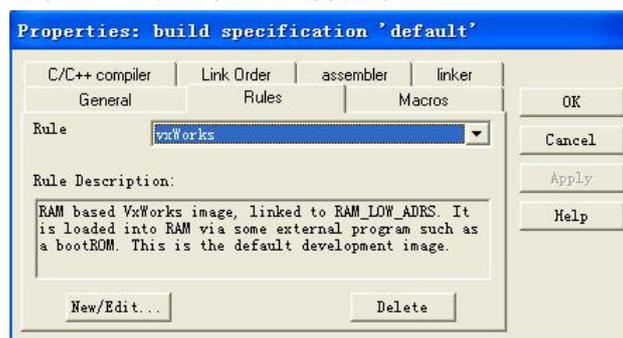


图 4.7 编译方式设置窗口

在编译规则设置选项卡，可以选择编译生成不同类型的目标文件，包括十六进制目标文件、二进制目标文件以及 ELF 格式的目标文件。映像的加载方式也可以根据实际情况进行选择，常用的类型包括压缩型映像、非压缩性映像、ROM 驻留型映像等。对于可引导工程，各映像类型的说明参见表 4.2。

对于可加载工程，可选的编译规则有 4 种，参见表 4.3 的说明。

表 4.2 各种映像类型说明

映像类型	说明（没有后缀的文件格式为 ELF）
VxWorks	基于 RAM 的 VxWorks 映像，链接到地址 RAM_LOW_ADRS，由类似于 bootRom 的程序加载到内存中。这是默认的开发类型映像
VxWorks_rom	由 ROM 开始运行的基于 RAM 的 VxWorks 映像。整个映像由 ROM 引导代码复制到内存中运行。这种映像通常在引导时较慢，但执行时较快
VxWorks_romCompressed	从 ROM 中引导的基于 RAM 的压缩型 VxWorks 映像。该类型映像的大小约为原映像的一半，但是由于需要在引导时解压缩，所以引导速度很慢，其运行速度与 VxWorks_rom 类型相同
vxWorks_romResident	ROM 驻留型映像。这种映像的程序段保留在 ROM 中运行，只将数据段复制到 RAM 中，所以这种映像具有最快的加载速度和最慢的执行速度。这种映像非常适合于内存资源有限的应用
*.bin	以 bin 为后缀的文件表示生成格式为二进制格式
*.hex	以 hex 为后缀的文件表示生成格式为 Intel 十六进制格式

表 4.3 可加载工程编译规则

规则类型	说明
*.out	编译工程文件，将其生成单一的可加载目标文件。该规则执行静态 C++构造
*.pl	编译工程文件，将其链接在一起。该规则不执行静态 C++构造，支持子工程但不支持加载。生成的目标文件可用于其他工程
archive	编译工程文件，并生成后缀名为.a 的库文件
objects	仅生成源文件对应的.o 目标文件

4.3.2 宏定义

在宏定义选项卡，Tornado 提供了各种用于控制编译过程的宏定义，如图 4.8 所示，其中一些宏定义相当于 makefile 中的变量。

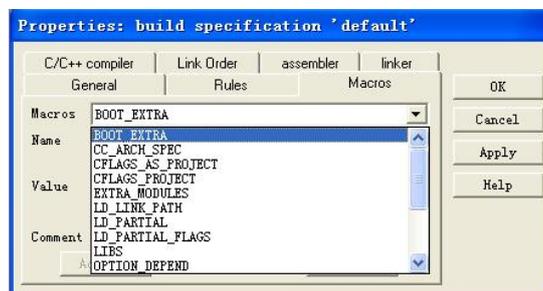


图 4.8 宏定义选项

这里值得注意的有表 4.4 所列出的几个定义。

表 4.4 部分重要的编译宏定义

宏定义	说明
EXTRA_MODULES	链接到 VxWorks 映像的其他外部目标模块
LD_PARTIAL	用于将几个目标文件合并到一个目标文件的链接命令

LIBS	链接到 VxWorks 映像的库文件。该参数与 EXTRA_MODULES 参数类似，一个针对模块文件，一个针对库文件
PRJ_LIBS	链接到可加载工程的目标文件或库文件
POST_BUILD_RULE	编译完成后需要执行的 Shell 命令
RAM_HIGH_ADRS	装载映像数据段的 RAM 地址。该定义必须与 config.h 中的定义相同
RAM_LOW_ADRS	VxWorks 映像的 RAM 中使用的基地址。该定义必须与 config.h 中的定义相同

在该选项卡中，还可以通过“Add/Set”按钮添加自定义的新宏，但这个功能通常用不到。

4.3.3 其他选项

设置窗口中的其他功能，包括编译器设置、链接器设置、汇编器设置都需要修改对应工具的参数，感兴趣的读者可以查阅这些工具相关的文献。

值得一提的是，编译器下的参数“-ansi”用于指示编译器按照 ANSI C 标准进行编译。如果去掉这个选项，C 语言源程序中可以使用“//”形式的注释；否则，只能用“/* */”来添加注释。

另一个参数是“-I”，该参数指明了编译器搜寻的路径。如果发生编译时找不到文件的情况，可以试着添加适当的路径来指示编译器的工作。

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