
2006' 第二届中国开源软件竞赛组织委员会

(文件七)

作品说明书

作品名称: CnPack IDE 专家包

作品版本: 0.8.1

摘 要

项目规格书

分析设计文档

用户使用手册

测试说明书

开发单位: CnPack 开发组

2006年 10 月 5 日制

目 录

| | |
|------------------------|----|
| 目 录..... | 2 |
| 第一部分 项目摘要 | 4 |
| 项目简介..... | 4 |
| 主要面向的用户群..... | 4 |
| 开发组介绍..... | 4 |
| 功能摘要..... | 5 |
| 用户反馈..... | 6 |
| 第二部分 项目规格书 | 7 |
| 摘 要..... | 7 |
| 一、引言..... | 7 |
| 1、编写目的..... | 7 |
| 2、背景..... | 7 |
| 3、参考资料..... | 7 |
| 4、术语和缩写词..... | 7 |
| 二、项目概述..... | 7 |
| 1、开放源码许可证声明..... | 7 |
| 2、功能与发展前景: | 8 |
| 3、特点..... | 9 |
| 4、应用范围: | 10 |
| 三、项目开发平台和工具..... | 10 |
| 四、应用领域..... | 10 |
| 第三部分 分析设计文档..... | 11 |
| 摘 要..... | 11 |
| 一、引言..... | 11 |
| 1、编写目的..... | 11 |
| 2、背景..... | 11 |
| 3、参考资料..... | 11 |
| 4、术语和缩写词..... | 12 |
| 二、项目需求..... | 12 |
| 1、需求分析..... | 12 |
| 2、软件功能设计: | 13 |
| 三、创作思路..... | 15 |
| 四、系统的组成和结构..... | 15 |
| 五、设计..... | 15 |
| 1、专家包框架设计..... | 15 |
| 2、CnOTACreators设计..... | 17 |
| 3、数据流设计..... | 19 |
| 4、用户界面设计..... | 20 |
| 5、软件运行平台设计..... | 22 |
| 6、部署与安装设计..... | 22 |
| 六、依赖的其它软件..... | 22 |

| | |
|-------------------------|----|
| 七、项目的组织和管理..... | 22 |
| 八、项目管理工具..... | 23 |
| 第四部分 用户使用手册..... | 25 |
| 摘要..... | 25 |
| 一、运行环境/系统使用说明..... | 25 |
| 二、软件安装:..... | 25 |
| 1、从二进制代码包的安装..... | 25 |
| 2、从源文件安装..... | 25 |
| 三、软件使用..... | 26 |
| 1、全局选项..... | 26 |
| 2、语言选项..... | 26 |
| 3、优化选项..... | 27 |
| 4、调试选项..... | 27 |
| 四、疑难问题解答..... | 27 |
| 1、代码格式化工具的问题。..... | 27 |
| 2、代码折叠功能。..... | 27 |
| 3、代码结构高亮功能。..... | 28 |
| 4、用空格完成代码模板的功能。..... | 28 |
| 5、代码输入助手不能自动弹出的问题。..... | 28 |
| 五、成员组成及联络方式..... | 28 |
| 第五部分 测试说明书..... | 29 |
| 一、测试目的..... | 29 |
| 二、测试计划..... | 29 |
| 三、测试环境..... | 29 |
| 四、测试方法以及测试用例..... | 29 |
| 五、测试结果..... | 31 |

第一部分 项目摘要

项目简介

CnPack IDE 专家包是 CnPack 项目组的主力产品之一，英文全称是 **CnPack IDE Wizards**，简称 **CnWizards**。它是一组集成在 Delphi/C++ Builder/Borland Developer Studio的IDE中，用于增强IDE功能、提高IDE的可用性及开发效率的免费、开放源码工具。

主要面向的用户群

CnPack IDE 专家包主要面对使用 Delphi 5 6 7 8、C++Builder 5 6 以及BDS 2005 与BDS 2006 的开发者。CnPack IDE专家包目前的发布版本包括简体中文、繁体中文、英语三种语言。CnPack IDE 专家包经过四年的发展，现已成为国内外首屈一指的大型Delphi系列IDE专家包产品，目前经网站数据统计，用户数保守估计稳定在三千五百左右。

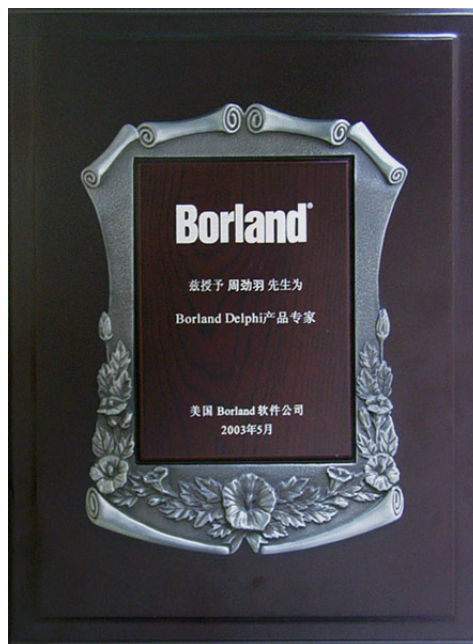
开发组介绍

CnPack 开发组由互联网上热爱程序开发的中国程序员自愿者组成，其目标是开发中国人自己的真正优秀的第三方免费开发包。

开发组当前(截止到 2006.10)有 200 多名正式报名的成员。项目采用网上多人协作的方式开发，属于国内规模最大的开源开发团队之一。

CnPack IDE 专家包是CnPack开发组的主要产品之一，国内外广受好评，用户数目众多，并获得了Delphi出品商Borland公司的官方认可。CnPack项目发起人周劲羽先生曾被Borland中国评为 Borland Delphi 产品专家（国内仅数十人有此殊荣），并赠送正版 Delphi 一套。

以下是项目发起人周劲羽获得的Borland Delphi产品专家证书：



图一 Borland Delphi产品专家证书

功能摘要

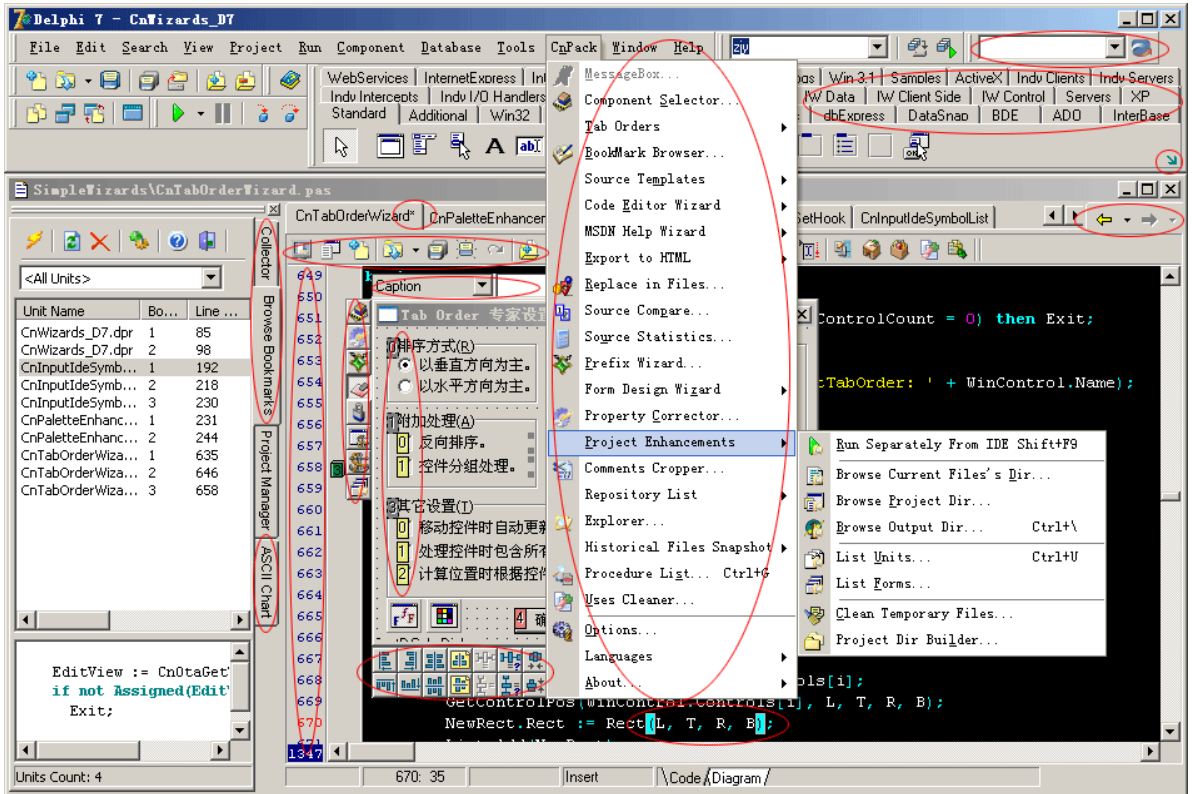
CnPack IDE 专家包在 Delphi/C++Builder/BDS的IDE中提供了十多个可独立设置的子专家，数十种功能，主要包括：

- 代码输入助手
- 增强的单元窗体列表功能
- TabOrder 设置专家
- 窗体设计器的浮动工具栏
- MSDN 帮助集成功能
- 代码编辑器工具栏与功能扩展
- 工程相关的扩展功能
- 组件面板多行显示支持
- 窗体置顶与折叠功能
- 源代码统计功能
- 引用单元清理
- 其他大量增强与改进

以及下列独立运行的工具：

- CnWizards 设置导入导出工具
- DFM 窗体转换工具
- IDE 配置备份/恢复和清除打开文件历史工具
- 调试信息输出单元 CnDebug 与调试信息查看器 CnDebugViewer
- ASCII 字符表

同时，还在开发过程中不断完善了CnPack组件包和公共代码库这批副产品。
如下的Delphi 7 的IDE界面截图部分反映了CnPack IDE 专家包的功能：



图二 部分功能的界面截图

用户反馈

CnPack IDE 专家包在发展的四年期间，受到了广大用户的热情支持。绝大多数用户对 CnPack IDE 专家包给予了很高的评价，众多国内外用户在论坛和邮件中的积极建议、错误报告等便是最好的证明。CnPack IDE 专家包帮助文档中的《有贡献人员列表》中记录了众多为 CnPack IDE 专家包付出过辛勤劳动的支持者，而且还有许多热心的朋友未列入其中。从我们 CnPack 网站的升级访问记录来看，CnPack IDE 专家包的用户数目保守估计在三千五百左右，相信这个用户群规模对于面向开发者的产品来说已经不小了。

第二部分 项目规格书

摘要

本部分介绍了CnPack IDE专家包的整体概况，包括项目规格、策划背景和开发运行平台等，还描述了CnPack IDE专家包作为开源软件发布所依据的协议，因此本部分可供粗略了解CnPack IDE专家包项目之用。

一、引言

1、编写目的

本节旨在介绍 CnPack IDE 专家包的项目规格，以使用户或读者能从外部角度对 CnPack IDE 专家包产品和项目本身的特点、规划等有个整体的了解。

2、背景

CnPack IDE 专家包策划之初是四年前也就是 2002 年，彼时 Borland 的开发工具正大行其道，Windows 平台上的 Delphi/C++Builder 系列 RAD 开发工具以严谨易用的特性，在开发界尤其是国内广受欢迎。而且 Delphi/C++Builder 以其开放的 Open Tools API 接口开放了 IDE 的一些内部特性，为第三方编写插件专家来增加 IDE 的功能提供了可能。

当时国外有几个成熟的插件专家包，比较著名的是 GExperts 和 CodeRush，前者免费、后者收费。两者都是针对国外用户的，界面为英文，对中国用户的开发习惯不太支持，并且两者在功能上还有一定疏漏。CnPack IDE 专家包在这方面不仅弥补了它们的不足，并且在发展过程中已经成长为能与之相抗衡的大型专家包。

3、参考资料

- 《Delphi Open Tools API 帮助》
- 《扩展 Delphi 的 IDE（中译版）》
- 《CnPack 开发方案》
- 《CnPack 发布协议》

4、术语和缩写词

- OTA: Open Tools API, Delphi 系列 IDE 对外提供的编程接口。用户可据此编写插件。
- 专家 (Wizard): 实现了 IOTAWizard 接口的 Delphi 系列 IDE 插件工具。
- 专家包: 一批专家的集合, 可独立安装配置并作为正式产品发布。
- BDS: Borland Developer Studio, Delphi 7 之后的系列 IDE 的官方名称。包括 C#Builder 1.0, Delphi 8/2005/2006 等。

二、项目概述

1、开放源代码许可证声明

CnPack IDE 专家包使用《CnPack IDE 专家包发布协议》进行发布，用户必须遵循此协议才能够使用传播此产品。本协议来源于 OpenSource 的官方开源协议，以下是 CnPack IDE 专家包发布协议的正文：

CnPack IDE 专家包发布协议

CnPack IDE Wizards (以下简称 CnWizards) 的版权 (2001-2006) 属于 CnPack 开发组以及其他提供代码的作者所有。本协议覆盖范围为 CnPack 开发组所书写的代码、文档以及其他工作内容。非 CnPack 开发组部分的版权事宜, 需与各自的作者联系。

本产品及源码发布所根据的协议来源于 <http://www.opensource.org> 的官方开源协议, 具体表述如下:

一、本协议允许所有人自由使用 CnWizards 的二进制发布产品。CnWizards 的源码在未经 CnPack 开发组正式授权的情况下, 禁止用于私有或商业场合。如果遵循以下条款, CnWizards 的源码允许使用于开放源码项目:

二、禁止 CnPack 开发组之外的组织或个人使用 CnWizards 的源码来发布名称中带有 CnPack 或 CnWizards 字样的新产品。如果某产品的源码是根据 CnWizards 的源码修改而来, 则其发布名称必须和 CnPack 以及 CnWizards 有着显著区别, 以使用户确信不是在使用 CnWizards。另外, 此产品的发布版本中必须附带一份未改动的本协议文本。

三、任何基于 CnWizards 源码的改动所衍生的 CnWizards 改良版本, 必须以源码补丁的方式通知 CnPack 开发组, 以保持原作者代码的完整性。CnPack 开发组不对 CnWizards 的任何改良版本提供技术支持。

四、任何使用 CnWizards 的产品必须以兼容于官方开源协议的方式发布, 此官方协议定义可从 <http://www.opensource.org> 获得。注意, CnPack 开发组与 CnWizards 的其他作者共同声明: CnWizards 目前的发布版本不提供任何明确或隐含的担保, 此类担保包括但不限于是否适合商业使用或是否适合某类特定目的用途等。CnPack 开发组不对使用 CnWizards 或源于 CnWizards 的其他产品而可能导致的问题与损失承担任何责任。

如有疑问, 可联系 CnPack 开发组以获悉未尽事宜。

2、功能与发展前景:

CnPack IDE 专家包面向的用户群是 Delphi 5 6 7、C++Builder 5 6 以及 Borland Developer Studio 2005/2006 进行开发的程序员, 经过四年的发展, 目前已经成为了国内外首屈一指的大型 Delphi 系列 IDE 专家包产品, 目前经网站数据统计, 用户数保守估计稳定在三千五百左右。

CnPack IDE 专家包产品日后的发展计划包括三方面, 一是对一些典型的需求但难度较大的功能进行开发, 比如代码格式化、源码结构高亮等; 二是完善对其他开发语种的支持和对 .NET 平台提供支持 (目前只支持 Win32 平台和 Object Pascal/C++ 语言); 三是发展用户协作能力, 建设类似于 CnPack 社区的交流平台。

CnPack IDE 专家包目前的主要功能包括:

- 代码输入助手: 能在编辑器的代码输入过程中自动弹出类似于 Code Insight 的自动完成窗口, 为代码中的输入提供了极大的方便。
- 源码统计专家: 按各种规则统计 IDE 中代码的有效行数。
- 组件前缀专家: 自动规范组件的命名。
- TabOrder 设置专家: 自动设置设计期窗体上各个组件的 TabOrder。
- 窗体置顶与折叠: 给 IDE 各个窗口添加置顶与折叠的功能, 方便使用, 提高屏幕的利用率。
- 组件面板增强: 允许组件面板多行显示, 并提供组件快速搜索的功能。

- MSDN 帮助集成: Borland自身的帮助一向简略, 此功能允许在IDE中调用MSDN或在线搜索帮助信息。
- 增强的单元窗体列表: 取代IDE自身的列表, 提供增强的查找功能。
- 窗体设计期工具栏: 提供方便的设计期窗体组件的处理功能。
- 编辑器工具栏和扩展: 在编辑器中提供强大的辅助处理功能。
- 源码高亮显示: 高亮显示源码中配对的括号, 增加代码的可读性。日后计划进一步扩展到其他标记配对高亮显示。
- 清理引用单元: 清理大型工程中各单元中不需要但uses了的单元。
- 历史文件管理: 记录和管理曾经打开过的文件并可管理快照, 为打开历史文件提供方便。
- IDE 配置备份/恢复: 备份和恢复 IDE 的各种配置。
- Repository 对象库: 提供一些实用的生成单元的模板专家。
- CnDebugger: 调试信息输出接口和调试信息查看器。

此外还有大量增强和扩展功能, 无法在此一一列出。

3、特点

CnPack IDE 专家包在设计之初就定位为符合中国人开发习惯和需求、对 GExperts 和 CodeRush 等大型专家包的功能补充的专家包。历经四年的发展, 目前 CnPack IDE 专家包的发展规模已经不亚于这两者, 部分需求普遍的、成熟的功能 CnPack IDE 专家包也全部实现, 而且 CnPack IDE 专家包自身支持多语言界面, 因此可以说, CnPack IDE 专家包是面向国内国外用户的 Delphi 系列 IDE 的大型专家包。

较之其他 Borland IDE 下的大型专家包产品, CnPack IDE 专家包有以下特点:

- 多语言支持界面: 可动态切换中英文, 为国内外用户同时提供支持。
- 在线更新, 随时报告新版本的发布情况供用户下载安装。
- 上述功能列表中的特色功能, 例如:
 - 代码输入助手: 能在编辑器的代码输入过程中自动弹出类似于 Code Insight 的增强的自动完成窗口, 包括代码、注释、引用等众多类型的模板, 为代码中的输入提供了极大的方便。如下图所示:



图三 代码输入助手的界面

- 源码比较: 能比较不同版本的源码内容, 同时支持 IDE 编辑器缓冲内容和磁盘文件。支持合并操作。
- 源码统计: 按一定规则统计 Pascal 代码和 C/C++ 代码的有效行数和注释行数等。可进行单文件统计、工程内统计、工程组内统计和指定目录统计, 支持工程、工程

组和多文件汇总。

- CnDebugger: 包括调试信息输出接口 CnDebug.pas 和调试信息查看器, 提供多种输出接口, 供后台调试信息输出查看之用。同时支持多进程、多线程的 Delphi/C++Builder 应用程序。
- 附带一批成熟的组件包和公共库供开发者使用。

4、应用范围:

硬件: 32 位x86 平台

操作系统: Windows 95/98/NT/2000/2003/XP

软件环境: 必须安装了Delphi 5 6 7 或C++Builder 5 6 或Borland Developer Studio 2005 2006 的其中一种或多种开发环境。

人机界面: 简体中文、繁体中文和英语。

三、项目开发平台和工具

CnPack IDE 专家包的主要开发平台是 Windows XP/2000, 主要开发工具是 Delphi 5/7 和 BDS 2006, 因为不同版本的 IDE 的专家必须在此 IDE 的编译器下编译, 因而 CnPack IDE 专家包所用到的相关的兼容开发工具包括了所支持的全部 IDE, 包括 Delphi 5/6/7、C++Builder 5/6 和 BDS 2005/2006 等。

CnPack IDE 专家包的源码管理工具是 CVS (包括服务端和客户端), 任务管理与错误跟踪系统则采用和 CVS 配套的、Web 方式的 CVSTrac。自动构建工具使用 want, 安装程序制作工具使用 NSIS 2.0。

四、应用领域

CnPack IDE专家包用于Delphi/C++ Builder/Borland Developer Studio的IDE中, 支持 Delphi 5/6/7、C++Builder 5/6和BDS 2005/2006。

CnPack IDE专家包源码中的CnPack组件包和公共代码库部分则包括了多种VCL组件和函数库, 可用于Delphi/C++Builder应用程序开发中, 具体领域包括Windows平台下的图形图像、多语/本地化、平滑字体、系统设置、历法计算、跟踪调试等。

第三部分 分析设计文档

摘要

本部分是CnPack IDE专家包的分析设计部分，介绍了CnPack IDE专家包的详细内容，描述了需求、设计、建模、部署等各方面的分析与设计细节。

一、引言

1、编写目的

本节旨在介绍 CnPack IDE 专家包的详细内容，描述了需求、设计、建模、部署等各方面的分析与设计细节。由于 CnPack IDE 专家包的庞大，我们不可能也不需要列出每一个子专家或子功能的详细设计内容，因此本文档只针对 CnPack IDE 专家包的一些关键点进行描述。另外 CnPack IDE 专家包已经附带了这批关键点有关的比较详细的需求与设计说明书，本文档由于篇幅关系必定有未能描述到的地方，读者需要参考源码包中的相应文档。

2、背景

编写本部分时，CnPack IDE 专家包已经发布 0.8.1 版本，各种功能与各项支持都已经比较完善，各类文档也有所齐全。正在努力宣传自身，扩大影响面。目前虽然随着 .NET 以及 Java 发展风潮的冲击，受影响的 Borland 已经转让了 Delphi 的生产线，但其后续产品 Turbo Delphi 等仍在积极发展，何况国内还有众多使用 Delphi 进行开发的程序员，因此 CnPack IDE 专家包仍然具有较大的发展潜力可挖。

3、参考资料

外部资料：

- 《Delphi 开发人员指南》
- 《Delphi Open Tools API 帮助》

开发组内部资料：

- 《扩展 Delphi 的 IDE（中译版）》
- 《Delphi 编码规范》
- 《CnPack 开发方案》
- 《CnWizards 概要设计说明书》
- 《CnWizards 框架详细设计说明书》
- 《CnWizards 调试说明书》
- 《CnOTACreators 设计说明书》
- 《CnPack 多语包需求说明书》
- 《CnPack 多语组件包概要设计说明书》
- 《CnDebugger 需求与设计说明书》
- 《CVS 使用说明》

4、术语和缩写词

- 插件 (Add-in), 以设计期包或 DLL 形式被设计期的 IDE 调用的扩展工具。
- 专家 (Wizard), 实现了 IOTAWizard 接口的 IDE 插件工具。
- 专家插件, 本文指 CnWizards 中由专家框架统一管理的专家插件。
- 仓库/模板专家 (Repository Wizard), 用来创建新的单元、窗体或工程的专家。
- 包 (Package), Delphi 中使用的特殊的动态链接库。
- 设计期包 (Design-time Package), 被编译为允许 IDE 在设计期装载的包。
- 运行期包 (Runtime Package), 被编译为允许 DLL 在运行期调用的包。
- 接口 (Interface), Delphi 中使用的 COM 风格的接口。
- 通知器 (Notifier), 由用户实现特定接口并由 IDE 在特定事件中调用的用户对象。
- 创建器 (Creator), 由用户实现特定接口的用于创建新的单元、窗体或工程的用户对象。
- 工程 (Project), Delphi 中的 Project。
- 单元 (Unit), Delphi 中的 Unit。
- 模块 (Module), 对应着一组在 IDE 中打开的逻辑上关联的文件集, 可以是一个单元、包含窗体的单元、工程文件等对象。
- 编辑器 (Editor), IDE 中用来设计和编辑模块的对象。

二、项目需求

1、需求分析

CnPack IDE 专家包的需求分析是开放式的, 也即需求的收集与整理是面向所有 Delphi 系列开发工具用户, 并且和开发同步进行。CnPack IDE 专家包在这几年的开发发展历程中, 不断听取来自于各方面用户的意见和建议, 吸收同类产品的优点, 逐步形成了目前的规模。

CnPack IDE 专家包的设计中部分采用了 ModelMaker 来建模, 但主要还是通过文档、图表和简略代码的方式来描述。由于不涉及到数据库, 因此无数据库方面的设计内容。

CnPack IDE 专家包所遵循的技术规范是 Delphi 系列 IDE 所提供的 Open Tools API 接口规范, 只有遵循此等规范, 才能顺利和 IDE 打交道, 起到增强功能。

CnPack IDE 专家包自身功能的需求规模庞大并且分类复杂, 在此无法一一阐述, 但 CnPack IDE 专家包的一些基础功能, 如框架、多语、调试输出等方面则相对固定, 说明如下:

框架: CnPack IDE 专家框架部分是整个专家包的核心, 它负责各个子专家的生命周期管理, 负责向 IDE 注册各个子专家并提供统一的服务接口。在 IDE 的角度看来, 框架本身只应该是一个普通专家, IDE 负责装载并调用它。但框架这个名义上的普通专家本身实际上起了个专家管理器的作用。CnPack IDE 专家包中的单独存在的子专家在向专家管理器注册后就受专家管理器的控制, 而不是直接注册到 IDE 中, 这样各项子功能的独立性可得到保证, 也易于统一控制、统一提供设置机制等。

多语: 多语言支持这一部分虽然只用在 CnPack IDE 专家包中, 但从需求的高度上来讲, 实际上应该独立成提供设计期运行期支持的多语组件, 而 CnPack IDE 专家包只不过是使用了这批组件而已, 这样有利于降低系统的耦合性和提高可用度。关于多语部分的详细需求, 请参考《CnPack 多语包需求说明书》。

调试: 由于专家是运行期被加载入 IDE 的 DLL, 并且由于 IDE 的环境的复杂性, 传统的单步调试下断点的功能便不再完全适用。CnPack IDE 专家包使用了一套自行设计开发的统一接口 CnDebugger 来在代码中输出关键调试信息并可供查看。和多语一样, CnDebugger 的需求高度也提升到了独立于专家包的程度。它应该能方便地用在普通的应用程序中, 而不是专家包的附庸。CnDebugger 由两部分组成, 一是供开发者使用的, 专用来输出调试信息的接口单元

CnDebug.pas，一是单独运行的独立程序，用来查看CnDebug.pas所输出的信息内容。此部分的需求详情，请参考《CnDebugger需求与设计说明书》。

2、软件功能设计：

CnPack IDE 专家包从功能上来讲有面向用户的功能和面向二次开发者的功能两部分，面向用户的功能以IDE的功能划分为依据，包括内部专家、独立功能的专家、IDE功能的扩展专家、Repository专家、属性编辑器等，以下是这些面向用户的功能所组成的子系统的划分列表：

- 内部专家：内部实现某些功能的专家，无需以专家的形式向用户开放。
 - “关于”子菜单。
 - “Languages”子菜单。
 - “设置”菜单。

- 独立功能专家：这批专家是可从菜单项上直接调用的专家，可划分为一单独的子系统。按其面向的扩展内容不同又可划分为以下几部分：
 - 面向窗体设计：
 - ◆ 窗体设计专家：提供大量组件尺寸、位置、对齐等功能。
 - ◆ 组件选择器：提供按不同方式选择窗体上的组件的功能，在组件众多时很有用。
 - ◆ TabOrder专家：自动设置设计期窗体上各个组件的TabOrder。
 - ◆ 属性修改器：按一定规则批量修改组件属性。
 - 面向代码处理：
 - ◆ 源码统计专家：以各种方式统计源码的行数。
 - ◆ 代码编辑器专家：提供各种代码片断的处理功能。
 - ◆ 书签管理专家：管理源码中的书签。
 - ◆ 源码模板专家：提供一批代码模板并支持自定义模板。
 - ◆ 源码比较：比较磁盘文件和内存编辑器内容等。
 - ◆ 注释清除：按自定义规则清除源码中的注释。
 - ◆ 代码格式输出：将代码按高亮显示方式输出成带格式的带格式的代码文字。
 - ◆ 过程单元列表：列出当前单元的过程供搜索和快速跳转。
 - ◆ MessageBox 设计：以可视化的方式设计弹出消息窗口。
 - 面向文件工程：
 - ◆ 工程扩展工具：提供一批有关工程的扩展功能，包括扩展单元窗体列表、备份、清除临时文件、构建工程目录等。
 - ◆ 打开历史文件：记录曾经打开的文件供重新快速打开，并提供快照功能。
 - ◆ 批量文件替换：弥补 IDE 无批量文件替换的功能。
 - ◆ MSDN 集成：集成 MSDN 工具供搜索，未安装 MSDN 时进行在线搜索。
 - ◆ 引用单元清理：清理大型工程中各单元中不需要但 uses 了的单元。
 - ◆ 文件管理器：在 IDE 中增加一资源管理器窗体方便使用。
 - 面向设置备份：
 - ◆ IDE 设置备份/恢复工具：备份和恢复 IDE 自身的设置。
 - ◆ CnWizards 设置导入/导出工具：备份和恢复 CnWizards 的设置。

- IDE 功能扩展专家：IDE 功能扩展专家是对 IDE 的某部分功能的扩展和增强，往往不属于单独某个独立的功能调用，而是附帶在 IDE 已经实现的功能上，起增强和扩展的作用。

这批功能也可划分为一单独子系统。

- 代码输入助手：扩充 Code Insight 的功能，方便用户代码输入。
 - 代码编辑器扩展：增加代码编辑器工具栏、行号显示、高亮配对显示、浮动按钮等。
 - 窗体设计器扩展：在设计期窗体上增加工具面板，供用户方便设计。
 - 组件面板扩展：增加组件面板的多行显示功能、增加组件快捷搜索功能。
 - 对象查看器扩展：将修改过的属性以黑体显示。
 - 窗口置顶折叠扩展：为 IDE 的窗口增加置顶和折叠的功能，提高屏幕利用率。
 - CPU 调试窗口扩展：为 CPU 调试窗口增加复制汇编代码的功能。
- Repository 专家，这类专家也划分为一子系统，是用来在 IDE 中生成新的单元、窗体、工程等的模板专家：
 - DUnit 用例生成：生成 DUnit 的用例单元或工程，需要安装 DUnit。
 - CnMemProf 工程生成：生成 CnMemProf 工程，需要安装 CnMemProf 内存管理单元。
 - INI 读写单元生成：按 INI 内容生成自动读写 INI 内容的对象类单元。
 - 属性编辑器：属性编辑器是 CnPack IDE 专家提供的，覆盖 Delphi 自身默认属性编辑器的增强功能，这也是一单独子系统，它们包括：
 - 集合属性编辑器
 - 布尔属性编辑器
 - 字符串属性编辑器
 - 工具提示属性编辑器
 - 字符串列表属性编辑器
 - 文件名属性编辑器
 - Constraint 属性编辑器
 - 滚动条属性编辑器
 - 字体属性编辑器
 - Align 属性编辑器
 - 组件名称属性编辑器

而面向二次开发的功能划分的内容中，除了上述功能列表外，还包括了以下五大部分内容：

- CnPack IDE 专家包框架提供的专家注册管理服务。
- 对 Open Tools API 功能接口和 IDE 自身功能的封装。
- 对各种 OpenTools API 通知器/事件的封装。
- 提供多语控制接口，供子专家转换界面语言时使用。
- 其他公共运行库。

公共构件则包括多语组件、窗体 Scale 自适应组件和 CnDebug 调试信息输出接口。此部分内容均可独立于 CnPack IDE 专家包而作为单独的组件库发布。CnPack 多语组件包允许用户在设计期遍历窗体或工程，针对它们生成所有对象所有字符串属性的条目列表，并可将其存入多语存储文件中。同样的方法可以生成多个多语存储文件，每个文件代表工程中的一种语言。运行期修改多语文件即可达到通过切换语言来修改界面字符串的目的，而无需重新编译程序。另外多语组件包在运行期切换语言的功能主要包括：遍历所有窗体，遍历每一个窗体上的每一个对

象的每一个字符串属性，并从多语文件中查找其对应的翻译条目来进行替换。同时，对一些特殊的控件和其下属对象等提供翻译控制处理。

三、创作思路

CnPack IDE 专家包的创作是个长期的集思广益的过程，凝聚了数十位专家开发人员的创意，更凝聚了数百用户的需求提炼。CnPack IDE 专家包在设计之初仅仅只是定位为符合中国人开发习惯和需求、对 GExperts 和 CodeRush 等大型专家包的功能补充的专家包。但在开发和发布的过程中，众多用户和支持者将他们在 IDE 使用过程中发现的不足和需要增强的地方提供给我们开发组，鞭策着我们根据可行性和需求的通用程度制定开发计划并投入许多开发人力，同时部分热心的朋友也捐献了他们自己写作的简单专家代码，授权我们开发组移植入 CnPack IDE 专家包。因此专家包日益壮大，已经成长为可和国际上任何一款 IDE 大型专家包相抗衡的产品。专家包现在的成长局面，和众多热心朋友的支持是分不开的。

在 IDE 这种复杂的环境下进行开发需要许多技巧，同时 Delphi 的运行期包机制和 RTTI 机制也给专家功能的实现思路提供了广阔的空间。譬如在 CnPack IDE 专家包中广泛使用的挂接机制，便是通过 RTTI 和内存地址改写等手段，在 IDE 内部已实现的功能的基础上动态挂接上我们自己的处理代码。目前这种挂接机制也设计成了独立组件，包括 MethodHook、MenuHook、ActionListHook、ControlHook 等。这批 Hook 挂接组件的应用，加上多语、通知器封装、在线更新结合用户统计、代码输入助手等，都是 CnPack IDE 专家包中独具特色的创意的实例。

四、系统的组成和结构

CnPack IDE 专家包的系统结构基本上是按功能划分的，各个子功能间耦合度很低，只有公共的服务部分给众多子专家提供着统一的支持。它们包括：

- 专家管理器
- 内部专家
- 独立功能的专家
- IDE 功能的扩展专家
- Repository 专家
- 属性编辑器
- 公共运行库
- 对 OTA 接口和 IDE 自身功能的封装
- 对各种 OpenTools API 通知器/事件的封装
- 多语支持
- CnDebugger 调试接口

具体细节划分的内容不再和上文重复列举。

五、设计

1、专家包框架设计

CnPack IDE 专家包框架指不涉及具体专家功能，但为所有专家服务，实现公共功能的部分。它的设计是整个设计工作的重点，拥有一个设计良好的框架能给后续的开发带来很多便利。

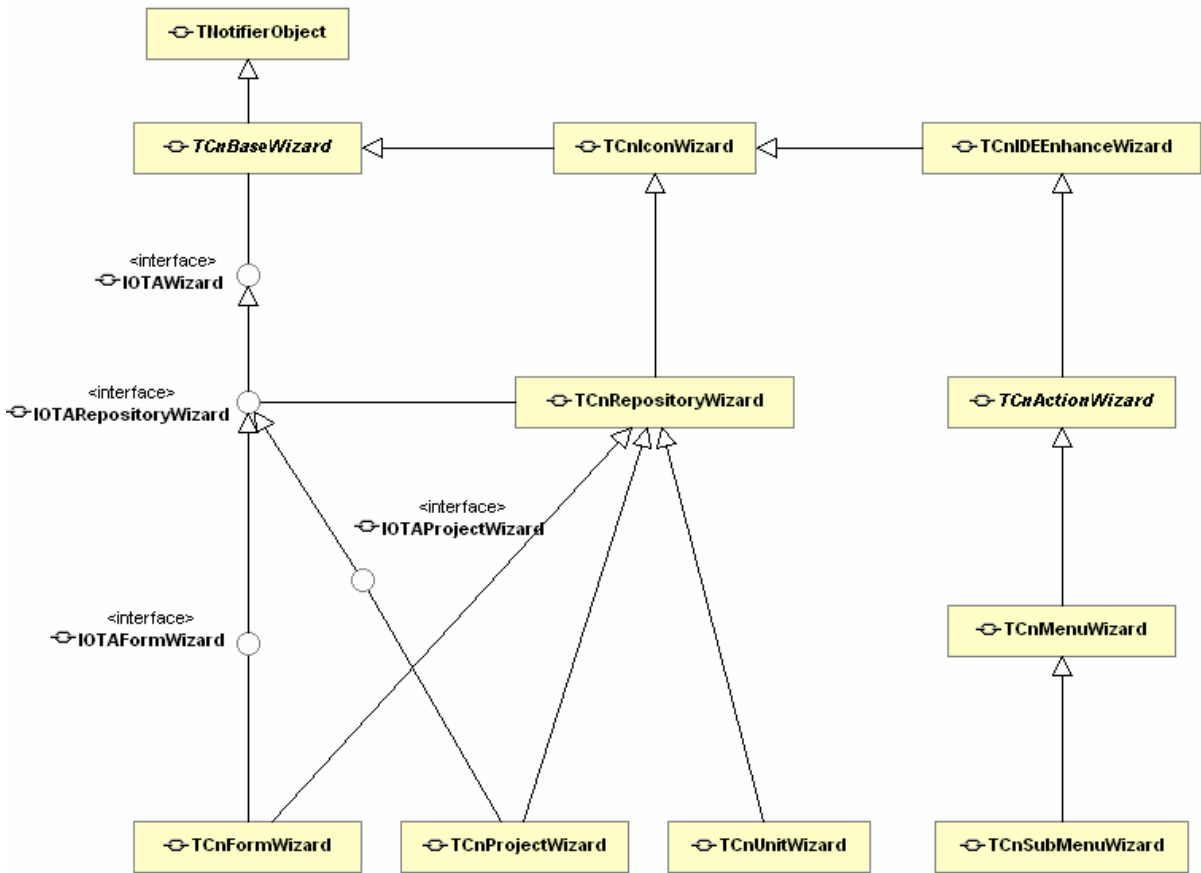
CnPack IDE 专家包的框架负责完成在 IDE 主菜单中插入 CnPack 菜单、配置和管理快捷键、

专家子菜单、提供专家设置界面调用、实现关于窗口，以及定义专家基础类等功能。同时，框架还要实现专家管理器，负责专家列表的维护、注册、释放等操作。

专家基础类包括以下基类：

- TCnBaseWizard，专家抽象基础类，该类实现 IOTAWizard 接口，并实现一些公共的基础方法。
- TCnIconWizard，带图标的抽象基类，允许专家拥有一个专用图标。
- TCnIDEEnhanceWizard，扩展 IDE 功能的抽象基类。
- TCnActionWizard，带快捷键的抽象类，实现 IDE 中的 Action 注册、快捷键绑定等功能，从该类派生出的子类可以用快捷键调用，并可加入到工具栏中。
- TCnMenuWizard，带菜单的基础类，通常 IDE 扩展专家从该类派生，可以通过主菜单来调用。
- TCnRepositoryWizard，模板向导基础类，实现 IOTARespositoryWizrd 接口，其子类放在 IDE 的 New 模板仓库中，通常并不直接用它来派生子类。
- TCnUnitWizard，单元模板向导，从该类派生出的向导用于创建一个单元。
- TCnFormWizard，窗体模板向导，从该类派生出的向导用于创建一个窗体和单元。
- TCnProjectWizard，工程模板向导，从该类派生出的向导用于创建一个工程。

它们之间的关系见如下 UML 类图：

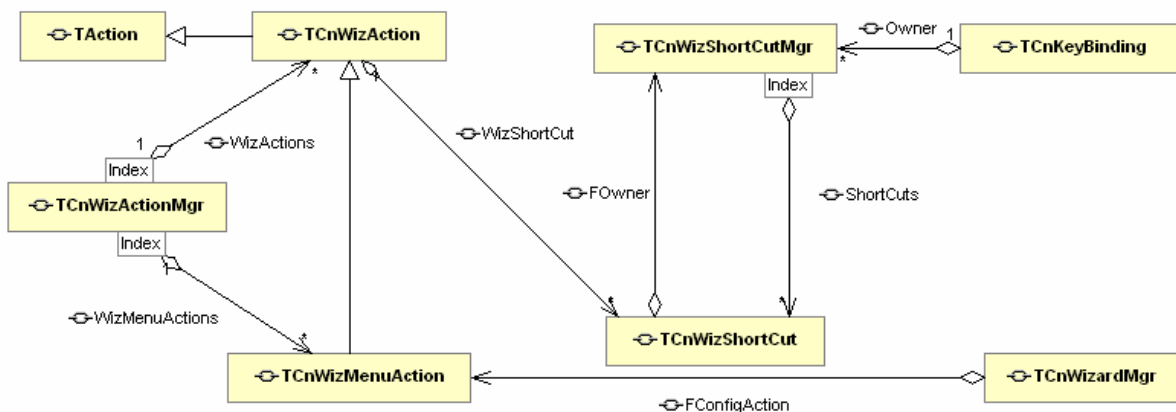


图四 专家包基础类派生关系图

除了专家基础类以外，框架还要实现其它一些辅助功能：

- 快捷键注册和管理部分，由 TCnWizShortCut、TCnWizShortCutMgr、TCnKeyBinding 三个类组成，负责向 IDE 中注册快捷键绑定、维护快捷键列表。TCnWizShortCut 为简单的快捷键、响应事件封装类，TCnWizShortCutMgr 为快捷键列表管理器，维护一个 TCnWizShortCut 对象列表。TCnKeyBinding 为实现了 IOTAKeyboardBinding 接口的类，TCnWizShortCutMgr 通过使用它来向 IDE 中注册快捷键。
- Action、菜单注册和管理部分，由 TCnWizAction、TCnWizMenuAction、TCnWizActionMgr 组成，负责创建和维护注册到 IDE 中的 Action、菜单项列表。TCnWizAction 为 TAction 子类，通过创建 TCnWizShortCut 对象实现在 IDE 中注册快捷键的功能。TCnWizMenuAction 在 TCnWizAction 的基础上增加了菜单项，对应于专家基础类的 TCnMenuWizard。TCnWizActionMgr 管理器用于维护专家的 Action 列表。
- TCnWizardMgr 专家管理器，用于维护专家列表，并完成一些公共的功能。包括启动时创建专家、注册代码编辑器、运行其他初始化功能等。

它们之间的关系见如下 UML 类图：



图五 专家包框架管理类图

更多详情，请参考代码包文档中的《CnWizards 概要设计说明书》和《CnWizards 框架详细设计说明书》。

2、CnOTACreators 设计

Delphi 系列 IDE 的 OpenTools API 中使用了 IOTACreators 等接口提供 Repository 的功能以达到在 IDE 中生成源码、窗体、工程等文件的目的。一般的 Repository 专家都需要按不同的具体环境和设置以生成不同内容的文件，为了增加灵活性，CnPack IDE 专家包中设计了一套使用 TCnTemplateParser 类来根据模板生成文件的类体系，称之为 CnOTACreators。简而言之，CnOTACreators 是 CnPack IDE 专家包中为 Repository 系列模板向导专家设计的、能够灵活应用模板来生成单元窗体工程源码的类库。

CnOTACreators 类体系使用了自编的 TCnTemplateParser 类作为其文件生成的工具，模板文件使用自定义的单元文件和窗体文件等。CnOTACreators 体系的核心作用是用

TCnTemplateParser 的 Tag 替换机制，自动实现 IOTACreator 及其子接口的多个复杂函数，而将真正的创建设置集中控制在类的几个替换处理函数中，以简化专家子类的开发。

TCnTemplateParser 类是整个 CnOTACreators 类的模板驱动核心，它类似于用在网页开发中的 TPageProducer 类，用于根据模板进行不同的标签替换，从而产生不同的输出内容。TCnTemplateParser 的 LoadTemplateFile 方法可以加载模板文件。模板文件是普通文本文件，当调用 TCnTemplateParser 的 Content 属性要求它产生页面时，它会从头到尾扫描模板文件，当遇到<#Name>之类的标签的时候，就会触发 OnTag 事件供调用者自定义标签替换内容，从而达到灵活设置其内容的目的。

CnOTACreators 类体系包括以下几个类：

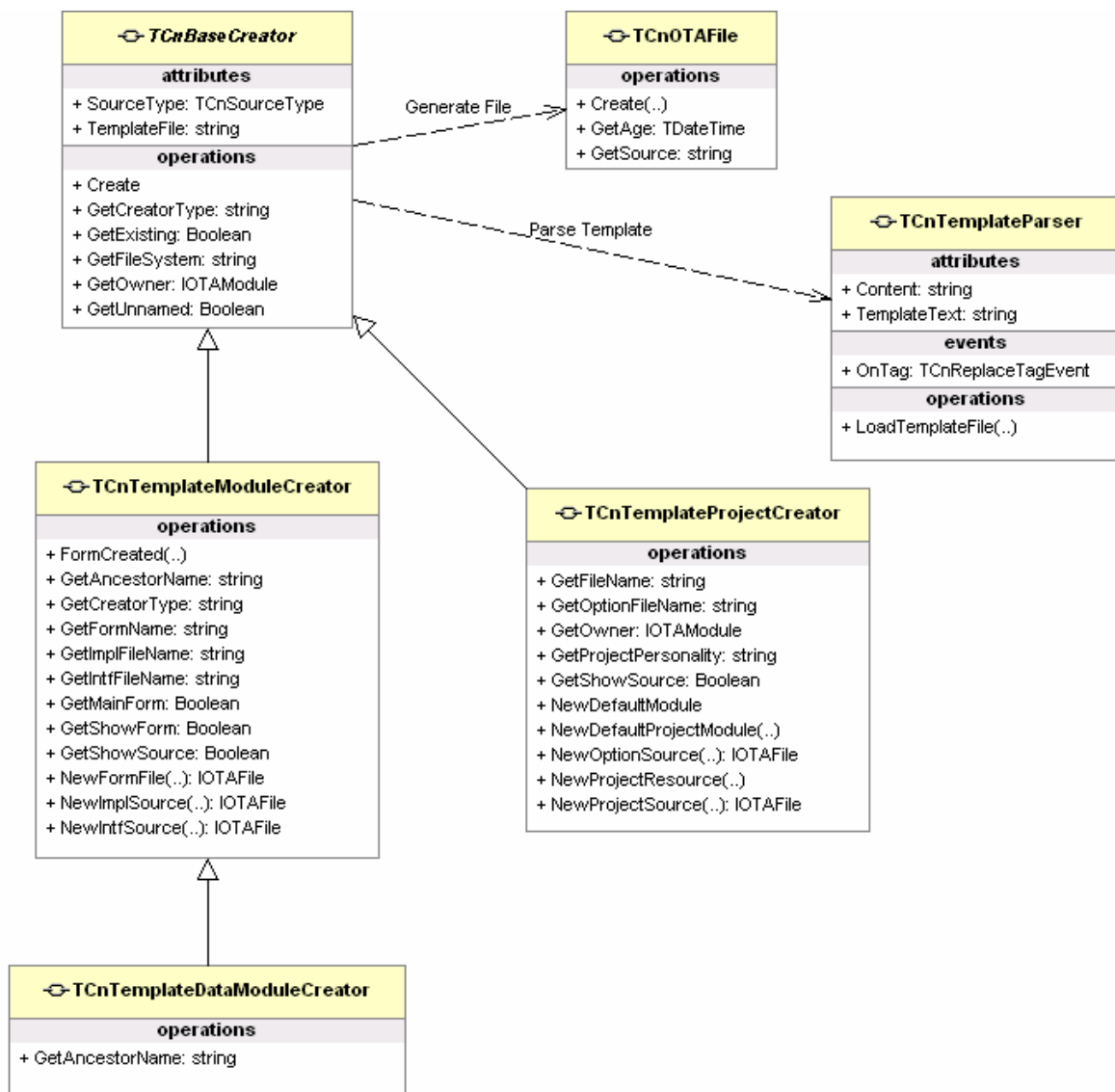
- TCnBaseCreator：它是所有 Creator 的基类，实现了 IOTACreator 接口。TCnBaseCreator 是一个抽象的文件创建接口实现类，不能被直接使用，具体的不同文件类型的创建需要落实到其子类。它内部使用了 TCnTemplateParser 来实现基本的 Tag 替换机制。
- TCnTemplateModuleCreator：它是创建单个单元或包括窗体的单元的类，实现了 IOTAModuleCreator 接口，能够根据模板文件类型设置其生成文件的类型和内容。
- TCnTemplateProjectCreator：它是创建工程的类，继承于 TCnBaseCreator，实现了 IOTAProjectCreator 接口，能够创建一工程。
- TCnTemplateDataModuleCreator：它是创建数据模块的类，继承于 TCnTemplateModuleCreator，简单重载了 IOTAModuleCreator 接口中的 GetAncestorName 返回“DataModule”标识，和 TCnTemplateModuleCreator 大同小异。
- TCnOTAFFile：它是一个简单的 IOTAFile 接口的实现类，能够接受和表示被创建的文件内容。Creators 需要用到它来向 IDE 提供 IOTAFile 接口。

另外还应该有 TCnTemplateProjectGroupCreator，实现 IOTAProjectGroupCreator，用来创建工程组。因应用非常稀少，所以暂不设计。

利用 CnOTACreators 类体系进行 Repository 专家开发也比较方便。一般来说，只需根据需求，选择基类继承一子类，然后重载其模板文件获取和标签替换处理的函数，写上自己的返回内容。执行时通过调用 IDE 服务中的创建模块接口 CreateModule 并传入一 ModuleCreator 实例即可。

在 IDE 最终生成文件内容的时候，需要我们提供一个实现了 IOTAFile 接口的具体类的实例给它，由它调用此接口以获得文件的具体内容。我们使用了一个通用的实现 IOTAFile 接口的类 TCnOTAFFile 来实现此功能。这个类很简单，创建的时候指定其内容字符串，然后在需要的时候，将对象实例返回即可。

它们之间的关系见如下 UML 类图：



图六 CnOTACreators 类图

更多详情，可参考《CnOTACreators 设计说明书》。

3、数据流设计

CnPack IDE 专家包内部存在两种数据流，一是专家包自身的设置信息的保存载入和自定义机制的处理，二是 IDE 内部数据尤其是源码的读出写入转换输出等。

对于第一种数据流，专家包的框架提供了完整的支持。专家包框架封装了注册表中各个专家的信息保存载入的细节，仅通过一 INI 对象传递给专家实例供其 Load/Save 普通的设置信息，免去了专家自身手动保存的麻烦。另外对于部分不适宜保存在注册表中的数据结构如不定长列表等，专家包框架也提供了统一的初始化和用户自定义的手段：初始化的自定义设置文件均保存在 CnPack IDE 专家包发布目录的 Data 目录下，用户修改后的设置文件则被迁移到 Users 目录下。专家包启动载入设置时 User 目录优先查找，Users 目录未找到文件才转至 Data 目录搜索。这些内部机制均封装在 CnPack IDE 专家包的全局设置单元中，不仅提供了统一的存储位置调整手段，也为未来的专家包的升级功能做好了准备：安装新版专家包后，新的初始配置数据只需放入 Data 目录中，用户自定义的设置则在 User 目录中保持不变，如果用户想恢复初始化设置，只需删除

User 目录即可。CnPack IDE 专家包还设计了一单独的“CnWizards 设置导入导出工具”专用于专家包设置的导入导出以及恢复初始设置，读者可参考其代码。

对于 IDE 内部的代码流，Open Tools API 提供了可用的接口来获得其内容，但其接口繁琐不一致，不利于专家代码直接使用。CnPack IDE 专家包设计了一工具类 TCnEditFiler 来实现根据文件名读写 IDE 缓冲区和文件的功能。如果需要读的文件已在 IDE 中打开，则读取内存缓冲区，否则读取磁盘文件，写入时操作类似。同时 TCnEditFiler 还封装了因 BDS 中编辑缓冲区使用 UTF-8 的内码而需要的 ANSI 转换操作，给处理代码的专家类提供了很大的方便。

4、用户界面设计

CnPack IDE 专家包的界面设计有三个基本准则，一、符合 IDE 自身窗体的布局要求，和 IDE 风格保持一致；二、在不同的 DPI 也即屏幕字体大小不同的情况下，要能保证界面显示仍然正常，组件大小调整适当；三、支持多语动态切换。

对于第一点，要求开发人员在代码编写期间注意窗体布局风格即可，具体包括组件间距、TabOrder、距窗体边缘距离等各种细节。

对于第二点，CnPack IDE 专家包设计了 CnFormScaler 组件，在 DPI 发生变化时自动缩放窗体以及其他可视化控件。该组件也被设计为独立于专家包而存在，目前已集成入 CnPack 组件包中，可单独发布使用。该组件内部通过挂接窗体的 WndProc 来实现对窗体位置尺寸变化的响应，在不同的 DPI 下对所有组件实现缩放并调整 Constraint 属性。CnPack IDE 专家包的所有普通窗体都继承于一使用了 CnFormScaler 的父类窗体，因此可实现窗体尺寸、控件尺寸等缩放的统一控制。

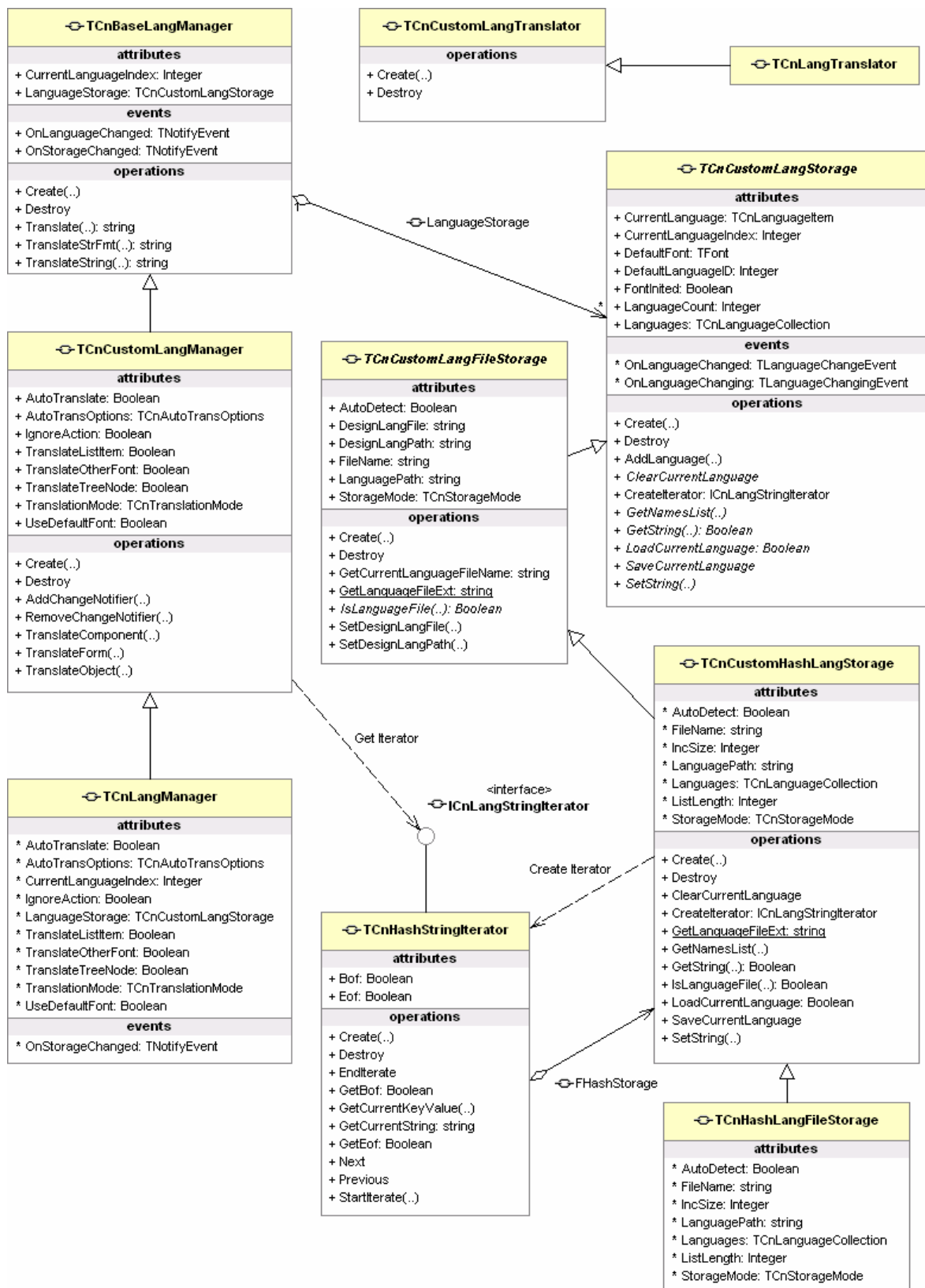
对于第三点多语的支持，CnPack IDE 专家包同样是在上述的父类窗体中加入了多语组件的翻译动作，从基类上实现了子窗体界面的自动翻译，并且提供重载函数供语言变化时调用，以控制内部语言切换时所需要的自定义动作。至于多语组件的设计，则又是另外一部分相当大的篇幅内容，这里只简略描述一下：CnPack 多语组件包能够替现有的 Delphi 工程加入多语翻译的能力。一个工程所具有的翻译能力应该包括三部分：**语言条目的存储能力、运行期切换语言的能力和设
计期支持的能力**。为工程提供这三种功能的组件分别是多语存储组件、多语管理器和多语翻译器。它们基本上能完成所提到的所有功能。

如下是 CnPack 多语组件包中的三类组件列表：

| 功能需求 | 类系列 | 基类 | 注册组件类名 |
|--------------|--------------------|-------------------------|-----------------------------|
| 多语条目 存储功能 | 多语存储 组件系列 | TCnCustomLangStorage | TCnHashLangFileStorage 等 |
| 多语翻译 功能 | 多语管理 器系列 | TCnBaseLangManager | TCnLangManager |
| 设计期支 持功能 | 多语设计 期翻译器 系列 | TCnCustomLangTranslator | TCnLangTranslator |

其中，TCnCustomLangStorage 及其子类用来存储多种语言的翻译条目，而 TCnLangManager 连接了 TCnCustomLangStorage 组件，可通过设置当前语言来激活 TCnCustomLangStorage 中的某种语言并从中获得翻译条目，然后 TCnLangManager 负责使用翻译条目来更新界面。TCnLangTranslator 则是设计期的支持，能收集窗体的待翻译字符串并供手工翻译之用。

它们的 UML 类图如下图所示：



图七 CnPack 多语组件包主要类的类关系图

更多详情，可参考《CnPack 多语组件包概要设计说明书》。

5、软件运行平台设计

CnPack IDE 专家包运行于 32 位 x86 平台的 Windows 95/98/NT/2000/2003/XP 上，此部分无特殊设计内容。

6、部署与安装设计

CnPack IDE 专家包的安装程序采用 NSIS 2.0 脚本打包制作，脚本中提供简体中文、繁体中文和英文三种安装界面，通过读注册表自动检测已安装的 IDE 类型，允许可定制安装，用户可自由选择安装所有支持的 IDE 中的一个或多个专家包。

六、依赖的其它软件

CnPack IDE 专家包运行期依赖于 Delphi 5 6 7 或 C++ Builder 5 6 或 Borland Developer Studio 2005 2006 的其中一种或多种开发环境。

七、项目的组织和管理

CnPack 项目组属于网络协作模式，人员居住分散，而且都是业余时间开发，工作进度不易保证，因而进度控制较松。同时成员数目众多，分工明确，也为项目的顺利开展打好了基础。整个项目开发过程分为若干个阶段，每个阶段又由管理员设计若干个相对独立的任务，而开发员则选择自己有能力完成的任务进行开发，最终完成整个项目。

CnPack IDE 专家包项目成员包括以下职位，部分职位可兼任：

- 项目管理员：负责整个 CnPack IDE 专家包项目的管理与协调工作，同时也是主力开发人员。
- 核心组成员：对 CnPack 项目有较大贡献、比较深入地了解 CnPack 项目框架结构与运作方式的开发人员，有直接的代码提交权限。
- 普通开发员：参与 CnPack IDE 专家包代码的编写者，一般无直接的代码提交权限，而是将代码提交给管理员，验证后由管理员提交以保证质量。
- 代码移植员：将志愿者捐献的代码（包括其他语种）按 CnPack 的标准修改或改写成符合 CnPack 规范的代码的人员。
- 测试员：负责平时测试 CnPack IDE 专家包自动构建版的相关功能的人员，包括部分热心用户。
- 文档编写员：负责编写需求、设计、测试、帮助文档的人员，部分由相应的开发人员兼任。
- 论坛管理员：CnPack 网站论坛版主，负责回应用户的各种疑难问题和建议。
- QQ 群管理员：负责 CnPack 四个 QQ 群的交流与用户管理。

项目成员间的交流很重要，CnPack 项目组的交流途径包括邮件列表、论坛、QQ 群、MSN 等即时通讯工具。

整个开发的过程是螺旋上升的，可序次细化为以下几步：

1、内容拟定

定义项目子功能的详细描述，这一部分工作在专家包设计期进行，主要由核心组的设计人员负责人完成，在开发过程中可适当进行修改。这一阶段的工作以编写文档为主。

2、 结构设计

设计专家包和子功能的框架结构，包含类层次、发布属性、事件、方法、单元划分等具体内容，以及主要标识符的定义。结构设计以文档结合代码编写进行，由核心组人员或该功能负责人完成。结构设计完成后应发到管理员处进行统一整理。

3、 公共代码编写

这一步为可选步骤，部分项目在创建主代码前需要编写一些预备代码，如通用过程库、常量定义等，由开发人员完成

4、 主代码创建

开发人员最重要的工作内容，包括开发人员调试和单元、功能测试过程。

5、 核心组开发评测

主代码完成后，发送到核心组或管理员处对代码进行检查、评测、整理和提交。

6、 开发组内部测试。

开发组测试人员对已完成的代码在不同的操作系统、开发工具中进行功能、稳定性、兼容性测试。

7、 帮助、文档、例子编写。

由开发组和文档人员根据项目功能、测试结果编写帮助文件、说明文档及演示代码。

8、 集成到开发包中发布，外部测试。

网站维护和项目集成人员定期对已完成的项目集中发布，供下载和外部测试，这一轮的“开发”过程完成。

9、 根据测试结果修正、改进。

项目的“维护”和后期改进工作。当专家包项目做大的升级时（主版本变化），再回到前面几步重新进行。

八、 项目管理工具

CnPack 项目使用 CVS 作为源码管理工具，管理员以及核心组成员拥有 CVS 帐户的提交权限。其余开发人员提交的代码必须由管理员或核心组成员验证后手工提交。CVS 库对外界开放匿名只读的权限，CVSROOT 为

```
:pserver:anoncvs@www.cnpack.org:/var/cvshome/cnpack
```

模块名为 cnpack 和 cnwizards，两个模块需要 Update 到本地同一目录才可编译。详情可参考《CVS 使用说明》。

CnPack 项目使用和 CVS 配套的 CVSTrac 作为任务管理及错误跟踪系统。CVSTrac 是国外 D. Richard Hipp 为 CVS 开发的基于 Web 的任务管理及错误跟踪系统，CnPack 开发组将其移植到了 Windows 平台上并加入了多语言支持，这也是 CnPack 项目发展过程中的副产品之一。

CVSTrac 使用任务单驱动开发的方式，类似于 ClearDDTS 与 ClearCase 的协作模式。当用户报告了一个错误，经管理员或核心组成员确认后，管理员或核心组成员可在 CVSTrac 中创建一任务单，任务单记录了问题的来源、产品版本、问题描述等。然后由管理员或其他成员研究后送交至某开发人员进行修改，修改成功后提交代码，测试成功后将此任务单标记为测试状态，发布里程碑后则改为完成状态。这是一任务单的完整生命流程。CVS 的提交记录可在任务单中记录显示，方便回溯跟踪。

对于新特性的开发任务和文档的任务，流程和上面类似，只是任务单的类型为新项开发、文档等类型。

开发人员可通过三种方法参与开发：第一种是“分配式”，小组负责人根据任务难度及小组成

员具体情况，将新任务指定给一个成员并与之联系，如果该成员接受了这项任务，负责人就会在 CVSTrac 的任务单中相应任务后加入此开发员的名字，表示该成员正在为这个任务工作。开发员在任务接近结束时应通知设计人员，并最终提交任务代码。

第二种称为“认领”，也就是开发员看到新任务描述后，可向负责人发信联系，表示自己可以胜任此工作，负责人收到信后更新任务表，将该任务交由该成员负责。

第三种参与开发的方法称为“默许协作式”，在这种方式中，开发员可以不必事先通知设计人员就根据文档着手开发，在开发接近结束时再通知设计人员，并最终提交任务代码。

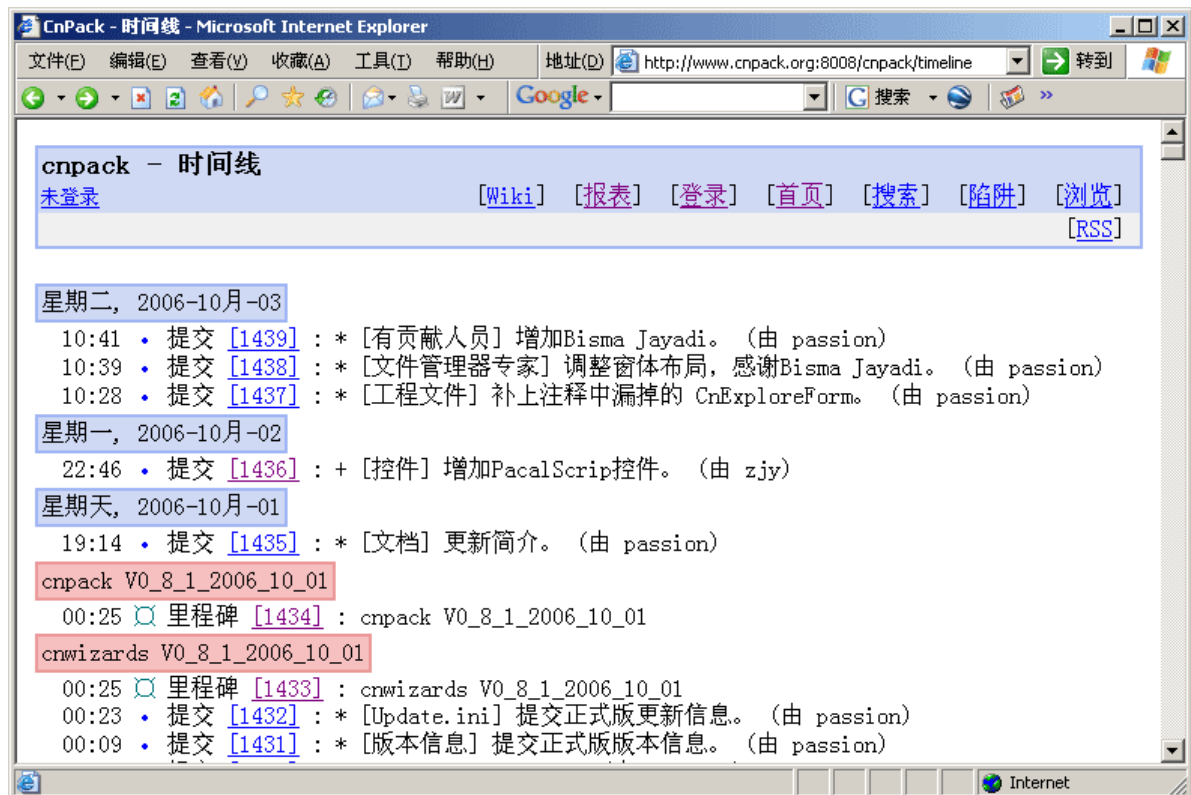
CnPack 项目组的 CVSTrac 地址为：

<http://www.cnpack.org:8008/cnpack/index>

其中记录了 CVS 提交记录的时间线地址为：

<http://www.cnpack.org:8008/cnpack/timeline>

CVS 时间线如图所示：



图八 CVSTrac 中显示的 CnPack 的 CVS 时间线

CnPack IDE 专家包还启用了每日自动构建的项目发布方式，项目采用 want 来做自动构建，当 CVS 发生变化时，由项目构建负责人（一般由管理员兼任）运行 CnPack IDE 专家包的 want 脚本来构建并发布一每日构建版。每日构建有助于 CVS 提交质量的控制，为测试提供了方便，并为正式版的发布节省了许多人力物力，在 CnPack IDE 专家包的项目工作中发挥着重要的作用。

CnPack IDE 专家包的每日构建版的下载地址为：

<http://www.cnpack.org/downbuilds.php>

第四部分 用户使用手册

摘要

本“使用手册”部分是CnPack IDE 专家包官方帮助文件的内容摘要。CnPack IDE 专家包目前拥有完整的帮助文件，包括简体中文、繁体中文和英文三种语言的版本。由于篇幅量巨大的关系，无法在此文中全部列出，这里只将关键内容摘录部分在此。欲查看其完整内容，请参考cnwizards源码中的Help目录中的相应HTM文件，或直接查看安装后的帮助文件CnWizards_CHS.chm（英文用户请查看CnWizards_ENU.chm）。

一、运行环境/系统使用说明

CnPack IDE 专家包的运行环境：

硬件：32 位 x86 平台

操作系统：Windows 95/98/NT/2000/2003/XP

软件环境：必须安装了 Delphi 5 6 7 或 C++Builder 5 6 或 Borland Developer Studio 2005 2006 的其中一种或多种开发环境。

二、软件安装：

系统安装说明：

1、从二进制代码包的安装

直接在资源管理器中双击运行安装包启动 CnPack IDE 专家包安装程序即可。安装程序会提示选择安装语言种类，并根据 IDE 的安装情况安装相应版本的专家。

2、从源文件安装

手动：启动某 IDE，打开专家的对应版本的工程文件，编译出一 DLL，然后在注册表中添加相应的专家键值。重启此 IDE 后，CnPack IDE 专家包便安装到此 IDE 中。

自动：CnPack IDE 专家包使用了 want 来进行自动构建的工作。完整的 want 的 xml 脚本包括开发期的 CVS 更新，编译各个 IDE 的专家 DLL，编译独立工具，编译资源文件、编译帮助文件，使用 NSIS 进行打包，FTP 上传，CVS 提交新版本信息等。其中许多功能可有选择性地执行。使用时只需运行 cnwizards\Make 目录下的 want.exe 并根据其提示信息进行操作即可。

不过，上述自动构建对机器的软件环境有比较大的要求，需要安装的工具如下：

- CVS 客户端，如 WinCVS 等，用来联网获取最新的 CVS 代码和提交新版本信息。
- Delphi 5，编译 Delphi 5 下的专家包和资源 DLL 以及其他独立工具。
- Delphi 6，编译 Delphi 6 下的专家包。
- Delphi 7，编译 Delphi 7 下的专家包。
- Delphi 7.1，编译 Delphi 8 下的专家包。
- C++Builder 5，编译 C++Builder 5 下的专家包。
- C++Builder 6，编译 C++Builder 6 下的专家包。

- BDS 2005，编译 BDS 2005 下的专家包。
- BDS 2006，编译 BDS 2006 下的专家包。
- Microsoft HTML Help Workshop，编译帮助文件。
- NSIS2.0 正式版，压缩生成安装包。

注：以上编译器开发工具均可使用免费的个人版系列进行编译，无需企业版或架构版等。

三、软件使用

1、全局选项

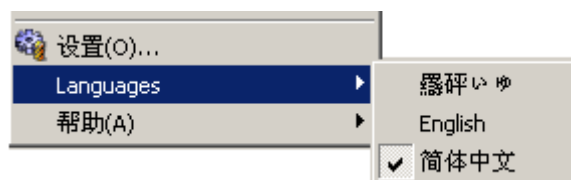
CnPack IDE 专家包的全局选项在 CnPack 菜单项的“设置”菜单项中，点击此菜单项会弹出专家管理器设置窗口，它包括各个子专家的基本信息和设置、以及属性编辑器和其他全局设置选项等。如图：



图九 CnPack IDE 专家包全局设置窗口

2、语言选项

CnPack IDE 专家包的正式版本支持简体中文、繁体中文和英语三种界面语言，可以在运行期动态切换。使用时只需点击 CnPack 菜单项，在 Languages 中选择需要的语言即可。切换语言后，所有有关于 CnPack IDE 专家包的运行期内容都会切换至此种语言，调出的帮助内容也不例外。如图：



图十 CnPack IDE 专家包语言切换菜单

3、优化选项

用户在使用中如不需要某些专家，可禁用他们。禁用有两种，一种是在专家设置对话框中选中不需要的专家，不勾选“允许该专家为活动状态”的复选框，这样此专家的 Active 属性为 False，许多功能便不起作用；更彻底的是在高级设置中让不需要的专家不再创建，这样专家类连创建实例的机会都没有，除了占用内存之外，几乎不会产生任何运行期影响。这是最彻底的禁用办法，需要 IDE 重新启动。

其他的优化选项，请参考帮助文件中每个专家的详细功能说明。

4、调试选项

CnPack IDE 专家包的源码中使用了 CnPack 开发组的 CnDebugger 工具来记录输出调试信息，但在正式版本中不提供调试信息功能（使用编译开关关闭了）。CnPack 网站上有另外的打开了编译开关后构建的调试版供下载。当用户下载调试版本安装后，就可以通过启动 CnPack IDE 专家包附带的 CnDebugViewer 工具来查看 CnPack IDE 专家包输出的调试信息了。此功能给 CnPack IDE 专家包的排错带来了很大的方便。输出及查看调试信息的详情可参考源码中附带的文档《CnWizards 调试说明书》。

四、疑难问题解答

本疑难解答来自于CnPack论坛上的“常见问题解答”，其中包括部分新功能需求以及开发计划的解释，部分内容较为专业，适合于中高级用户阅读。

1、代码格式化工具的问题。

CnPack IDE 专家包目前不包含代码格式化工具，而此工具是许多用户都需要的。目前网上比较好的开源代码整理工具主要有 DelForEx 和 Jedi Code Formatter。前者历史悠久，使用也很广泛。后者则是基于语法分析的整理工具，由 Jedi 组开发。由于 DelForEx 本身存在不少缺陷，而 JCF 使用 MPL 协议不太方便整合，所以 CnPack 也有自己的 CnCodeFormatter 开发计划。该工具由开发组“古月春秋”捉刀，目前已经有可运行的雏形了。但是由于代码整理涉及到一些比较复杂的算法和技术，特别是对条件编译的处理比较麻烦，加之负责该项目的成员精力有限，所以迟迟未对外发布，需要耐心等待。

2、代码折叠功能。

BDS 2005 发布后，很多用户提到了在低版本IDE的编辑器中实现代码折叠的这个需求。代码折叠编辑技术本身并不复杂，但是由于代码编辑器作为 IDE 最重要的一个组成部分，涉及到后台和底层很多东西。要在现有的 IDE 代码编辑器中扩展折叠功能，需要做大量的测试和复杂的技术。如果用一个新的支持折叠的编辑器来取代现有的，又很难与代码提示、编译等现有功能整合。所以这个功能目前还只是可研阶段，暂时没有实现的计划。

3、代码结构高亮功能。

即类似 CodeRush 的 begin..end 连线功能。这个已经在开发计划中了，近期的版本支持了括号高亮功能，就是该功能的前奏。要实现这个功能并不复杂，Dragon P.C 朋友已经捐助了部分实现代码，开发组现在关注的是如何提高性能，特别是在超万行的大单元中，要保证处理性能并不容易。当前的括号高亮实现是经过多次优化的，即使是在三万行以上的单元中，仍然不会有性能损失。而 Castalia 提供的括号高亮在这样的环境下明显变慢。一旦解决性能问题，这个功能将在最新发布版本中提供，请关注更新信息，欢迎参与测试。

4、用空格完成代码模板的功能。

很多用惯 CodeRush 的朋友，建议我们增加类似的功能。不过在 CnPack IDE 专家包中，我们已经提供了代码输入助手这个特色功能，将不打算再增加使用空格来完成的代码模板了。两者各有所长，空格完成的好处是速度快，而代码助手虽然需要多敲几下键，但是可以不用记忆模板名称，而且可以自由选择列表项，避免误输入。这是使用习惯问题，请大家理解。

5、代码输入助手不能自动弹出的问题。

如果用户使用的是未安装升级包的 BDS 2005，请安装最新的升级包。由于原始的 BDS 2005 内部存在缺陷，会导致输入助手取不到必须的服务接口而无法正常工作。另外，还有一种情况是如果用户开启了某些输入法，在输入点号后，IDE 自己的自动完成将无法弹出，助手也不会出来。这是因为 IDE 自带的代码完成功能不支持这种情况，用户可以在 IDE 编辑器设置中将 Code Completion 功能禁用掉，这样输入助手会自动取代它，并在输入点号后弹出符号列表。

五、成员组成及联络方式

由于CnPack开发组成员众多，目前有200多人，因此无法在此一一列出。这里只列出目前负责项目管理和发布等工作的两名管理员：

刘啸，CnPack现任管理员，本次开源软件竞赛参赛负责人。

联系方式：137xxxx0220，master@cnpack.org。

周劲羽，CnPack项目创始人，现任管理员。

联系方式：zjy@cnpack.org

CnPack开发组其他成员分布在中国各地、跨越海峡两岸，成员众多，分工包括项目管理、代码开发、代码移植、项目构建、测试、文档编写、论坛管理、QQ群管理等，在此不一一列举。具体开发人员和贡献细节，请参考帮助文件中的《有贡献人员名单》。

第五部分 测试说明书

一、测试目的

CnPack IDE 专家包的测试目的是追求各项专家功能的正确性、完备性和高效性，尽量排除专家包的各种错误和不足，为用户提供稳定的发布版本，也为开发组进一步的宣传、发展打下坚实的基础。

二、测试计划

CnPack IDE 专家包的测试内容包括两部分：面向用户操作的功能点和提供公共服务的组件和运行库。

CnPack IDE 专家包的测试过程也是一个随着开发而螺旋上升的过程，并且新增功能和测试修正同步进行。对于专家包面向用户的功能，其集成测试计划以网站上发布的全功能每日构建版为驱动的基础，当专家包新增了某一功能并且集成到某一全功能构建版中后，开发组的测试人员以及部分热心用户会下载安装测试其新增功能（注意：在此全功能构建版之前，可能存在开发者提交的中间过渡版本，通常为不完备或未经单元测试的临时版本，这类型的专家包构建版不作为集成测试的基础），并提供反馈给开发组管理员或开发人员。

一般专家在发布正式版前会发布系列 Preview 版，此时要求测试人员比较全面地测试各项功能，测试通过后才允许发布正式版。

三、测试环境

CnPack IDE 专家包的测试环境大体同其运行环境，但要求更全面：

硬件：32 位 x86 平台

操作系统：Windows 95/98/NT/2000/2003/XP

操作系统语言：简体中文、繁体中文、英文

IDE 环境：Delphi 5 6 7 或 C++ Builder 5 6 或 Borland Developer Studio 2005 2006。

其他专家环境：GExperts、CodeRush、DelFor、Castalia，用来测试和 CnPack IDE 专家包是否冲突等问题。

需要注意的是开发组的测试人员的人力物力有限，不可能覆盖到如此众多的操作系统、IDE 和第三方专家的全部组合，仅仅在主流配置上做测试验证。另外，由于测试人员时间空间上的分散性，测试内容手段的交流和结果报告等往往是口头或其他即时交流工具的形式，一般不形成正式的测试文档和测试报告。源码包的文档中有一篇《CnWizards 部分功能测试报告》，其形式也较不正式，覆盖面也不广，仅仅是一篇针对部分新功能的测试报告，读者可以参考。

四、测试方法以及测试用例

CnPack IDE 专家包中面向用户操作的功能点，其测试方法以手工的黑盒集成测试为主，测试人员包括开发员、开发组测试人员和部分热心用户。至于提供公共服务的组件和运行库等内容，则通过编写测试用例运行来验证，一般是白盒单元测试，测试人员主要是开发人员本身，也有部

分开发组测试人员。

面向用户操作的功能点的测试用例基本上全部是按各个子专家划分的，如下（列表中是测试用例和应该达到的目标）：

- 内部专家：
 - “关于”子菜单：能正确显示关于窗口，正确调用链接和外部工具等内容。
 - “Languages”子菜单：能正确切换界面语言。
 - “设置”菜单：能正确弹出设置对话框并对各专家、属性编辑器和全局选项进行设置，设置内容生效。

- 独立功能专家：
 - 面向窗体设计：
 - ◆ 窗体设计专家：各项子功能能正确设置组件尺寸、位置、对齐等特性。
 - ◆ 组件选择器：能按照设定的条件正确选择组件。
 - ◆ TabOrder 专家：能按照设定的规则自动设置各控件的 TabOrder。
 - ◆ 属性修改器：能正确按指定规则修改组件的属性。
 - 面向代码处理：
 - ◆ 源码统计专家：能正确地统计源码的行数，正确地汇总。
 - ◆ 代码编辑器专家：各项子功能能正确地处理各个代码片断。
 - ◆ 书签管理专家：能正确地添加删除和管理源码中的书签。
 - ◆ 源码模板专家：能正确地根据代码模板生成代码，并正确地管理代码模板。
 - ◆ 源码比较：能正确地显示和比较磁盘文件和内存编辑器等内容，正确地合并。
 - ◆ 注释清除：能正确地按自定义规则清除掉源码中的注释而不影响原有代码。
 - ◆ 代码格式输出：能正确地将代码按高亮显示方式输出成 HTML 或 RTF 格式的代码文字。
 - ◆ 过程单元列表：能正确地列出当前单元的过程并顺利搜索跳转。
 - ◆ MessageBox 设计：能正确地根据以可视化的方式设计的弹出消息窗口生成代码。
 - 面向文件工程：
 - ◆ 工程扩展工具：各项子功能能正确地打开相应目录、弹出扩展单元窗体列表、正确地进行备份、清除临时文件、构建工程目录等。
 - ◆ 打开历史文件：能记录曾经打开的文件并重新打开，并正确地管理快照。
 - ◆ 批量文件替换：能正确地批量替换文件中的内容。
 - ◆ MSDN 集成：能正确地在 MSDN 中和网上进行帮助搜索。
 - ◆ 引用单元清理：能正确地清理大型工程中各单元中不需要但 uses 了的单元。
 - ◆ 文件管理器：能正确显示资源管理器窗体。
 - 面向设置备份：
 - ◆ IDE 设置备份/恢复工具：能正确地备份和恢复 IDE 自身的设置。
 - ◆ CnWizards 设置导入/导出工具：能正确地备份和恢复 CnWizards 的设置。

- IDE 功能扩展专家：
 - 代码输入助手：能正确弹出、上下文分析准确、速度快。
 - 代码编辑器扩展：能够正确显示代码编辑器工具栏、正确显示行号、高亮配对显示正确、浮动按钮工作顺利等。

-
- 窗体设计器扩展：正确显示设计期窗体上的工具面板。
 - 组件面板扩展：正确切换组件面板的多行显示、准确搜索组件。
 - 对象查看器扩展：准确地以黑体显示修改过的属性。
 - 窗口置顶折叠扩展：准确地按规则为 IDE 的窗口增加置顶和折叠的功能。
 - CPU 调试窗口扩展：CPU 调试窗口复制汇编代码功能正常工作。

- Repository 专家：

- DUnit 用例生成：生成 DUnit 的用例单元或工程，在安装了 DUnit 的情况下要求生成的工程或单元能顺利编译通过。
- CnMemProf 工程生成：生成 CnMemProf 工程，在安装了 CnMemProf 内存管理单元的情况下要求生成的工程能顺利编译通过。
- INI 读写单元生成：按 INI 内容生成自动读写 INI 内容的对象类单元，要求生成的 Object Pascal 单元或 C++ 单元能顺利编译通过。

- 属性编辑器：

- 集合属性编辑器：能正确地显示集合属性。
- 布尔属性编辑器：能正确地显示布尔属性。
- 字符串属性编辑器：能正确地编辑字符串属性。
- 工具提示属性编辑器：能正确地编辑工具提示属性。
- 字符串列表属性编辑器：能正确地编辑字符串列表属性。
- 文件名属性编辑器：能正确地显示和编辑文件名属性。
- Constraint 属性编辑器：能正确地编辑 Constraint 属性。
- 滚动条属性编辑器：能正确地编辑滚动条属性。
- 字体属性编辑器：能正确地显示字体属性。
- Align 属性编辑器：能正确地显示 Align 属性。
- 组件名称属性编辑器：能正确地利用组件前缀专家来编辑组件名称。

提供公共服务的组件和运行库的测试用例包括以下部分：

- 专家包框架和公共库：此部分已集成入各个子专家的功能测试。
- 多语测试用例：完整地测试多语组件在设计期生成并编辑语言文件的能力，测试多语切换静态窗体和动态创建窗体时的窗体翻译能力以及资源字符串的翻译能力。
- CnDebugger 测试用例：完整地利用 CnDebug 的各个接口输出各种类型的调试信息，验证 CnDebugViewer 能否完整接收并分类处理。
- 其他测试用例：如水波纹图像效果组件的测试用例（此组件用在专家包的关于窗口上）。

上述用例的源码，可参考 CVS 代码包中的 cnpack 模块部分。

五、测试结果

测试通过。