

CnPack 开发方案

V0.8.3.1231 beta3

作者：周劲羽(zjy@cnpack.org)

2001.12

一 关于 CnPack 开发包

一、开发 CnPack 的目的。

基于在大量的 Delphi 第三方组件代码充斥网络的今天，却仅有少量国产组件零星分布这样一个残酷的事实，我们决定以自己微薄的力量架构属于中国人自己的重量级免费第三方开发包。

开发包定名为“CnPack For Delphi 开发包”，简称 CnPack，即中国人自己的免费第三方开发包，开发包的设计语种为简体中文，正式版发布时可能会推出英文版和繁体中文版，目前计划支持 Delphi5、6(不排除将来支持 BCB 和 Kylix 的可能)。

开发 CnPack 的最终目的是为了提高中国 Delphi 程序员的开发效率，减少无谓的重复开发，推动中国软件业的发展。

二、CnPack 的版权及发布形式。

CnPack 开发包整体的著作权与版权属于 CnPack 开发组所有，开发包内包含的代码、单元、文档及相关内容的著作权与版权属于编写者与 CnPack 开发组共同所有。

CnPack 开发包以开放源码 (Open Source) 的形式发布，遵守 LGPL 条约，受 LGPL 条约的保护。

更详细的内容参见“版权声明”部分。

三、CnPack 开发内容。

整个 CnPack 开发包由以下几个部分组成：

(以下内容仅供参考，最终的开发内容将由开发成员、开发环境等因素决定)

- 1、工具组件包。
- 2、用户界面控件包。
- 3、符合中国人使用习惯的数据库、报表组件包。
- 4、网络通讯组件包
- 5、本地化、扩展的属性、组件编辑器。
- 6、IDE 扩展专家工具包。
- 7、CnVCS 版本控制系统。
- 8、本地化、功能强大的数据桌面、资源编辑器、组件编写工具等辅助开发工具。
- 9、大量可重用的窗体模板库。

详细说明参见“开发包内容”部分。

四、CnPack 开发网站。

CnPack 开发网站“中文 VCL 开发中心”为 CnPack 的开发及技术支持网站，开发组所有工作成果在此发布并接受自愿者报名以及为用户提供技术支持。

最终的开发包中将包含一个 IDE 扩展专家工具，将用户的 Delphi 开发环境与 CnPack 开发网站紧密集成，提供类似 Delphi.NET.CN 的功能。

详细说明参见“开发网站建设”部分。

五、CnPack 的开发形式及开发时间。

CnPack 采用多人合作的方式开发，整个开发工作由 CnPack 开发组 (以下简称开发组) 完成。

开发组由热衷于自由软件事业的自愿者自发组成。开发组根据工作内容的不同又分为开发包设计组、代码编写组、辅助开

发组及网站维护组，各组承担开发包中相应工作。详细说明参见“开发人员组织方案”部分。

开发包的开发时间定为 2002 年度，计划在第一季度内建构初具规模的开发网站、发布第一批测试代码，上半年内发布第一个较完整的测试包，2002 年底或 2003 年上半年发布 CnPack V1.0 正式版。

六、CnPack 设计规范。

为了保证最大的代码重用性及开发包的质量，开发必须按一定的规范来进行。

主要包括代码编写、标识符命名、版本定义、界面设计、帮助文件及相关文档编写等开发规范。

详见“开发包设计规范”部分内容。

七、CnPack 开发方法。

开发包包含的每一个单元都要经过代码编写、设计组修订、内部测试、文档整理、外部测试、错误修正等步骤，以保证开发内容的整体一致性及开发质量。

在多人开发中，开发方法的设计尤为重要，“开发包开发方法”部分内容详细定义了了在多人协同开发环境下开发包的具体实现方法。

二 版权声明

一、CnPack 的版权。

在下面的描述中，版权一词包括了软件著作权、署名权、版权，而作品则指开发包中每一个单独的代码、单元、软件、文档及相关内容，脱离了 CnPack 的作品不在此限制内。

- 1、CnPack 开发包整体的版权属于 CnPack 开发组所有。
- 2、CnPack 中每一个作品的版权属于其作者与 CnPack 开发组共同所有。
- 3、多人同时开发同一作品时，作品的版权属于所有作者及 CnPack 开发组共同所有。
- 4、同一作品由不同作者先后开发、修改时，CnPack 开发组拥有作品的版权，而每一个作者拥有其工作部分成果的版权。

二、最终用户许可协议。

CnPack 开发包以开放源码(Open Source)的形式发布，遵守 GNU LGPL(GNU Library or "Lesser" Public License) 条约，受 GNU LGPL 条约的保护。《附三 GNU Lesser Public License》有对该条约的详细描述，下面是普通用户需要注意的几点：

- 1、无论用户从何处以何种形式获得 CnPack，都是正版用户。
- 2、用户可以将 CnPack 中的组件、工具、源码用于学习用途和免费软件中，但未经开发组授权许可不得将它用于共享软件、商业软件及其它盈利性场合，CnPack 开发组保留对这一声明的解释权和修改权。
- 3、在保证 CnPack 整个软件的完整性的前提下，任何人可以以任何形式对 CnPack 进行拷贝、传播，但不得对 CnPack 进行收费。为传播 CnPack 而收取的成本费和服务费或者技术支持费不在此限制内。
- 4、CnPack 是没有使用担保的，对于因为使用 CnPack 而造成的任何损失由用户自己承担，开发组没有提供技术支持的义务。

三、对 CnPack 进行修改。

打算对 CnPack 进行改进和重新发布的实体请仔细阅读《附三 GNU Lesser Public License》。在该协议规定下，对 CnPack 的修改请遵循以下几点：

- 1、用户可以根据自身需要对 CnPack 进行改进，但改进后的作品也必须免费发布。
- 2、任何人不得将 CnPack 的整体或部分据为己有，不得将 CnPack 或其衍生作品以商业名义重新发布。

三 开发包内容

一、开发包内容规划。

开发包最终内容的制定由开发组与关注 CnPack 开发的热心朋友共同决定。

在最初设计阶段，计划包含以下几大部分。

二、不可视工具组件包。

包含若干从 TComponent 派生而来的不可视组件，用于封装操作系统底层功能调用、扩展应用程序功能以及提供一些有助于提高开发效率的封装代码。

已计划的组件有：

- 1、使用 INI 文件支持多语种的组件（采用类似于 RxLib 中 FormStorage 的操作方式）。
- 2、使用方便的文件打包、恢复组件。
- 3、可选择使用 Timer、MIDI 序列产生器或线程方式获得不同精度的定时器。
- 4、根据 CPU、内存、显存等参数给计算机分级的组件，以使程序在不同运行环境下使用不同品质的处理方式。
- 5、一个允许用键盘模拟鼠标操作的组件。
- 6、同时支持 Win9X/NT/2000 的 32 位屏幕鼠标取词组件。
- 7、农历、公历转换、日出日落时间计算、日蚀、月蚀推算及其它提供天文运算功能的组件。
- 8、应用程序病毒免疫组件。

三、用户界面控件包。

设计标准：消耗系统资源少、界面美观、使用方便、显示速度快。

已计划的内容有：

1、一套综合了 FastLib、Graphic32、ExtraPack、FlatSytyle 等图像处理包优点的快速图像处理库，提供接近 PhotoShop 的图像处理效果。

- 2、改进的平滑字体显示库，更快的显示速度、支持更多的特效、字体自动匹配及提供封装的平滑字体对话框等。
- 3、一组用于制作“操作向导”窗体界面的控件，包括带动态切换效果可在继承窗体中使用的多页控件、操作向导树等。
- 4、一组基于快速图像处理库和平滑字体库的界面控件。
- 5、一组方便使用的“Dock”风格菜单、工具条控件。
- 6、一组伪可视组件，如特效 Hint、窗体桌布、为普通控件增加移入时显示动态粒子效果的组件等。
- 7、一个仿 Delphi 代码编辑器操作方式，在 Win9X 下无 64K 限制的编辑器控件。
- 8、其它一些风格的界面控件，如仿 Foxmail 4.0 中提示窗口的控件、仿 Office Xp 风格的控件、液晶显示样式控件等。

四、符合中国人使用习惯的数据库、报表组件包。

为中国人设计的数据库、报表组件。

已计划的内容有：

- 1、扩展的数据库表格控件，支持多表头、自动缩放、多字体显示、多种内置编辑器、多种编辑器文本对齐方式等功能。
- 2、扩展的数据库感知组件。
- 3、用以取代无源码的 QuickReport 的报表组件包。

五、网络通讯组件包

扩展的网络通讯组件。

已计划的内容有：

- 1、串口通讯组件包，包含串口通讯、Modem 控制、串口设置对话框等组件。
- 2、并口通讯组件。
- 3、局域网通讯组件包。
- 4、用以取代无源码的 FastNet 的 WinSock 组件包。

六、本地化、扩展的属性、组件编辑器。

用来提高开发效率的工具，全部采用中文界面。

已计划的内容有：

- 1、改进的字符串列表属性编辑器。
- 2、改进的图像属性编辑器。
- 3、改进的数值属性编辑器，增加计算器和进制转换等功能。

- 4、 其它一些属性编辑器，可将已有属性的内容保存在属性字典中以供以后使用。
- 5、 一个通用的组件编辑器，可将组件的部分属性（自定义）保存在组件字典中以供以后使用。
- 6、 CnPack 中复杂组件的组件编辑器。

七、 IDE 扩展专家工具包。

用来提高开发效率及与开发网站集成的专家工具包，全部采用中文界面。

已计划的内容有：

- 1、 DLL 型 Wizard (Experts) 管理工具。
- 2、 一个以 DelForEx 功能为蓝本的源码格式化工具，提供可完全定制的代码处理方式、支持内嵌汇编格式化、大小写字母智能转换（在字典方式基础上）函数过程按类名、字母顺序排序等功能，最重要的是与 CnVCS 版本控制系统兼容。
- 3、 文本查找替换工具，在 Delphi 自带、GExpert 等工具基础上提供多文档替换、忽略注释或相反等功能。
- 4、 组件选择工具，提供按类型、名称（支持通配符）父控件、表格等方式选择单个或多个组件的功能。
- 5、 组件前后缀命名控制工具，可按标准或自定义的前后缀命名规范批量修改组件名、自动命名新组件的专家工具。
- 6、 代码仓库工具，提供功能强大的代码库、Tips 管理功能。
- 7、 宏替换工具。
- 8、 自动代码注释工具，可以产生一些简单的注释并允许用户扩充。
- 9、 “每日提示”工具，根据用户设置类别或智能分析让程序员在每次启动 Delphi 时学到一点新知识。提示内容可自动从开发网站更新。
- 10、 开发网站集成工具。程序员可利用该工具检索开发网站资源、与其它程序员在线交流、给开发网站或其它程序员留言或阅读留言、报告开发包 Bug、下载新的开发包等。
- 11、 单元引用分析工具，快速分析一个 Project 中引用到的所有单元（Delphi 内部的和外部的）和第三方组件，可自动创建组件包文件，用于快速将整个工程移植到其它的计算机上。
- 12、 使内嵌汇编支持 MMX、3DNow!、SSE 等扩展指令的工具（转换为 DB 指令）。
- 13、 允许改变所选控件（集）的父控件或新建一个窗口控件将当前选择的控件（集）包含进去或删除控件的父控件而保留控件的工具。
- 14、 其它辅助的 IDE 扩展设置，如修改 IDE 窗体字体、设置、定时存盘等。
- 15、 自动生成用户记录类型列表、类安全列表（List）源码的工具。
- 16、 其它一些窗体、模块、工程专家（RepositoryWizard）。

八、 CnVCS 版本控制系统。

用于多人协同开发及项目管理的 IDE 集成工具。

将 Delphi 的 Project 或 ProjectGroup 概念扩大到软件项目，包括源码、设计文档、帮助文件、资源、备份、安装程序等文档的集中管理。

已计划的功能有：

- 1、 “软件项目”文档管理功能。
- 2、 用于多人协同开发的单元访问权限控制功能。
- 3、 源码自动备份功能。
- 4、 源码修改自动记录功能并将改动登记到数据库允许恢复。
- 5、 系统版本号（主、子、Build）自动控制功能。
- 6、 单元头注释统一控制功能。

九、 辅助开发工具。

本地化、功能扩展的辅助开发工具。

已计划的内容有：

- 1、 数据桌面。
- 2、 资源编辑器。
- 3、 组件编写工具。

十、可重用的窗体模板库。

已计划的内容有：

- 1、 各种风格的关于窗口模板。
- 2、 各种风格的启动窗口模板。
- 3、 操作向导窗口模板。
- 4、 各种对话框模板。
- 5、 不同框架结构的主窗体模板。

四 开发人员组织方案

一、关于多人开发。

基于 CnPack 开发包结构庞大，涉及面广，开发工作量大的特点，开发形式确定为多人共同开发。

二、开发成员的确定。

开发组采用“个人自愿报名 - - 开发包设计组通过 - - 加入 - - 工作划分”的方式组建。

凡符合以下条件的程序员都可申请加入开发：

- 1、 有锐气、有责任心、愿意为中国的免费软件事业贡献一份力并从中汲取知识、经验且以之为乐。
- 2、 具有一定的 Delphi 开发或网站设计、维护能力。
- 3、 在不影响正常学习、工作的前提下，有足够的时间从事开发、维护工作。
- 4、 具备上网条件。

5、 自愿开发成员根据自身能力、条件、爱好选择适合的工作参与开发。《附一 Delphi 开发能力自我评测》是一份简单的测试可供开发人员评估自己的能力（仅供参考）。

有意加入 CnPack 开发组的朋友，请参考《附二 CnPack 开发组报名申请函》中内容及样本填写一份表单寄到开发包设计组（yygw@163.com）。

三、开发组成员维护。

开发组成员名单由开发包设计组维护，当成员由于某种原因不能继续从事开发工作时，可以自由退出开发组。

- 1、 开发组成员自愿退出，应向设计组提交申请，并将已有开发作品及相关资料一同发送。
- 2、 开发组定期与开发成员联系，超过指定期限仍未取得联系的成员视为自愿退出。
- 3、 退出开发组后，最终作品中由该成员开发的部分版权仍属于该成员及开发组共同所有。
- 4、 开发成员退出后，仍可在任何条件下重新加入开发，并获得所有原有的权利。

四、开发小组的结构。

整个 CnPack 开发组由开发包设计组、代码编写组、辅助开发组、网站维护组组成，开发成员可兼职。

各小组分工如下：

1、 开发包设计组。

负责制定开发计划、维护开发小组成员名单、编写开发文档协议、掌握开发进度、协调开发工作等开发管理工作。

2、 代码编写组。

负责开发包主代码编写。又分为不可视工具组件开发小组、界面控件开发小组、数据库报表开发小组、网络通讯开发小组、专家开发小组、版本控制系统开发小组及外部工具开发小组等相对独立的部分。

3、 辅助开发组。

负责开发包整理、测试、开发文档维护、提供开发素材及开发包中窗体模板库部分的开发等工作。

4、 网站维护组。

负责维护 CnPack 开发网站、整理、发布开发包、接收用户反馈意见、定期联络开发成员等工作。

五、开发包设计组。

由一名或多名开发成员组成。

在最终的开发包设计组中应具备以下条件：

- 1、存在并保持以下两种人（并不特指具体的人，也可以是一人具不同特质或多人互相补充）并取得默契与平衡。
 - 1）乐观进取、遇事果断、敢想敢做、在最困难的时候也能鼓舞士气、具备宏观思维能力的组织者。
 - 2）思维缜密、经验丰富、稳重严谨、擅于分析处理复杂关系、长于系统设计的建设者。
- 2、作为开发组核心组成部分，设计组成员要有足够的时间、精力以及上网条件来承担开发包的设计维护工作。
- 3、设计组成员应保持稳定。

六、代码编写组。

每个独立的开发小组由一到三名成员组成，是开发组的主要组成部分。开发成员可兼职。

1、不可视工具组件开发小组。

负责开发包中工具组件开发以及本地化增强功能的属性、组件编辑器开发等工作。开发人员应具备较多的 Delphi 与操作系统底层方面的知识。

2、界面控件开发小组。

负责开发快速图像处理库及窗口、图形界面控件。开发人员要求具有丰富的图像处理编程及控件开发经验。

3、数据库报表开发小组。

主要工作是开发符合中国人使用习惯的数据库及报表控件，以取代 Delphi 中无源码的 QuickReport 控件包及制作增加功能的数据库增强、数据感知控件。开发者应具有丰富的数据库编程经验。

4、网络通讯开发小组。

Delphi 中已提供了强劲的网络通讯、WEB 开发组件。如果有网络编程高手愿意改进无源码的 FastNet 组件包及增加串并口通讯、Modem 控制及其它相关网络组件，可成立该小组。

5、专家 (Wizard) 开发小组 (难)。

负责开发包中专家工具包的开发。编写专家要求对 Delphi 系统结构有相当深入的了解及相当丰富的编程调试经验，如果短期内没有合格的开发人员，开发进度将延期。

6、版本控制系统开发小组 (难)。

CnVCS 版本控制系统是开发包中最复杂的部分之一，同专家开发组，该部分开发进度待定。

7、外部工具开发组。

负责开发本地化的数据桌面、资源编辑器、组件编写工具等外部工具，以取代 Delphi 自带的工具。有一定 Delphi 相关开发经验的程序员可胜任该工作。

七、辅助开发组。

有志于开发而组件开发能力稍弱或学习工作较忙的自愿者可加入辅助开发组。

辅助开发组将是开发包得以最终完成的强有力保证，这一部分工作大体可分为以下几方面：

1、参与开发包的规划和测试。

制作开发包的最终目的是为了中国 Delphi 程序员的工作效率，离开了实用性就毫无意义。开发包的具体内容将由所有开发成员及热心朋友共同制定，开发完成的代码也将随时发布供测试使用。辅助开发组成员即为正式测试员。

2、维护开发文档、提供开发资料。

由于采用多人开发的形式，即使制定再严密的开发规范，也不能保证所有开发成果的严谨和一致，辅助开发成员将对已编写的代码进行规范化整理、注释、制作 Demo 及编写相应说明文档。

同时，辅助开发成员还将为代码编写成员提供由各种方式获取的免费源码、资料以及制作图标、图片等素材。

3、翻译文档。

辅助开发组还承担着翻译开发所需要的英文资料以及将帮助、文档翻译成英文的任务。

4、创建窗体模板库。

Delphi 提供的对象仓库 (Object Repository) 功能极大的方便了程序员的开发，除了 Wizard 型的专家，简单的窗体模板好象很少有人对它们进行收集整理和发布 (也许是我孤陋寡闻)，辅助开发成员还将创建大量可重用的窗体模板以提高开发效率。

八、网站维护组。

由上网方便、愿意参与网站维护管理的成员组成。

CnPack 的开发网站作为开发包的门面有着举足轻重的地位，而 CnPack 的一个设计目标则是开发包与开发网站的结合，最终可能提供类似于 Delphi.NET.CN 的功能。

1、整理、发布开发包。

随时整理开发成果、发布最新开发包供下载测试、公布开发进度、维护开发相关资料。

2、宣传、维护开发包及开发网站。

为了让更多的 Delphi 程序员能利用 CnPack 和各种优秀的开发工具、开发包提高工作效率，推动中国软件事业的发展，网站维护组成员还肩负着开发包及开发网站推广的重任。

3、接收测试反馈意见，协调开发成员交流。

如前所述，开发包以实用高效为最终目的，网站维护组成员应及时将测试人员和用户的反馈意见整理发给代码编写成员，并协调开发成员内部交流等工作。

五 开发网站建设

一、开发网站及其发展。

在开发计划中，CnPack 开发网站“中文 VCL 开发中心”(以下简称网站)，在不同的阶段其建设重点也不一样。

1、正式开发前，网站的任务是介绍 CnPack 的内容、吸收自愿者、听取热心朋友们的建议，为 CnPack 正式开发方案的确立做准备。

2、在开发前、中期，网站的中心主题是为 CnPack 的顺利开发而服务。

开发人员通过网站的内部论坛交流经验、交换意见、查找资料，在网站上发布最新的代码和文档，开发包设计组、网站维护组则通过网站联络开发成员、公布开发进度、发布开发包、接收自愿者报名以及听取用户意见、接收测试反馈、解答用户问题等。

3、在开发后期及开发包初步完成后，网站的作用逐渐扩大到与 VCL 使用、开发相关的方方面面。

网站不再只是 CnPack 的开发中心，而成为中国 Delphi 程序设计门户网站之一。

在开发资源方面，网站绝对不会象一些 Delphi 资源库那样只是个大而全的代码堆积库。相反，它应该以引导程序员使用组件、分析组件、编写组件、深入 Delphi 内部最终掌握面向对象的 Windows 程序开发以及最大的代码重用性为目的。网站上面的每一个组件、源码都应该具有典型性，配有详细的中文使用说明及 Demo。对国内外已有的优秀组件、源码，开发人员在对其进行分析后，负责为它们编写附加的中文说明文件及演示代码。

事实上，网上并不缺乏资料，而是缺少快速找到最佳资料的方法和从中切实汲取到经验提高效率的途径，而开发网站将在这一方面努力。

在 VCL 扩展方面，从最基本的控件基础知识、安装方法、控件在 Delphi 程序开发中的作用到简单控件的编写、VCL 体系结构到属性、组件编辑器、Open Tools Api，网站上都将有详细的介绍和演示。

作为“开发中心”，CnPack 的开发人员们还将通过网站这个窗口，在与用户交流的过程中不断为 CnPack 增加新的功能、控件。

4、最终发展方向：中国的 Delphi 在线开发网站。

在积累了大量开发维护经验后，最终的开发包中包含的 IDE 扩展专家工具，将把用户的 Delphi 开发环境与 CnPack 开发网站紧密集成，提供类似 Delphi.NET.CN 的功能。

二、开发网站建站方案。

开发网站的设计思想是：结构清晰、主题明确、风格一致、内容翔实。

与免费的开发包相应，网站上将不链接任何商业广告，但可能会有一些关于环保、友爱、学习等方面的公益广告。

更多网站构建方面的信息参见《“中文 VCL 开发中心”建站方案》。

六 开发包设计规范

一、设计规范是决定开发包最终质量的重要因素。

我希望 CnPack 既是一个功能强大的开发包，同时也是一份可供程序员学习参考的资料库，开发包的设计将严格按以下规范来进行。

二、代码、注释编写规范。

很多程序员对代码编写规范不屑一顾，在我读过的很多源码中，不乏编写得相当规范的，但更多的则书写凌乱、阅读困难，而我打开代码做的第一件事就是先用源码格式化工具处理一遍再阅读。

《附四 Delphi代码标准文档》是一份详细的代码编写规范说明，文档从<http://jediedit.heha.net>下载而来，感谢 [Riceball\(teditor@mailroom.com\)](mailto:Riceball(teditor@mailroom.com))及文档的其它作者们为此做出的贡献。

在本文档中定义的规范中与《附四》中规定冲突的内容以本文档中定义的为准。

每一个代码单元（.pas、.dpr、.inc、.asm 等）都应包含一个统一格式的文件头，以注明该单元的版权信息及相关内容，示例参见《附五 开发包单元标准格式》。

三、专用术语及标识符命名规范。

以下的定义在整个 CnPack 开发包内使用，不再另行注明。

1、开发包及相关术语定义。

“开发包”（CnPack）。开发包的全称为“CnPack For Delphi 开发包”，简称“CnPack”，在所有文档中以“CnPack 开发包”或“CnPack”或“开发包”来描述。

“组件”（Component）指由 TComponent 派生而来的 VCL 类，包括不可视组件和控件，很多时候组件特指不可视组件。

“控件”（Control）指由 TControl 派生而来的组件，包括图形控件和窗口控件。

“图形控件”（GraphicControl）指由 TGraphicControl 派生而来的控件。

“窗口控件”（WinControl）指由 TWinControl 派生而来的控件。

“伪可视组件”指封装了窗体或对话框的不可视组件，如 TOpenDialog、自定义的 Hint 组件等。

“专家”（Wizard 或 Expert）指实现了 IOTAWizard 接口的 IDE 扩展工具，在文档中 Wizard 和 Expert 一词具相同意义，推荐使用 Wizard，很多时候专家特指开发包中的以 DLL 形式提供的专家库中的专家工具。

“对象仓库专家”（Repository Wizard）指实现了 IOTARepositoryWizard 接口的专家，包括窗体专家和工程专家，简称“仓库专家”。

“窗体专家”（Form Wizard）指实现了 IOTAFormWizard 接口的仓库专家。

“工程专家”（Project Wizard）指实现了 IOTAProjectWizard 接口的仓库专家。

“菜单专家”（Menu Wizard）指实现了 IOTAMenuWizard 接口的专家，加载后在 Help 菜单中增加新项。

“AddIn 专家”（AddIn Wizard）指 IOTAWizard 接口并以 DLL 形式提供的专家，通过使用 IDE 提供的服务接口，在后台运行或手工在主菜单中插入新的菜单项。

“工程”（Project）、包（Package）、设计时间包（Design-time packages）、运行时间包（Runtime packages）、窗体（Form）、对话框（Dialog）、属性编辑器（PropertyEditor）、组件编辑器（ComponentEditor）及其它 Delphi 中的专用术语按通用的定义使用。

2、标识符命名标准。

如果不说明，下面的定义中“标识符”一词仅指单元中 interface 部分出现的声明内容，要求区分大小写。

所有由开发组成员编写的单元、窗体、模块、组件、程序（以下简称模块）一律以 Cn 为前缀，比如 CnFastRGB（单元）、TCnResEdtMainForm（窗体类）、CnDBDataModule（数据模块实例）、TCnButton（组件）、CnVCS（程序）、CnDefine.inc（预编译包含文件）等。

所有引用非 CnPack 开发包的外部模块，模块名保持不变，如 zlib、Jpeg 等。

出现在 interface 部分的非组件类、记录、枚举、集合等类型的名称推荐也以 Cn 为前缀，以避免冲突。

在代码内部出现的所有文本字符串（除了绝对不需要本地化的以外）应定义为资源字符串（resourcestrings），一个相对

独立的组件包应建一单独的字符串资源文件包含所有可本地化的字符串，并以 CnXXXConsts 命名。如 CnDBConsts 为数据库组件包用到的资源字符串常量定义单元。

各个组件、单元的大部分标识符命名应在正式开发前由开发小组负责人制定。

开发人员在确定单元框架后应将它发给设计组统一管理，以避免潜在的命名冲突。

四、开发包版本定义。

版本号定义分开发包版本号及单元（组件、专家、软件）版本号两部分。

1、开发包完整的版本号以 VA.BB.BuildCCCC DDDD 的形式出现。

其中 A 为一位（或二位）主版本号，在正式版发布前从 0 开始；BB 为二位数版本号，从 00 开始；BuildCCCC 为四位建立开发包的日期（MMDD 格式）；DDDD 为附加说明，Alpha 为内部测试版，Beta 为外部测试版，无为正式版，附加说明后可跟一位数字。

如 CnPack V1.01.Build0313 Beta2 含义为 3 月 13 日发布的 1.01 版开发包的第二次外部测试版。

2、单元版本号在单元内部出现，标识其内部版本，主要在开发组内部使用。

单元版本号大致遵循开发包版本号的约定，只是可不带 Build 及附加测试说明部分，但单元文件内应注明更新日期，另其主版本号从 1 开始。如 CnDBGGrid V1.03。

五、界面设计规范。

开发包中包含的属性、组件编辑器、专家工具及软件的界面应保持风格一致，要求操作方便、界面美观、资源占用低。

设计用户界面时注意以下几点：

- 1、同一个或同一类软件、工具界面风格要求一致。
- 2、控件的 Tab Order 要求合理设置。
- 3、尽量多为控件、菜单分配快捷键或热键。
- 4、控件位置应排列整齐、错落有致。
- 5、尽量使用框架（Frames）和继承窗体技术以实现代码重用。
- 6、窗体及控件应带记忆功能，能自动保存最后一次的状态。

六、帮助文件编写。

为了将开发包帮助文件加入到 Delphi 的 Open Help 中，帮助文件以 Win95 风格.hlp 形式出现。

开发包中的每一个组件、工具、程序都将带有完整的帮助和 Example，正式发布的开发包安装程序将自动把帮助文件注册到 Delphi 的帮助环境中，而帮助文件的格式也遵循 Delphi 帮助文件格式规范。

开发包设计语种为简体中文，故帮助文件也以简体中文为首选语种，如果正式版发布时推出英文版和繁体中文版，再统一进行翻译。

七、相关文档编写规范。

开发包在开发过程中将创建大量的文档，包括开发包开发方案、各部分开发方案、组件、工具功能定义、结构设计、说明文档等，这些文档都应使用标准的格式编写：

- 1、文件格式使用 Doc 格式（Word97/2000/Xp 兼容）、Txt 格式、Html 格式三种，视具体情况而定，推荐使用 Doc 格式。
- 2、图像格式使用 Jpeg、Gif、Bmp 三种格式（光标、图标除外），推荐使用 Jpeg 和 Gif 格式。
- 3、文档中使用的正文文字，推荐使用宋体，Verdana，小五号。
- 4、《附六 开发包文档标准格式》中包含了标准的文档格式例子、模板、样式表。
- 5、在最终发布的开发包文档中，一律使用统一标准的文档格式，建议开发人员在编写时尽量遵循这些规范，以减少发布整理工作。

七 开发包开发方法

一、开发平台和开发工具。

在最初设计阶段，要求开发包支持的平台为简体中文 Windows9X/Me/NT/2000/Xp+Delphi5/6，开发过程中使用简体

中文为首选语种。正式发布后，可能会推出英文版和繁体中文版，同时不排除将来支持 BCB 和 Kylix 的可能。

由于采用多人开发方式，开发阶段开发人员使用的平台和工具很难统一，为了保证最终产品的平台兼容性和可移植性，要求开发人员在开发时考虑周到、细心测试。

1、 操作系统。

开发人员可在简体中文 Windows9X/Me/NT/2000/Xp 中选取用于开发的操作系统平台，推荐使用 Windows98SE 或 Windows2000，有条件的开发人员在代码编写完成后可在不同平台上进行测试，也可将代码交由专职测试人员进行兼容性测试。某些与操作系统底层相关的代码如鼠标取词组件，可采用推出不同的版本、使用条件编译、运行期自动识别等方式来实现平台兼容。

2、 开发语种。

开发包在开发期，所有对外发布的文字（帮助、文档、说明等）一律使用简体中文，单元头注释块统一使用中文、英文两段描述文字，代码注释可使用中文或英文或两者兼有，推荐使用中文或两者兼有。

3、 开发工具。

基于对当前主流开发工具的考虑，开发包将不支持 Delphi1-4，目前仅设计为支持 Delphi5/6，开发人员可在这两种工具中选取，推荐使用 Delphi5，因为支持 Delphi5 的代码要移植到 Delphi6 中相对要容易一些。正式发布的开发包中，将采用以下两种发布形式之一：

- 1) 同一个包同时支持 Delphi5/6，使用条件编译指令来实现。这种方法的好处是维护容易，但开发困难。
- 2) 针对 Delphi5/6 发布两个包。采用这种方法将增加维护工作量，但较易实现。

最终采用哪种方式视具体开发情况而定。

二、 开发包整体开发过程。

整个开发包的开发进程分为以下几个阶段：

1、 开发包原型设计及开发网站建设。

包括开发包功能定义、开发方案拟定、网站建设方案拟定、开发组组建等前期设计工作，由开发组核心成员完成。

开发网站也应在正式开发前投入使用，以提供开发成员相互交流的场所。

这一阶段拟在 2002 年 1 月底完成。

2、 前期代码、资料准备。

在开发方案基本定型、开发组组建完成、网站建立后，开发人员开始准备开发资料、创建公共代码（公共过程、函数、常量定义）、收集光标图标、图片、声音等开发资源。同时，这一段时间也是开发人员特别是同一组的成员互相熟悉、磨合的过程。

开发包中一些功能的实现是有先后顺序的，这些因素也将在这一期间考虑。

这一阶段拟在 2002 年 2 月 - 3 月进行。

3、 主功能代码实现。

开发过程中最重要的部分，将消耗开发人员大量的精力和时间来完成。

根据最终开发包内容及开发人员状况，这一阶段将需要数月至一年的时间。

4、 测试发布。

由于多人业余开发进度的不确定性，主代码的编写将会是一个长期的过程，故测试期开发包的发布采用分散的方式进行。每个开发小组甚至单个组件都可在编写、内部测试结束后单独发布测试。

开发组定期对已完成的部分进行整理，发布较完整的开发包。而从最初发布测试版到第一个正式版的推出将需要相当长的时间。

5、 维护升级。

开发包的维护和完善将贯穿于开发过程始终。

在规划中，预计在 2002 年底或 2003 年上半年发布 CnPack V1.0 正式版，正式版发布后的升级视具体情况而定。

三、 开发包各部分开发过程。

在理想状态下，开发包各部分的开发以类似流水线作业的方式进行：

（下面的文字中，“项目”一词指组件、专家工具、软件等开发包内容。）

1、 内容拟定。

定义项目功能的详细描述，这一部分工作在开发包设计期进行，主要由设计组及各小组负责人完成，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。

设计项目的框架结构，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，由开发小组负责人或该项目负责人完成。结构设计完成后应发到设计组统一整理。

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，由开发小组完成。

4、 主代码创建。

开发小组最重要的工作内容，包括开发人员调试和测试过程。

5、 开发包设计组评测。

主代码完成后，发送到设计组对代码进行登记、评测和整理。

6、 辅助开发组内部测试。

辅助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。

由辅助开发组人员根据项目功能、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。

网站维护组定期对已完成的项目集中发布，供下载和外部测试，项目的“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。当项目做大的升级时（主版本变化），再回到前面几步重新进行。

四、 多人开发与版本控制。

在多人开发中，组织和协调工作将是开发最重要也最困难的部分。

由于开发人员、地点、时间的离散性，多人同时开发同一单元是不现实的，故采取以下的方法进行开发控制：

（下面使用“项目”一词来描述一个开发小组的工作内容或组件包或单独的软件）

1、 同一项目，只安排 1 - 3 名经验较丰富的成员开发，由小组负责人负责规划和协调。

2、 在项目正式开工前，开发成员应起草一份较完整的开发计划，同时公共代码、开发资源也应在前期准备充分。

3、 同一项目的开发人员应互相了解、熟悉，包括基本资料、编程风格、开发能力等。

4、 项目的开发计划应详细到模块接口定义及具体单元内容，原则上同一单元在同一阶段由单人开发，而最后的整理与组合由项目负责人及成员完成。

5、 使用 CVS 服务器进行版本控制管理。

6、 项目成员在开发期间应保持紧密的联系，如果联系中断或成员中途退出，应及时与设计组联系。

（完）