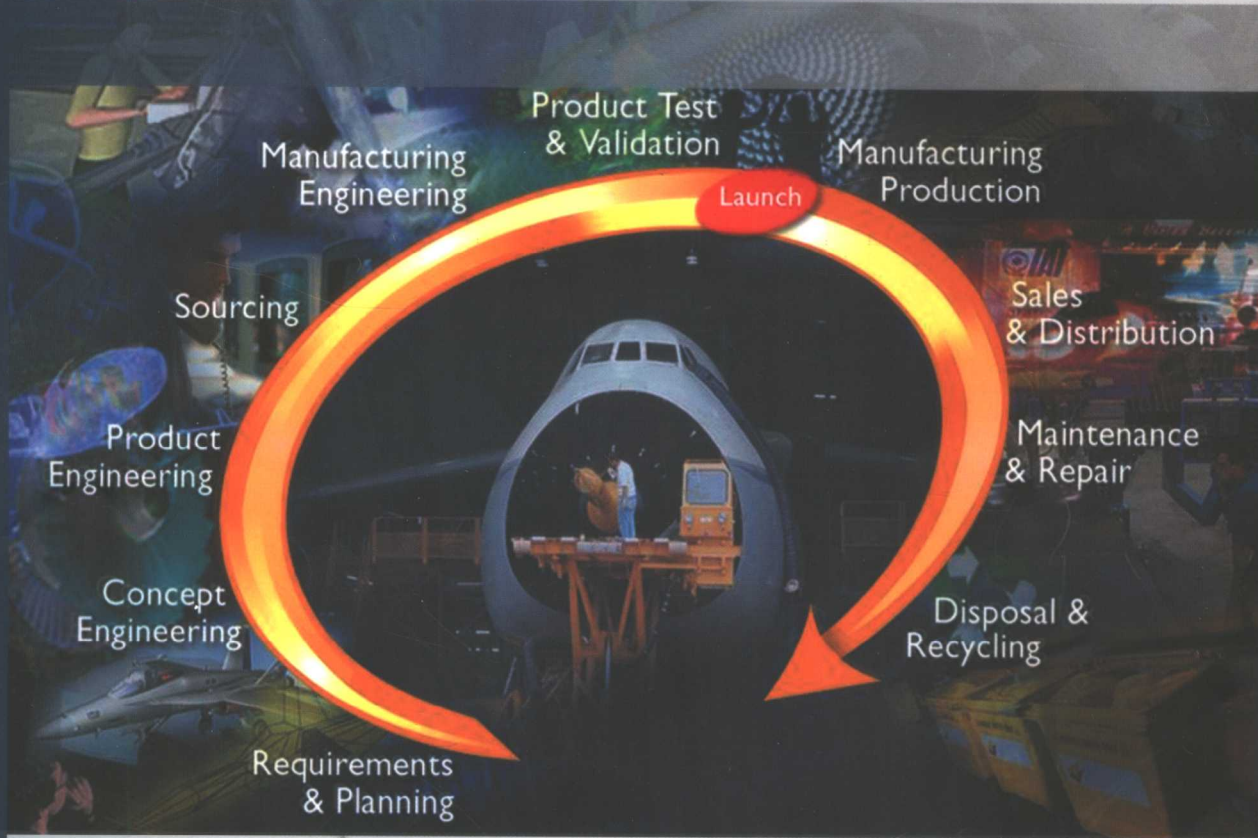




UGS PLM 应用指导系列丛书

The PLM Company

UG 应用开发教程与实例精解



黄翔 李迎光 编著
李迎光 审校



清华大学出版社

UGS PLM 应用指导系列丛书

UG 应用开发教程与实例精解

黄翔 李迎光 编著

李迎光 审校

清华大学出版社

北 京

内 容 简 介

UG 应用开发是指在 Unigraphics 软件平台上, 结合具体的应用需求, 总结行业的设计知识和经验, 开发面向行业和设计流程的 CAD 系统。本书在介绍使用 UG/Open 工具进行 UG 应用开发的基础上, 系统阐述了应用系统开发的组织、开发规范的制定和开发技巧等。本书还给出了 UG/CAD 应用标准化工具 (UCDCS) 的开发实例, 配书光盘中提供了该工具软件, 并公开了核心源代码。

本书融入了作者多年来从事 CAD/CAM 研究与开发的心得与体会, 通过阅读本书可以理解和掌握应用系统开发的构架、规划, 以及开发规范的制定, 以确保开发目标的实现与软件质量。本书可作为 UG 应用开发人员的高级培训教材, 适用于具备 UG 初步应用知识和 C/C++ 编程基础的人员。

版 权 声 明

本系列丛书为 UGS PLM Solutions (中国) 公司 (原名: 优集系统 (中国) 有限公司) 独家授权的中文版培训教程与使用指导。本书的专有出版权属清华大学出版社所有。在没有得到 UGS PLM Solutions (中国) 公司和本丛书出版者的书面许可, 任何单位和个人不得复制与翻印。

版权所有, 违者必究。

“Copyright 2000 by Unigraphics Solutions Inc.

Original English Language Edition Copyright

2000 by Unigraphics Solutions Inc. All Rights Reserved”

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图 书 在 版 编 目 (CIP) 数 据

UG 应用开发教程与实例精解/黄翔, 李迎光编著. —北京: 清华大学出版社, 2005.4
(UGS PLM 应用指导系列丛书)

ISBN 7-302-10649-5

I. U… II. ①黄… ②李… III. 计算机辅助设计-应用软件, UG IV. TP391.72

中国版本图书馆 CIP 数据核字 (2005) 第 020117 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 许存权

文稿编辑: 李虎斌

封面设计: 姜凌娜

版式设计: 崔俊利

印 装 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印 张: 14.75 字 数: 317 千字

版 次: 2005 年 4 月第 1 版 2005 年 4 月第 1 次印刷

书 号: ISBN 7-302-10649-5/TP·7211

印 数: 1~5000

定 价: 29.00 元(附光盘 1 张)

前言

Unigraphics 是当今世界上紧密集成的、面向制造业的 CAID/CAD/CAE/CAM 高端主流软件。UGS 公司在全球已拥有 42 000 家客户, 装机量达 330 万台。自从 1990 年进入中国市场以来, UG 软件发展迅速, 已广泛应用于航空航天、汽车、机械、家电等领域。随着 UG 的普及, UG 的深化应用不断得到加强, 许多优秀的用户已不满足于 UG 现有的功能, 需要将设计过程中获取的产品开发经验, 包括整个产品的几何模型、逻辑规则、开发过程、标准工程方法和制造规则融入到 CAID/CAD/CAE/CAM 系统中。UG 提供了较完善的应用开发环境, 在该环境下, 应用开发程序可以建立与 UG 系统的链接, 使新开发的功能和原有功能无缝地集成在一起。为了使用户理解与掌握 UG 应用开发的方法, 我们在总结多年开发经验的基础上, 整理出了本书。

本书共分 6 章, 第 1 章介绍了 UG 应用开发的基本概念; 第 2 章介绍了 UG/Open API 的基础知识; 第 3 章系统论述了 UG 应用开发框架和流程, 内容涉及应用向导、程序框架、环境配置、工程目录结构和项目的发布等; 第 4 章全面总结和提出了 UG 应用开发规范, 包括编码命名规则、常用标准对话框、信息的发布、函数的监控以及资源的分配与释放等; 第 5 章总结了我们多年来在 UG 应用开发实践中的一些经验和技巧, 涉及的内容包括辅助开发工具、程序的调试、装配操作相关开发、二维制图相关开发、用户配置文件的访问和一些实用方法等; 第 6 章详细介绍了 UG/CAD 应用标准化工具 UCDCS 的开发过程, 包括系统的功能模块与设计流程等。UCDCS 是为规范 UG 系统的应用, 从信息集成的角度出发, 在 CAD 系统应用层面上提出的企业 CAD 应用标准化工作方法, 内容涉及文件结构、文件数据组织、种子文件定制和配置文件管理等, 该系统已成为规范 UG 应用的必备工具, 在国内外得到广泛使用并获得好评。

本书在介绍 UG/Open 的基础上, 深入阐述了如何组织应用系统开发、如何制定开发规范, 以及一些开发技巧等。为了提高用户应用 UG 的水平, 本书提供了 UCDCS 工具软件, 并公开了核心源代码, 用户可以直接安装运行附盘上的 UCDCS 系统, 也可以根据企业具体的标准化需求, 对源代码进行修改, 创建具有自己风格的 CAD 应用标准化工具。

本书面向有志于从事 UG 应用开发并具备 UG 初步应用操作和 C/C++ 编程基础的开发人员。读者通过学习本书能够掌握 UG 的应用开发方法并应用于开发实践。

潘志毅、方挺立和余晶等研究生参加了本书的编写和源代码的整理工作。此外, 在 UCDCS 的开发过程中得到了 UGS (中国) 有限公司方正、张社教、章军、张振亚和余国华先生的大力支持和帮助, 在此表示衷心的感谢。

欢迎读者与我们联系: xhuang@nuaa.edu.cn 和 welcome.li@nuaa.edu.cn。

作者

2004 年 12 月

目 录

第 1 章	UG 应用开发概述	1
1.1	Unigraphics 简介	1
1.2	CAD 软件的二次开发	2
1.3	UG 应用开发工具	3
1.4	一个简单的程序	3
第 2 章	UG 应用开发基础	6
2.1	UG/Open API 语法基础	6
2.1.1	概述	6
2.1.2	UG/Open API 数据类型	7
2.1.3	UG/Open API 函数	9
2.1.4	UG 对象类型与操作	11
2.2	UIStyler 对话框设计	13
2.2.1	UIStyler 可视化界面	13
2.2.2	控件设置	14
2.2.3	控件布局	15
2.2.4	回调函数定义	16
2.2.5	对话框文件的输出	18
2.3	对话框控件的访问	18
2.3.1	对话框控件的访问方法	18
2.3.2	访问字符串输入框 (String)	19
2.3.3	访问按钮控件 (Push Button)	20
2.3.4	访问单选按钮 (Radio Box)	22
2.3.5	访问下拉列表框 (Option Menu)	24
2.3.6	访问单选列表框 (Single Selection List)	26
2.4	MenuScript 应用	28
2.4.1	菜单脚本文件及其语法	28
2.4.2	扩展 UG 功能菜单	30
2.4.3	工具栏文件及其语法	32
第 3 章	UG 应用开发框架	34
3.1	一个具体实例	34

3.1.1	使用 MFC 应用向导	34
3.1.2	设置 UG 应用开发环境	36
3.1.3	创建用户菜单和工具图标	38
3.1.4	编写消息映射代码	39
3.1.5	创建 UG 风格对话框	42
3.1.6	创建 Windows 风格对话框	47
3.1.7	工程项目的发布	49
3.2	应用向导与动态链接库	51
3.2.1	UG/Open AppWizard	51
3.2.2	Win32 Dynamic-Link Library	51
3.2.3	MFC AppWizard (dll)	51
3.2.4	动态链接库技术	52
3.3	User Exit	53
3.3.1	User Exit 概述	53
3.3.2	ufusr()	53
3.3.3	ufsta()	54
3.3.4	其他 User Exit	55
3.4	注册工程路径	57
3.4.1	配置文件法	57
3.4.2	修改环境变量法	58
3.5	菜单激发方式	58
3.5.1	菜单注册函数	58
3.5.2	“菜单激发应用”方式	59
3.5.3	“菜单激发对话框”方式	60
3.5.4	两种菜单激发方式的比较	61
3.6	工程目录结构	62
第 4 章	UG 应用开发规范	64
4.1	编码命名规则	64
4.1.1	文件命名规则	64
4.1.2	函数命名规则	64
4.1.3	类型命名规则	65
4.1.4	常量与宏的命名规则	65
4.1.5	变量命名规则	65
4.1.6	tag_t 命名规则	67
4.1.7	路径命名规则	68
4.2	常用标准对话框	69
4.2.1	消息对话框	69

4.2.2	文件(夹)操作对话框.....	71
4.2.3	选择菜单列表对话框.....	72
4.2.4	对象选择对话框.....	73
4.2.5	其他标准对话框.....	75
4.3	信息的发布.....	77
4.3.1	UG 信息窗口.....	77
4.3.2	提示行与状态行.....	78
4.4	函数的监控.....	80
4.4.1	UF 函数的监控.....	80
4.4.2	自定义函数的监控.....	83
4.5	资源的分配与释放.....	84
4.5.1	C/C++资源分配与释放.....	84
4.5.2	UF 资源分配与释放.....	85
第 5 章	UG 应用开发经验与技巧	87
5.1	辅助开发工具.....	87
5.1.1	UF 函数检索工具.....	87
5.1.2	UF 对象识别工具.....	88
5.1.3	对象属性操作工具.....	89
5.2	程序的调试.....	91
5.2.1	输出中间结果.....	91
5.2.2	断点调试.....	91
5.2.3	使用断言.....	91
5.2.4	查询日志.....	92
5.3	装配操作相关开发.....	93
5.3.1	装配树的结构.....	93
5.3.2	遍历装配树.....	95
5.3.3	部件对象间的访问.....	97
5.3.4	装配体中的几何对象.....	100
5.3.5	部件的装入.....	101
5.3.6	配合关系的创建与编辑.....	102
5.4	二维制图相关开发.....	105
5.4.1	属性的导入.....	105
5.4.2	图框模板文件的导入.....	105
5.4.3	符号和文本控制符.....	105
5.5	用户配置文件的访问.....	106
5.5.1	访问 UG 根路径法.....	106
5.5.2	设置环境变量法.....	107

5.5.3 访问 DLL 路径法.....	107
5.6 其他经验技巧.....	109
5.6.1 菜单命令的加锁与解锁.....	109
5.6.2 设置 Undo 标记.....	110
5.6.3 引用集操作相关开发.....	112
5.6.4 装配复制方法.....	114
5.6.5 用户数据的传递.....	117
5.6.6 UF 宏的使用.....	119
5.6.7 NULL_TAG 的使用.....	120
5.6.8 菜单激发可执行文件.....	121
5.6.9 若干建议.....	121
第 6 章 CAD 应用标准化工具的开发.....	122
6.1 UCDCS 的开发背景.....	122
6.2 UCDCS 的基本内容.....	123
6.2.1 UCDCS 概述.....	123
6.2.2 文件结构.....	124
6.2.3 文件数据组织.....	125
6.2.4 种子文件定制.....	127
6.2.5 配置文件管理.....	128
6.3 UCDCS 主要模块简介.....	131
6.3.1 创建新文件模块简介.....	132
6.3.2 文件规范化检查模块简介.....	134
6.4 UCDCS 主要模块的设计流程.....	135
6.4.1 创建新文件模块设计流程.....	135
6.4.2 文件规范化检查模块设计流程.....	136
6.4.3 细节设计.....	137
附录 A UCDCS 代码.....	143
A1 FILENEW 模块代码.....	143
A2 CHECK 模块代码.....	174
参考文献.....	219

第 1 章 UG 应用开发概述

1.1 Unigraphics 简介

Unigraphics (简称 UG) 是全球主流 MCAD 系统, 是计算机辅助设计、辅助制造、辅助工程和产品数据管理 (CAD/CAM/CAE/PDM) 一体化的软件系统之一。自从 UG 推出以来, 在航空航天、汽车、通用机械、工业设备、医疗器械以及其他高科技应用领域的机械设计和模具设计、加工等领域得到了广泛的应用。

Unigraphics CAID/CAD/CAM/CAE 系统提供了一种基于过程的产品设计环境, 使产品开发从设计到加工真正实现了数据的无缝集成, 从而优化了企业的产品设计与制造。UG 面向过程驱动的技术是虚拟产品开发的关键技术, 在面向过程驱动技术的环境中, 用户的全部产品及其精确的数据模型能够在产品开发全过程的各个环节保持相关, 从而有效地实现了并行工程。

该软件不仅具有强大的实体造型、曲面造型、虚拟装配和生成工程图等设计功能, 而且在设计过程中可进行有限元分析、机构运动分析、动力学分析和仿真模拟, 从而提高了设计的可靠性。同时, 可采用建立的三维模型直接生成数控代码用于产品的加工。另外它所提供的应用开发语言 UG/Open GRIP 和 UG/Open API 功能强大、简单易学, 便于用户开发专用 CAD 系统。具体来说, 该软件具有以下特点:

(1) 具有统一的数据库, 真正实现了 CAID、CAD、CAE 和 CAM 等模块之间无数据交换的自由切换, 并且可实施并行工程。

(2) 采用复合建模技术, 将实体建模、曲面建模、线框建模、显示几何建模与参数化建模等建模技术融为一体。

(3) 采用基于特征的建模和编辑方法作为实体造型的基础, 形象直观, 类似于工程师传统的设计方法, 并能采用参数驱动。

(4) 曲面设计采用非均匀有理 B 样条作为基础, 可用多种方法生成复杂曲面, 特别适合于汽车外形和汽轮机叶片等复杂曲面的造型。

(5) 二维图功能强大, 可方便地从三维实体模型直接生成二维工程图, 可以按照 ISO 标准和国标生成各种剖视图、标注尺寸、形位公差和汉字说明等。

(6) 以 Parasolid 为实体建模核心, 目前许多著名 CAD/CAE/CAM 软件均以此作为实体造型的基础。

(7) 提供了界面良好的应用开发工具, 并能通过高级语言接口, 使 UG 的图形功能与高级语言的计算功能紧密结合, 便于用户开发专用 CAD 系统。

(8) 具有良好的用户界面, 绝大多数功能都可通过图标实现; 进行对象操作时, 具有自动推理功能; 在每个操作步骤中, 都有相应的提示信息, 便于用户做出正确的选择。

1.2 CAD 软件的二次开发

随着 CAD 应用领域的不断扩大和应用水平的不断提高, 用户需求与 CAD 系统规模之间的矛盾日益增加, 没有一个 CAD 系统能够完全满足用户的各种需求。作为商品化的 CAD 软件产品, 是否拥有一个开放的体系结构, 是衡量该软件的优劣性、适用性和生命力的重要标志, 而是否拥有一个开发简便、运行高效的二次开发平台又是开放式体系结构的核心和关键。目前, 主流的 CAD 软件都具有用户定制功能并提供二次开发工具。

通过 CAD 软件的二次开发工具可以把商品化、通用化的 CAD 系统用户化、本地化, 即以 CAD 系统为基础平台, 在软件开发商所提供的开发环境与编程接口基础之上, 根据自身的需要研制开发符合相关标准和适合企业实际应用的用户化、专业化、知识化、集成化软件, 以进一步提高产品研发的效率。在通用 CAD 基础上融入专业知识构建专用 CAD 系统是当前深化 CAD 应用的潮流。

把用户的设计思想转化为特定的新功能需要以下基本要素, 这些基本要素构成了 CAD 软件二次开发平台的基本结构:

(1) 通用 CAD 软件——管理层。通用 CAD 软件是整个开发的基础, 是二次开发应用程序的宿主。它应具有比较完备的基本功能, 即使没有二次开发应用程序, 它也能满足基本的使用需求。在二次开发平台结构中, 通用 CAD 软件属于管理层, 它所负责的工作主要包括用户界面定制、图形显示、文档数据管理、交互流程控制、消息分发和应用程序的管理等。

(2) 编程开发环境——开发层。开发者采用某种计算机高级语言(如 C/C++ 等)在特定的开发环境中进行应用程序的开发。由于通用的集成开发环境(如 VC++、VB 和 Delphi 等)具有功能强大、使用简单、可靠性强和生成代码效率高等优点, 目前一般都在通用的集成开发环境中进行二次开发。在二次开发平台结构中, 编程开发环境属于开发层, 它主要包括应用程序源代码的编辑、编译、链接、调试和代码优化等。

(3) 应用程序编程接口(API)——支持层。编程开发环境仅提供了一般性的语言支持, 在二次开发过程中, 还需要提供相应的 API 支持。通过这些 API 接口, 二次开发应用程序可以建立与原软件应用程序的链接, 使新开发的功能和原有的功能无缝集成。在二次开发平台结构中, 应用程序编程接口属于支持层, 它是用户开发的应用程序与 CAD 软件之间进行链接、通信和互操作的通道。

(4) 开发者的设计思想——知识层。一般来说, CAD 软件开发商通过以上 3 个层的引入就为用户提供了二次开发的工具和方法。此外, 二次开发应用系统还需要融入开发者的设计思想。开发者将其设计思想通过二次开发工具和方法, 并结合原有的 CAD 系统功能, 才能构成具有实用价值的应用程序。在二次开发平台结构中, 用户设计思想属于知识层,

它是开发者知识和能力的体现，是二次开发技术的应用和实践。

1.3 UG 应用开发工具

UG 应用开发（又称 UG 二次开发）是指在 UG 软件平台上，结合具体的应用需求，总结行业的设计知识和经验，开发面向行业和设计流程的 CAD 系统。UG 软件中的应用开发模块提供了较为完整的应用开发工具集，利用该工具集可对 UG 系统进行用户化裁剪和开发，用以满足实际的应用需求。UG/Open 是一系列 UG 开发工具的总称，是 UG 软件为用户或第三方开发人员提供的最主要的开发工具。它主要由 UG/Open API、UG/Open GRIP、UG/Open MenuScript 和 UG/Open UIStyler 4 个部分组成。

（1）UG/Open API（又称 User Function）是一个允许程序访问并改变 UG 对象模型的程序集。UG/Open API 封装了近 2000 个 UG 操作的函数，可以方便地对 UG 的图形终端、文件管理系统和数据库进行操作，绝大多数的 UG 操作都可以用 UG/Open API 函数实现。UG/Open API 是最常用的 UG 应用开发工具，本书以下章节主要介绍 UG/Open API。

（2）在 UG/Open API 工具发布之前，UG 的应用开发工具是 UG/Open GRIP（Graphics Interactive Programming）。GRIP 是一种专用的图形交互编程语言，它与 UG 系统集成，可以实现 UG 环境中的大部分应用操作。GRIP 语言与一般的通用语言一样，有其自身的语法结构、程序结构、内部函数以及与其他通用语言程序相互调用的接口。虽然 GRIP 的功能远不如 UG/Open API 强大，但由于其某些独特的优点，仍然有大量用户在使用。

（3）UG/Open MenuScript 是创建用户化菜单的工具。MenuScript 支持 UG 主菜单和快速弹出式菜单的设计和修改，通过它可以改变 UG 菜单的布局，添加新的菜单项以执行用户应用开发程序、User Tools 文件和操作系统命令等。

（4）UG/Open UIStyler 是开发 UG 对话框的可视化工具，其生成的对话框与 UG 集成，用户可以方便、高效地与 UG 进行交互操作。该工具的使用避免了复杂的图形用户接口 GUI 编程，直接将对话框中的基本控件进行组合与布局，可以创建满足不同功能需求的 UG 风格对话框。

1.4 一个简单的程序

为便于用户了解 UG/Open API 的使用，下面将采用 UG/Open API Wizard 建立一个简单的应用开发程序，其功能是显示一个消息对话框。运行平台为 Windows 2000 Professional，开发平台为 Unigraphics NX3 和 Visual C++ 6.0，运行结果如图 1-1 所示。



图 1-1 HelloWorld 消息对话框

创建步骤如下：

步骤 1: 使用 Unigraphics NX 应用向导新建工程项目

打开 Visual C++ 6.0, 选择菜单命令 File→New, 界面上弹出 New 对话框, 如图 1-2 所示。选择对话框中的 Projects 选项卡与项目列表框中的 Unigraphics NX AppWizard V1 项, 然后在 Location 文本框中输入工程路径, 在 Project name 文本框中输入工程名称 “HelloWorld”, 最后单击 OK 按钮确定, 弹出 Unigraphics NX AppWizard V1-Step 1 of 2 对话框。

如果在项目列表框中找不到 Unigraphics NX AppWizard V1 选项, 可将本书所附光盘根目录下的 UgOpen_v19.awx 文件复制到 Microsoft Visual Studio\Common\MSDev98\Template 路径下, 当 VC 启动时, 自动创建 Unigraphics NX AppWizard V1 项。

步骤 2: Unigraphics NX AppWizard V1-Step 1 of 2

设置选项如图 1-3 所示, 然后单击 Next 按钮。

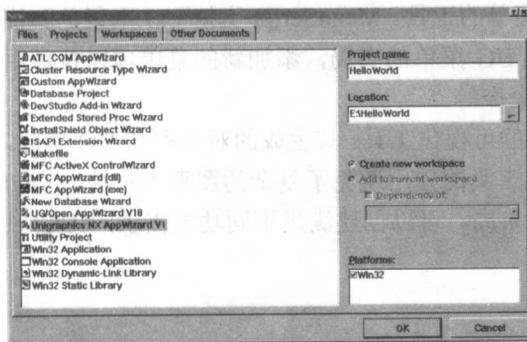


图 1-2 New 对话框

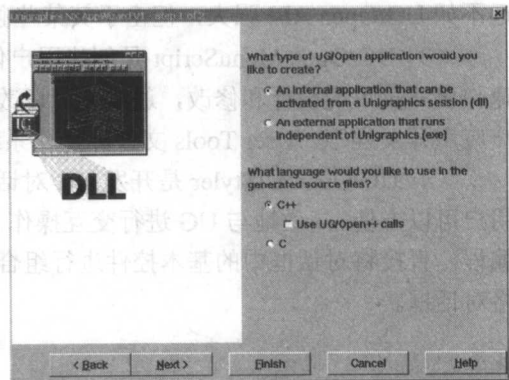


图 1-3 Unigraphics NX AppWizard V1-Step 1 of 2

步骤 3: Unigraphics NX AppWizard V1-Step 2 of 2

在弹出的对话框中 (如图 1-4 所示), 只选中 How would you like to activate the application? 选项中的复选框 Explicitly (ufusr), 然后单击 Finish 按钮。

步骤 4: New Project Information

系统弹出 New Project Information 窗口, 如图 1-5 所示, 窗口中列出了所有选项信息。

确认无误后单击 OK 按钮，系统将在指定的目录下自动创建程序的框架文件。

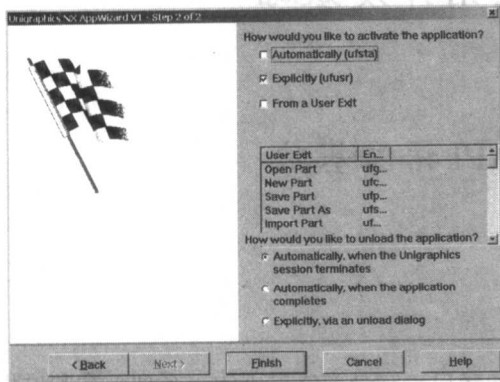


图 1-4 Unigraphics NX AppWizard V1-Step 2 of 2

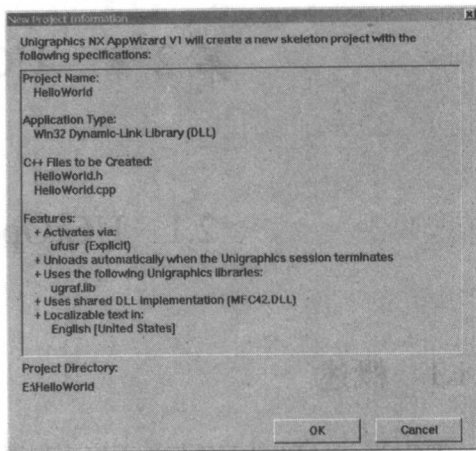


图 1-5 选项信息

步骤 5: 添加代码

打开文件 HelloWorld.cpp，在 ufusr() 函数中添加如下代码：

```
extern "C" __declspec(dllexport) void ufusr( char *parm, int *returnCode, int rlen )
{
    /* Initialize the API environment */
    int errorCode = UF_initialize();
    if ( 0 == errorCode )
    {
        /* TODO: Add your application code here */
        //添加代码
        uc1601("Hello World!",1);           // 弹出消息对话框
        /* Terminate the API environment */
        errorCode = UF_terminate();
    }
    /* Print out any error messages */
    PrintErrorMessage( errorCode );
}
```

编译该程序，系统将在工程路径下的 Debug 文件夹中生成 HelloWorld.dll 文件。启动 Unigraphics NX3，选择菜单命令 File→Execute UG/OPEN→User Function（或按快捷键 Ctrl+U），弹出 Execute User Function 对话框，选择文件 HelloWorld.dll，单击 OK 按钮，出现如图 1-1 所示的消息对话框。

第2章 UG 应用开发基础

2.1 UG/Open API 语法基础

2.1.1 概述

UG/Open API 是 UG 与外部应用程序之间的接口,它是 UG 提供的一系列函数和过程的集合。通过 C/C++语言编程调用这些函数和过程可以实现以下几个功能:

(1) 对 UG 文件及相应模型进行操作,包括 UG 模型的构建、编辑,装配体的建立、遍历,以及工程图纸的创建等。

(2) 在 UG 主界面中创建交互式程序界面。

(3) 创建并管理用户定义对象等。

UG/Open API 函数和过程一般定义在\${UG_BASE_DIR}\UGOPEN 目录下的头文件中。表 2-1 列举了一些常用的头文件名称及其描述。

表 2-1 常用的头文件名称及其描述

头文件名称	头文件描述
uf.h	UG/Open API 的公共类型和函数定义
uf_assem.h	与装配有关的函数和结构定义
uf_attr.h	与部件和对象属性有关的函数和结构定义
uf_defs.h	UG/Open 接口所需数据类型和宏的定义
uf_disp.h	与显示有关的函数和结构定义
uf_draw.h	与制图中截面线和图纸等有关的函数和结构定义
uf_drf.h	与制图中符号、标注和尺寸等有关的函数和结构定义
uf_exit.h	与用户出口有关的函数和结构定义
uf_fam.h	与零件族有关的函数和结构定义
uf_mb.h	与 MenuScript 中对象属性操作有关的函数和结构定义
uf_modl.h	与模型创建、查询有关的函数和结构定义
uf_modl_features.h	与特征建模有关的函数和结构定义
uf_modl_general.h	普通建模函数及数据结构声明
uf_object.h	与 UG 对象及属性操作有关的函数和结构定义
uf_object_types.h	UG 对象的类型定义
uf_part.h	与部件文件操作有关的函数和结构定义

续表

头文件名称	头文件描述
uf_styler.h	与 UIStyler 有关的函数和结构定义
uf_ui.h	与用户界面功能有关的函数和结构定义
uf_view.h	与视图有关的函数和结构定义
uf_wave.h	与 WAVE 功能有关的函数和结构定义
uf_weight.h	与重量管理有关的函数和结构定义

UG/Open API 的应用范围包括定制个性化用户界面、开发在 UG 软件平台上的用户专用软件, 以及开发与其他软件的接口等。

2.1.2 UG/Open API 数据类型

除了 C 语言标准的数据类型外, UG/Open API 还提供了一些自定义的数据类型, 如 `tag_t` 类型、结构类型、联合类型与指针类型等, 它们统一用后缀 “_t” 表示, 且这些数据类型的指针用后缀 “_p_t” 表示。

1. `tag_t` 类型

`tag_t` 类型是 UG/Open API 使用最多的数据类型之一, 它在头文件 `uf_def.h` 中定义为:

```
typedef unsigned int tag_t, *tag_p_t;
```

在 UG 环境中, `tag_t` 是 UG 对象的句柄, 即 UG 对象模型的惟一标识。它是一种不重复的无符号整型数值, 主要用于标识应用程序中的对象, 如部件、草图、曲线、属性和表达式等。UG 应用程序只能访问句柄, 而不能直接访问句柄所指示的实际对象。程序一般通过调用 API 函数获取句柄, 并且在其他 API 函数中使用这个句柄, 以引用它指示的对象。在此过程中, 句柄的实际值对程序来说是无关紧要的。此外, `tag_p_t` 是指向 `tag_t` 数据类型的指针。

以部件对象为例, 下面的代码首先获取工作部件的 `tag_t` 值, 然后获取该句柄所引用部件对象的文件名。

```
tag_t tWorkPart;  
tWorkPart = UF_ASSEM_ask_work_part();  
char sFileSpec[256+1];  
UF_PART_ask_part_name(tWorkPart, sFileSpec);
```

2. 结构类型 (Structure type)

UG/Open API 采用 C 语言的语法定义了一些常用的结构类型, 将相互联系的不同类型

数据封装在一起使用，这些结构类型用后缀 “_s” 表示，例如：

```
struct UF_STYLER_item_value_type_s
{
    int reason;
    const char *item_id;
    int subitem_index;
    int count;
    int item_attr;
    int indicator;
    UF_STYLER_value_t value;
}

typedef struct UF_STYLER_item_value_type_s UF_STYLER_item_value_type_t,
                                         *UF_STYLER_item_value_type_p_t;
```

结构体 UF_STYLER_item_value_type_s 封装了对话框控件的基本属性，如 reason、item_id 和 value 等。UG/Open API 使用关键词 typedef 将该结构类型定义为新类型 UF_STYLER_item_value_type_t，用户可以使用它定义结构变量，然后引用结构体内的数据成员，例如：

```
UF_STYLER_item_value_type_t data;
UF_STYLER_ask_value(dialog_id, &data);
```

3. 联合类型 (Union type)

有时需要将不同数据类型的变量放在同一段内存单元中，这种使不同变量占用同一段内存的结构类型称为联合类型。联合类型与结构类型的定义形式相似，但含义不同。联合类型变量所占的内存长度等于其内部最长数据成员的长度，而结构类型变量所占的内存长度则是其各数据成员所占内存长度之和。

UG/Open API 采用 C 语言的语法定义了一些常用的联合类型，使用后缀 “_u” 表示。以联合类型 UF_STYLER_value_u 为例，下面给出其详细定义：

```
union UF_STYLER_value_u
{
    char    *string;           /* A string value */
    char    **strings;        /* An array of strings */
    int     integer;          /* An integer value */
    int     *integers;        /* An array of integers */
    double  real;             /* A double value */
    double  *reals;           /* An array of doubles */
}
```



```

UF_UI_selection_p_t selection;      /* A pointer to a selection structure */
UF_STYLER_notification_p_t notify; /* A pointer to the notification structure */
};
typedef union UF_STYLER_value_u UF_STYLER_value_t;

```

联合体 `UF_STYLER_value_u` 封装了对话框控件的属性值, 如 `string`、`integer` 和 `real` 等。UG/Open API 使用 `typedef` 关键词将该联合类型定义为新类型 `UF_STYLER_value_t`, 它是 `UF_STYLER_item_value_type_t` 中的数据成员 `value` 的类型, 例如:

```

UF_STYLER_item_value_type_t data;
data.value.integer = 1;

```

4. 枚举类型 (Enum type)

枚举类型将变量的值一一列举出来, 变量的值只限于列举出来的值的范围。UG/Open API 采用 C 语言的语法定义了一些常用的枚举类型, 使用后缀 “_e” 表示。以枚举类型 `UF_ASSEM_load_status_e` 为例, 下面给出其详细定义:

```

enum UF_ASSEM_load_status_e
{
    UF_ASSEM_ok,
    UF_ASSEM_suppressed,
    UF_ASSEM_not_loaded,
    UF_ASSEM_instance_suppressed
};
typedef enum UF_ASSEM_load_status_e UF_ASSEM_load_status_t;

```

枚举类型 `UF_ASSEM_load_status_e` 表示部件在装配体中的状态。UG/Open API 使用 `typedef` 关键词将该枚举类型定义为新类型 `UF_ASSEM_load_status_t`, 例如:

```

char sPartPath[] = "E:\\Example\\part.prt";
tag_t tPrt = NULL_TAG;
UF_PART_load_status_t error_status;
UF_CALL(UF_PART_open( sPartPath, &tPrt, &error_status ));

```

2.1.3 UG/Open API 函数

从本质上说, UG 应用开发就是通过 UG/Open API 函数 (简称 API 函数) 的集成与组合来实现相应的功能, 因此 UG/Open API 函数在 UG 应用开发中至关重要。