

张宏伟 (深圳联想电脑公司 QDI 研发部 518026)

# MMC 概述

摘要: 本文主要介绍 MMC 管理系统及 Snap-in 插件的系统框架、和开发性研究。

关键词: MMC ISV Snap-in 插件 特性页 Taskpads

## 1 前言

微软管理控制台 MMC(Microsoft Management Console 以下简称 MMC)是由微软公司开发的一种管理控制软件。它通过向独立软件开发商 ISV(Independent Software Vendor)提供软件开发接口(SDK),使独立软件开发商能够很方便地在 MMC 基本框架下开发出管理应用软件,从而使管理软件在风格、构架、应用接口等方面都统一起来,使管理软件在扩展性、易维护性和开发的简便性方面都具有较大的突破。

MMC 实际上是一种运行环境,这种环境作为 Snap-in (由 Microsoft 或 ISV 提供的,往往包括管理特性)的载体,使 Snap-in 插件能够将其管理特性或结果呈现出来的一种多文档界面(MDI)。这样 MMC 就形成了 MMC 界面作为客户器、Snap-in 插件作为服务器的客户器/服务器(Client/Server)模式。

分,用于 MMC 与用户进行交互;服务器由 Snap-in 管理器和两个 Snap-in 插件组成系统基本构架,Snap-in 插件管理器管理各个不同的 Snap-in 插件,同时也负责将各种配置存储到 MSC 文件中。它和 Snap-in 插件都是由开发商提供。对于用户来说,服务器层是完全透明的。用户通过打开 MYTOOL.MSC 文件,将路由监视器和事件查看器这两个 Snap-in 插件调入 Snap-in 插件管理器,接着用户通过菜单、工具栏、向导或帮助等使 Snap-in 插件管理器调用 Snap-in 插件接口得到结果后,将结果显示到 MMC 界面的结果块或范围块中。

## 2 系统构架

MMC 模型的系统构架如图 1 所示:

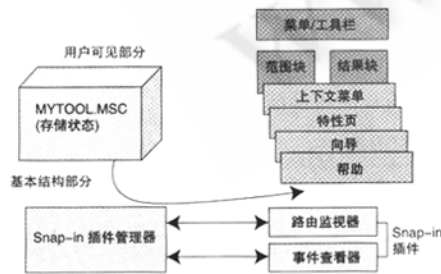


图 1

如图所示,MMC 采用了客户器/服务器(Client/Server)模式,客户器由系统提供的 MMC 界面组成,是用户可见部

## 3 MMC 界面的组成

一个完整的 MMC 界面如图 2 所示:

MMC 界面由父框架和多文档界面子窗口两部分组成。

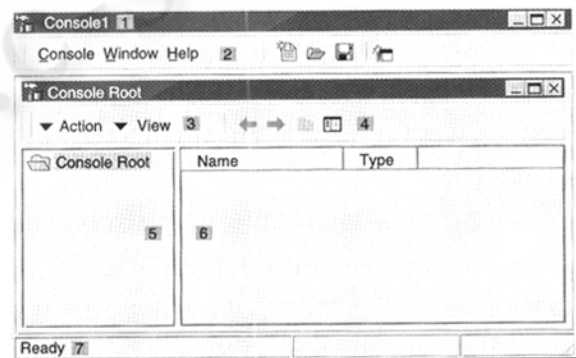


图 2

### 3.1 父框架

父框架包括:

- (1) MMC 标题栏(图中标号 1 所示):用于显示当前 MMC 文件的名称;

(2) MMC 主菜单和工具框(图中标号 2 所示):由左边的 MMC 菜单和右边的与菜单相对应的快捷工具栏组成,MMC 菜单分三项:控制台(Console)管理菜单,窗口(Window)管理菜单和帮助(Help)菜单;

(3) 状态栏(图中标号 7 所示):显示 MMC 系统状态信息。  
3.2 多文档界面(MDI)子窗口

MDI 子窗口包括:

(1) MDI 子窗口菜单(图中标号 3 所示):它包括作用菜单(Action),Snap-in 插件提供的各个命令形成的下拉菜单可以放在这里,以及视图(View)菜单,它为用户提供不同的视图选项;

(2) MDI 子窗口工具栏(图中标号 4 所示):由与 MDI 子窗口菜单相对应的快捷工具图标组成;子窗口菜单和子窗口工具栏合称为子窗口命令栏;

(3) 范围块(控制台树,图中标号 5 所示):显示可由用户实施系统管理的内容,由命名空间形成的树性结构组成,每一个树节点代表一个可管理的对象,任务或视图;

(4) 结果块(图中标号 6 所示):用于显示对范围块中所选系统节点进行管理配置后形成的结果,它们可能是文件夹的内容,也可能是网页,基于 ActiveX 控件的与管理有关的视图。

对于具体的应用,MMC 界面也可能只含有其中几项。

## 4 Snap-in 插件

每一个 MMC 工具都是由作为服务器的 Snap-in 插件小工具组成的实例构成,每一个 Snap-in 插件代表了一个单位的管理特性,也就是说 Snap-in 插件是管理集合的最小单位。从技术上来说,每一个 Snap-in 插件是一个在 MMC 进程环境下执行特定管理任务的 OLE 进程内(In-proc)服务程序。

一个或多个 Snap-in 插件可以组装成由 MSC(Management Saved Console)作为扩展名的文件,如图 2 所示的 MYTOOL.MSC 文件。用户可以通过打开这个文件建立 Snap-in 工具。用户也可以将这个文件通过 Email,FTP 等工具或者用磁盘传送给另一台计算机,在这台计算机上,用户即使没有安装相应的 Snap-in 插件,也可以通过打开 MSC 文件,在 MMC 上执行管理功能。因为当系统装载 MSC 文件时,MMC 会自动通过网络下载相应的 Snap-in 插件。这样用户可以方便地将不同的 Snap-in 插件组装成适合自己使用的 MSC 文件了。因此 Snap-in 插件相对于别的管理工具就具有面向任务,集成性的优点。

Snap-in 插件还可与 Java, IE, ActiveX 一起工作。

### 4.1 Snap-in 插件的分类

Snap-in 插件分两种:主 Snap-in 插件和扩展 Snap-in 插件。主 Snap-in 插件提供单一管理功能,且不依赖于其他的 Snap-in 插件。扩展 Snap-in 插件只能扩展特定的节点类型,它由生成它的父 Snap-in 插件激活。

有许多 Snap-in 插件同时支持这两种方式。

### 4.2 Snap-in 插件的菜单和工具栏

MMC 的菜单和工具栏由两部分组成,MMC 提供的菜单,工具栏以及 Snap-in 插件提供的菜单和工具栏。

对于每一个节点的上下文菜单可以由控制台提供的,由主 Snap-in 插件提供和由扩展 Snap-in 插件提供的。如图 3 所示:

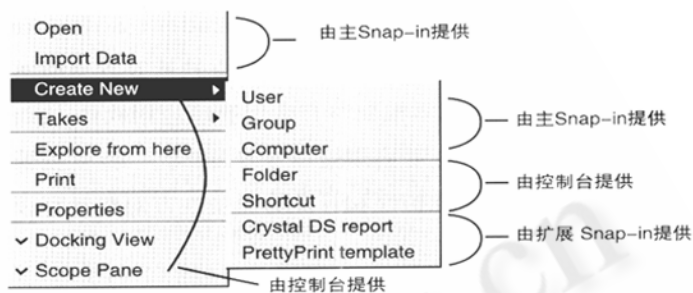


图 3

菜单的间隔栏不能由 Snap-in 插件提供。间隔栏代表了菜单项的不同区域, Snap-in 插件可以在 Create New 和 Tasks 中加入下拉菜单。菜单项的顺序是由它们的加载顺序决定的,用户不能安排它们的顺序。

### 4.3 特性页扩展

目前的 MMC 不支持有模式对话框和无模式特性页。因此,MMC 推荐使用无模式特性页。无模式特性页必须使用标准外壳大尺寸(252\*218)。这是为了保证 MMC 界面的统一性,美观性和简洁性。

### 4.4 Taskpads

MMC 的功能还可以通过它提供的内置的对 Taskpads 功能的支持加强, Taskpads 可以通过使用 HTML 和 ActiveX 技术来编写。

MMC1.1 版不支持在线帮助。

## 5 Snap-in 插件的创建和设计

实际上,每一个 Snap-in 都可以在注册表中的 HKLM

\\ Software \\ Microsoft \\ MMC \\ Snapins \\ 找到。

在 Snap-in 插件的设计中,控制台树主要用于显示可配置的对象和对象载体,而 Action 用于使用工具栏,菜单和 taskpad 的对象。

控制台树可由三种节点构成:载体节点(可以包含许多叶节点),可视节点(如网页页面)和叶节点(在结果块中显示)。

控制台提供了大量的面向任务的服务,但是这些服务都是通过菜单,结果块,和 taskpad 来组织的,控制台树只包含有对象和对象载体,而不能使用单独的任务,单独的任务可以使用 taskpad 提供的基于任务的用户界面。

### 5.1 控制台接口

用于添加或删除控制台树的栏目。接口 IheaderCtrl 被 Snap-in 插件用于在结果块中添加或删除栏,或改变栏的名称或栏的宽度。这个接口是由 MMC 的节点管理器来实现的。Snap-in 还使用接口 IImageList 来使用户能够向控制台树或结果块添加图标或图形。IResultData 接口则使用户能够添加,删除,寻找或者修改结果块中的栏目,同时这个接口还使用户能够改变结果块的外观造型,它也是由 MMC 的节点管理器来实现的。当结果块要显示虚列表时,系统会调用 IResultOwnerData 接口来增强虚列表特性。接口 ISnapinAbout 则用于显示 Snap-in 插件的名字,版本和版权信息。同时,MMC 也可利用这个接口来从 Snap-in 插件得到静态文件夹的图标。

### 5.2 特性页接口

IPropertySheetCallback 是基于 COM 的接口,用于向特性表中添加特性页,其他特性页操作有关的接口还有: IPropertySheetProvider, IExtendPropertySheet, 函数有 MMCPropPageCallback 和 MMCPPropertyChangeNotify。

### 5.3 菜单接口

对于菜单的操作 Enable, disable 和隐藏使用 ICosoleVerb 接口,它可以对普通菜单项中的剪切,删除,粘贴,拷贝,重命名,刷新和打印项进行操作。此外, IContextMenuProvider 接口和 IContextMenuCallback 接口用于创建新的菜单项。前者由 Snap-in 插件实现,后者由节点管理器实现。IMenuButton 接口使用户能够添加和管理 Snap-in 插件菜单按钮。

### 5.4 工具栏接口

创建和管理控制块的接口是 IControlBar,创建工具栏的接口是 IToolBar,它们都是由节点控制器来实现的。IExtendControlbar 则是由用于进程内(in\_process)服务的

DLL 提供的对象链接与嵌入(OLE)接口。

### 5.5 Help 接口

MMC1.1 版不支持帮助,在 MMC1.2 版中加入了 Snap-in 插件提供的帮助的支持,接口 ISnapinHelp 用于实现对 CHM 帮助文件(微软标准帮助文件)的支持,这个接口由 Snap-in 插件实现。IDisplayHelp 接口用于使帮助文件能够在结果块显示出来,这个接口由 MMC 实现。

## 6 应用

在实际应用中往往利用活动模板库(简称 ATL)来进行编程,主要原因是 ATL 速度快,代码量小,支持多种接口,自动生成注册表项,有添加 Snap-in 插件类的向导等优点。

以下以微软的 Visual C++ 为例,说明建立 MMC Snap-in 插件的过程。首先,利用 ATL COM AppWizard 建立一个新项目(如 MySnapIn),接着在服务器类型中选择动态链接库(DLL)类型,完成后,在插入菜单(Insert)中选择新 ATL 对象,再在对象(Objects)中选择 MMC Snapin 即可,这样系统自动生成描述接口的 IDL 文件以及四个类和两个结构:

类 CmySnapIn: 该类由类 IComponentData 派生而来,较重要的成员变量有 m\_pNode。

类 CMySnapInComponent: () 该类由类 IComponent 派生而来,较重要方法有 Notify(),它相当于 CView 类,实现菜单,Taskpad 等功能,支持结果块。

类 CMySnapInAbout: 该类由类 ISnapInAbout 派生而来。

类 CMySnapInData: 该类由类 CsnapInItem 派生而来,较重要方法有 Notify(),代表范围块每一个节点,支持范围块。

结构 ScopeDataItem: 包含范围块的各项数据。

结构 ResultDataItem: 包含结果块的各项数据。

通过编译所生成的 dll 文件由系统自动注册,然后运行 MMC,在 Console 菜单下选择 Add/Remove Snap-in...,选择你的 Snap-in 类型,再选择添加后,会发现刚编译的 Snap-in,将其加入到 MMC 中就可应用这个 Snap-in 了。接着可以添加别的接口和类,在 MMC 中形成各种功能。■

### 参考文献

- 1 Microsoft Management Console Snap-In Author's Guide White Paper Microsoft.
- 2 Microsoft NT Server White Paper.