

CONTAX IVR V. 2.1 技术手册

用户指南

第一版

(C) 2002-2004 技电股份有限公司

版权所有



第一章：CONTAX IVR 是什么？	2
1.1 IVR 是什么？	2
1.2 为什么 CONTAX IVR 比其它同类产品更出色？	3
第二章：安装 CONTAX IVR	4
2.1 系统需要	4
2.1.1 安装步骤	4
2.1.2 第一印象：CONTAX IVR	8
第三章：主窗体	11
3.1 单元格、控件、属性、和变量	12
3.2 旁边的工具条	13
流程控件单元格	14
电话控件单元格	15
媒体控件单元格	16
3.3 顶端工具条	17
3.4 着手开始：探索 FLOW DESIGNER	18
3.4.1 语音播放单元格图标	18
3.4.2 单元格图标的选取与取消选取	19
3.4.3 单元格图标的删除	20
3.4.4 在两个单元格之间放置连结路径	20
3.4.5 删除两个单元格之间的连接线	21
3.4.6 移动选定的图标	22
3.4.7 剪切/复制单元格图标及连结线	22

3.4.8 粘贴单元格图标及连结线.....	22
3.5 属性表窗体.....	23
3.5.1 下拉式单元格列表.....	25
3.6 变量表窗体.....	25
第四章：你的第一个应用程序.....	27
4.1 把它分割成细小的部分.....	28
4.2 投入到设计中.....	28
4.2.1 应用一：问候.....	29
4.2.2 保存制作完毕的应用程序.....	32
4.2.3 验证：模拟你的电话流程.....	32
4.2.4 声音信息的录制.....	34
4.2.5 使用文本朗读(Text-To-Speech).....	35
第五章：选择和结果.....	37
5.1 条件和循环.....	37
5.1.1 应用二：来话自动转接.....	37
5.1.2 应用三：用不同方式做同一件事.....	42
5.1.3 资料输入(GetKeys)对比按键输入(GetDTMF).....	43
第六章：子流程：简化复杂的事情.....	52
6.1 什么是子流程？.....	52
6.1.1 应用四：具有语音邮件功能的来话自动转接程序.....	53
6.1.2 保存与命名子流程.....	54
6.1.3 生成一个子流程库.....	56

第七章：变量和表达式	57
7.1 变量表窗体.....	57
7.1.1 在变量表中增加变量.....	57
7.1.2 分配一个变量.....	58
7.1.3 使用表达式编辑器.....	59
7.1.4 局部变量与全局变量.....	61
7.1.5 应用五：来话自动转接(AutoTransfer)中的其它变量.....	61
第八章：数据库的使用	67
8.1 什么是数据库？.....	67
8.2 创建一个数据库.....	68
8.3 掌握数据库的功能.....	70
第九章：和传真、E-MAIL、短信息协同工作	73
9.1 传真发送.....	73
9.2 应用六：传真自动应答(也称按需传真 FAX ON DEMAND).....	74
9.3 接收一个传真.....	76
9.4 使用电子邮件(E-MAIL).....	78
9.4.1 应用七：发送邮件.....	78
9.4.2 接收电子邮件.....	79
9.4.3 应用八：邮件接收.....	81
9.5 使用 SMS(短消息系统 SHORT MESSAGING SYSTEM)发送信息.....	83
9.6 WEB 集成.....	85
9.7 其它控件单元格.....	86
9.7.1 多方会谈.....	86

9.7.2 线路交换.....	87
第十章：程序的运行.....	88

第一部分 介绍

CONTAX IVR 应用程序开发手册

欢迎使用 CONTAX IVR 一个面向客户服务并整合了语音、传真、电子邮件和短信息的新一代交互式语音应答系统。本手册可以使你在之前没有任何编程经历的情况下掌握如何使用 CONTAX IVR。另一方面，对于经验丰富的程序员，本手册同样有助于他们快速理解和熟悉 CONTAX IVR。包括设计、测试和配置 CONTAX IVR 应用程序所需的基本知识和规则。

本手册遵循自然技能学习过程的步骤，即从你现在所掌握的大多数技能，比如如何朗读，如何骑自行车，如何驾驶及如何游泳中得出的学习方法。因为有人帮助你，所以你学会了这些技能。有人给你念字母和单词，并告诉你单词的意思。你反复重复这些词直到可以正确发这些词的音。有人帮助你骑车，在后面推你并告诉你踩踏板直到你能向前骑而不摔倒。

相似地，本手册将带领你经过如下的步骤：在前面的章节中使用 CONTAX IVR 的基本编程组件创建简单的应用程序；一步一步地向您介绍系统范例程序是如何创建的。当然，随着你对相关组件及过程的进一步了解，这边的介绍将变得不够具体。

你将会执行这些应用程序，并对它进行测试以发现它做了什么，以及是什么使它不能完成某功能。当你尝试了所提供的全部应用程序之后，你就会掌握设计 CONTAX IVR 应用程序的基本技能和知识。

第一章：CONTAX IVR 是什么？

1.1 IVR 是什么？

IVR 是指交互式语音应答(Interactive Voice Response)；它是一个自动电话信息系统，运用固定语音菜单及来自数据库中的实时数据对电话呼入者进行语音答复。呼入者通过按话机上的数字键来作出回应。许多先进的 IVR 系统允许来话者用词或短语来与系统交互，这种技术被称为自动语音识别 Automatic Speech Recognition(ASR)。

IVR 系统可以 24 小时全年不间断的为来话者提供所要的信息。IVR 系统也被用来减少人力成本，它免去了人工不停地去回答简单、重复性的问话。

大多数 IVR 系统运行在装有特殊语音卡的个人电脑(PC)上，由这些语音卡处理来自电话界面的语音信号。语音卡包含有数字声音处理(DSP)芯片。IVR 系统也可以与局域网(LAN)及广域网(WAN)相联。

基于 PC 架构的 IVR 系统通常与公司内部电话交换机(PBX)的一路或多路模拟分机线相联。一个基于 WEB 架构的 IVR 系统也可以联接到 Internet。语音卡可以和 PBX 进行交互操作，进行例如发送 FLASH 信号、号码外拨、检测 hang-up 信号、侦测呼叫终止等操作。语音卡也可以接听呼叫、启动呼叫与终止呼叫。

1.2 为什么 CONTAX IVR 比其它同类产品更出色？

目前市场上绝大多数的 IVR 系统仅能支持语音方式，一些更高级的 IVR 系统可以同时支持 email。然而 CONTAX IVR，真正的新一代 IVR 系统，允许用户所创建的应用程序中包含语音、传真、email、简讯(SMS)，并可以通过新颖、高效的方式对数据进行无缝的访问。 CONTAX IVR 同时提供文本朗读功能 (text-to-speech)，允许你的 IVR 系统对来话者“大声地朗读”文档内容。

此外，CONTAX IVR 提供了简单易用的可视化工具，简化了程序设计者创建应用程序的过程，便于管理者监测、管理 IVR 应用程序。如果你是个程序开发者，你仅仅只需将单元格从工具栏拖到设计窗体中，点击单元格图标上的指针连结各个单元格图标，就可以组成所要的电话流程图。经过如此简单的步骤之后，你可以运行仿真器，对你所设计的电话流程进行检验、跟踪、除错及修改。

你可以根据需要轻松地配置 CONTAX IVR，不论其是基于应用程序还是基于 WEB，或者同时基于两者。在将 IVR 应用程序复制到目的服务器前，你可以在一台单独的计算机上通过流程设计器完成流程的设计、修改及测试。

CONTAX IVR 支持 TAPI (Telephony API) 兼容的语音卡。TAPI 是 Microsoft 及 Intel 开发的适用于计算机电话集成技术的应用程序编程界面。CONTAX IVR 也可以和用户交换机系统 (PBX) 协同工作，提供电话的转接和多方会谈功能。

第二章：安装 CONTAX IVR

2.1 系统需要

运行 CONTAX IVR 需要以下的环境：

- Intel Pentium III 450MHz / 128MB 内存 / 10MB 硬盘空间
- Microsoft Windows2000(Professional 或 Server)及 IIS 5.0 或以上
- 具有 TAPI 驱动的模拟语音卡

具有串行口驱动的传真调制解调器 -- 针对传真应用

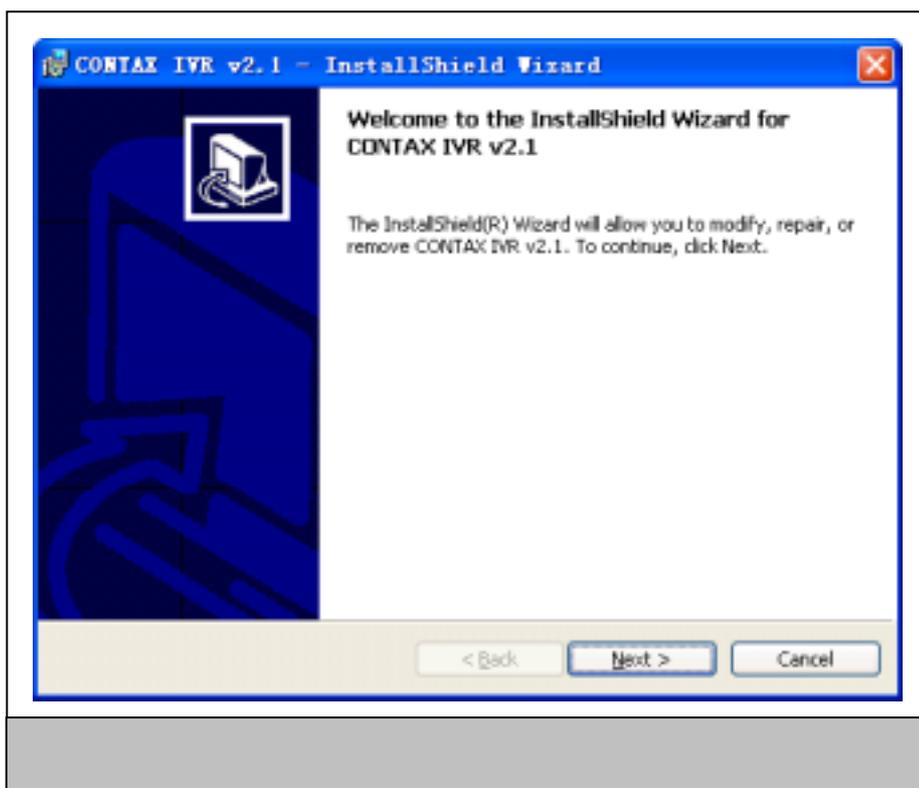
Microsoft SAPI 5.0(Windows XP 标配，CONTAX IVR 光盘中附带)

PBX 系统 -- 用于电话转接及会议功能

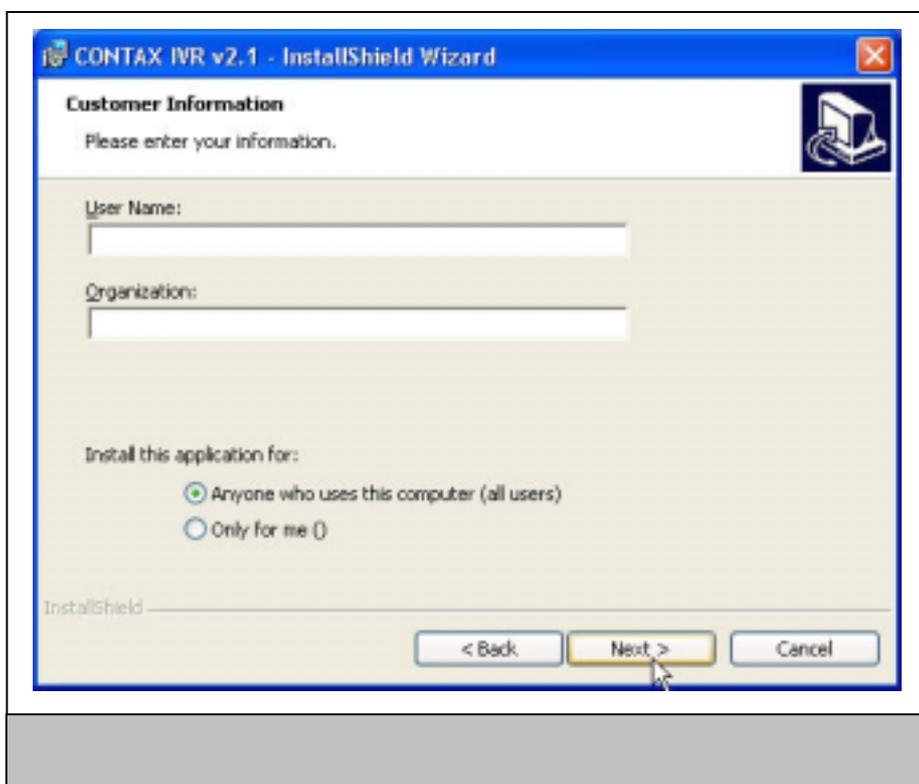
2.1.1 安装步骤

将 CONTAX IVR 光盘放入光驱内。

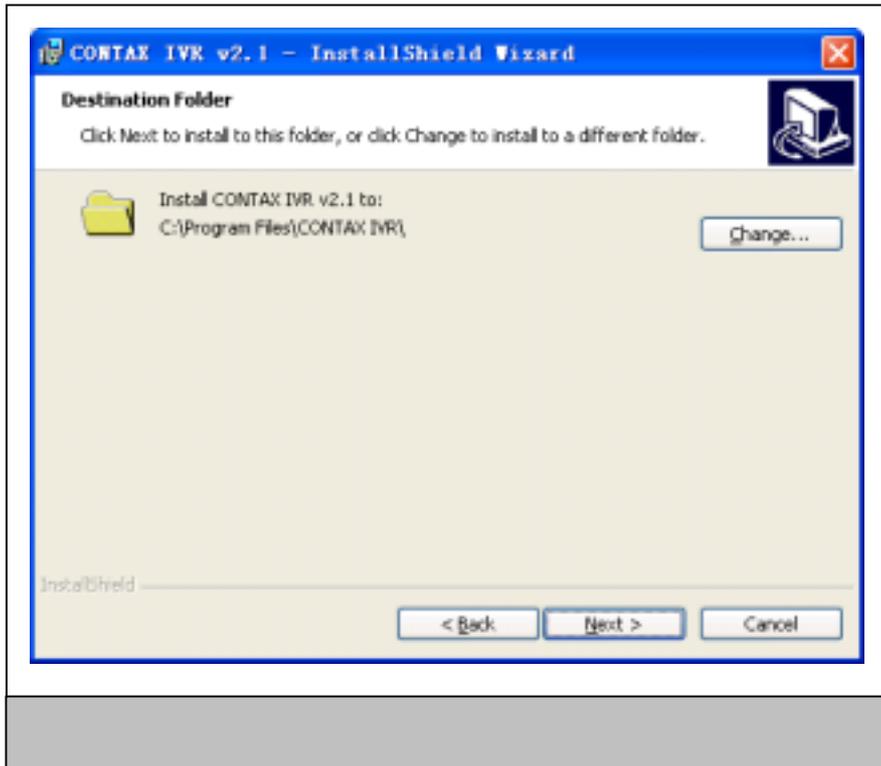
双击 setup.exe 开发安装，出现 CONTAX IVR 2.1 版安装向导，按提示依次完成安装。



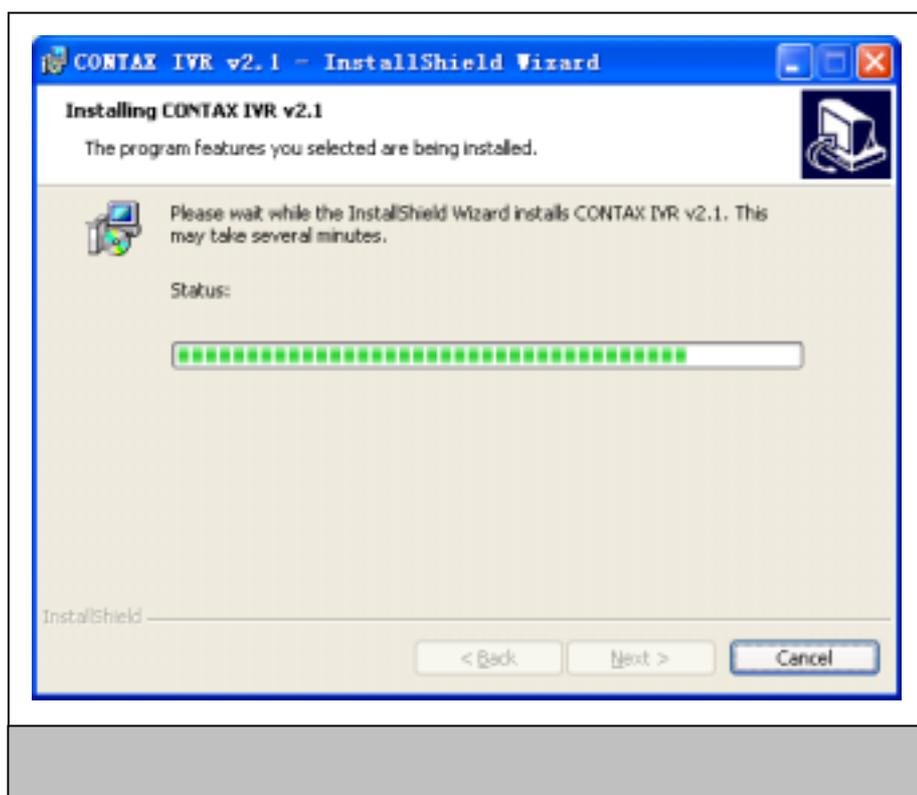
单击下一步，你会看到如下图形：



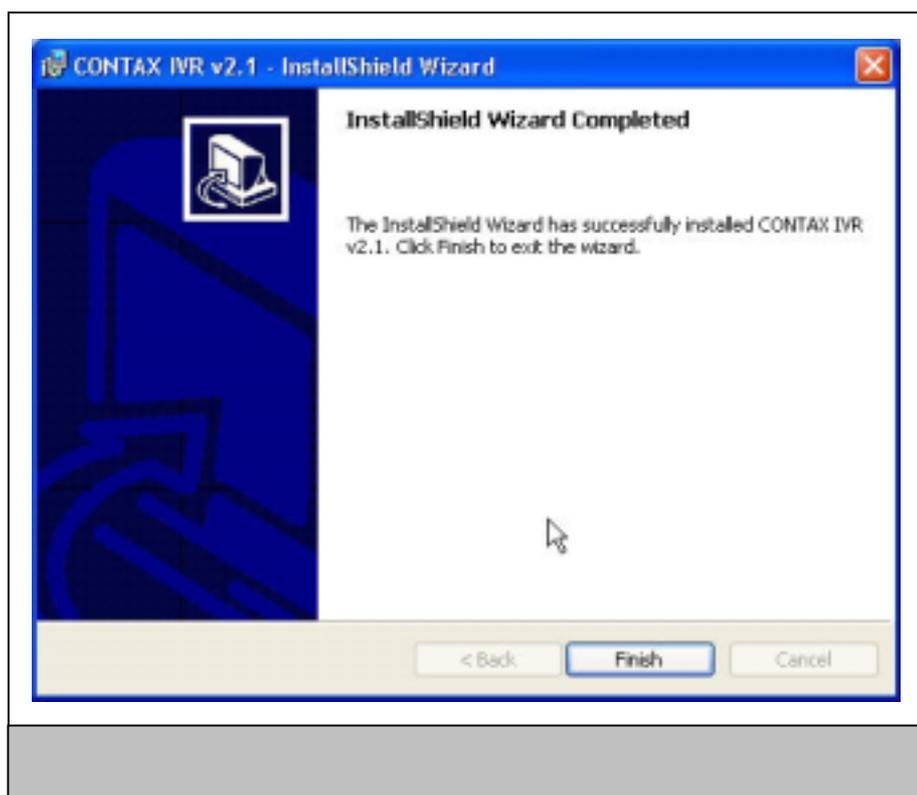
在对应的框中输入你的名字及组织或公司名称。下一步，选择安装路径，CONTAX IVR 的默认安装路径为“C:\Program Files\ContaxIVR 2.1\”，如果按“Change”，可以安装于你所选择的目录中。



路径选择完毕，单击“下一步”，程序开始安装，如下图：



安装完成，出现下图：

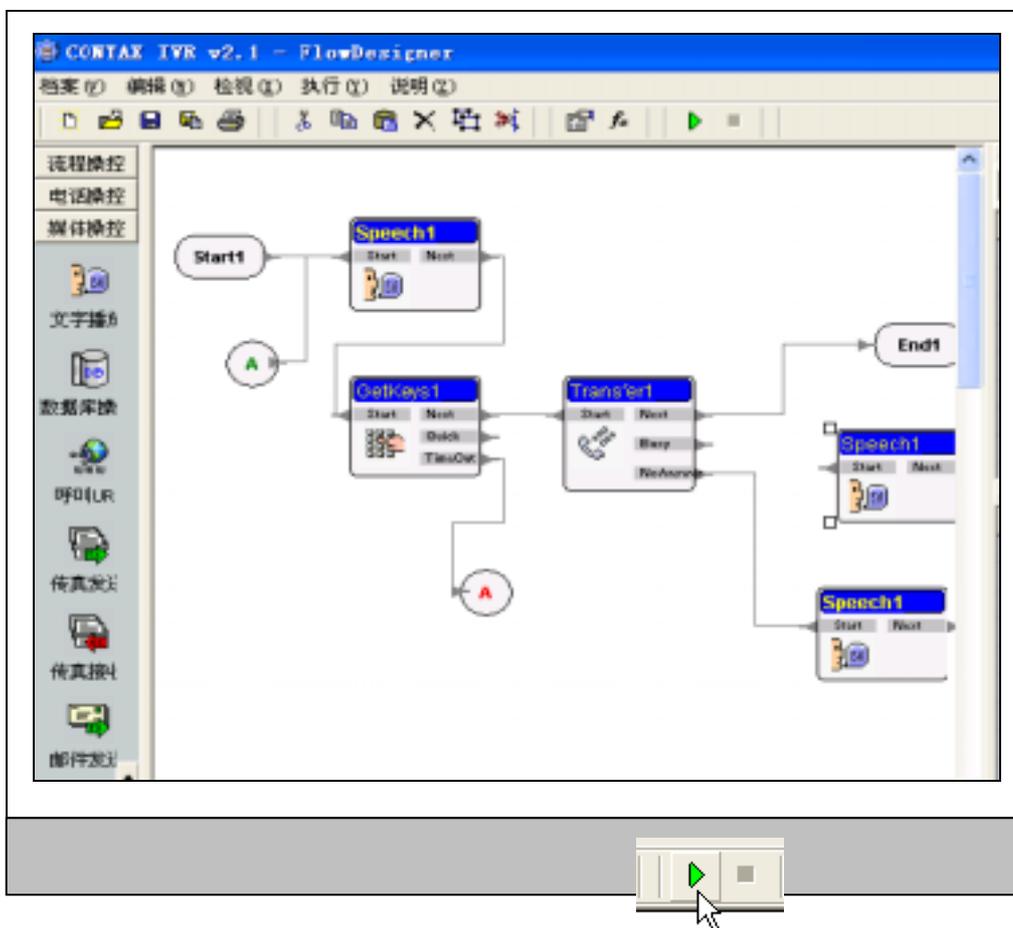


2.1.2 第一印象：CONTAX IVR

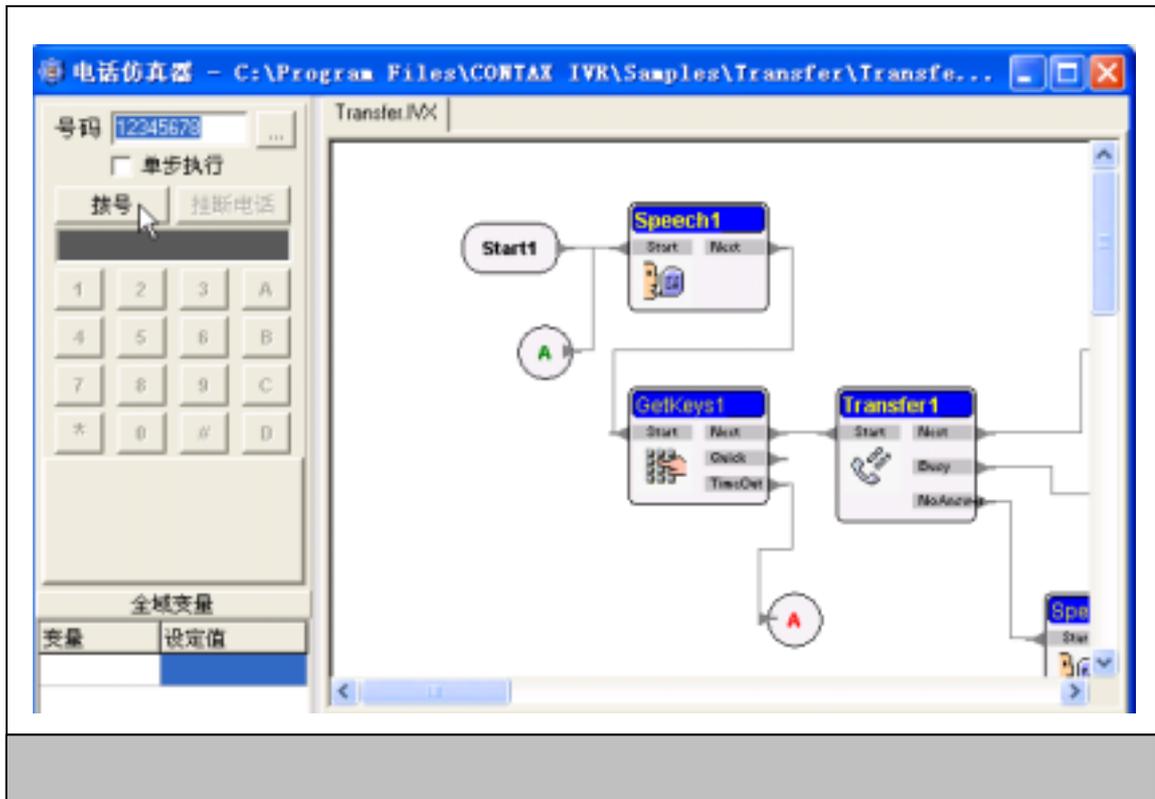
为了使你对 CONTAX IVR 的工作方式有一个更好的概念，我们可以测试安装光盘中的例子应用程序。

首先，运行 CONTAX IVR Flow Designer，单击**开始 > 程序 > CONTAX IVR > Flow Designer**。在 Flow Designer 的下拉菜单中单击**档案|开新档案**。出现对话框，打开 CONTAX IVR 的安装路径，选择 **Transfer** 文件夹，单击**打开**。你可以看到一个名为 **Transfer.IVX** 的文件，选择该文件，单击**打开**。

当 *Transfer.IVX* 文件被打开时，如下图：

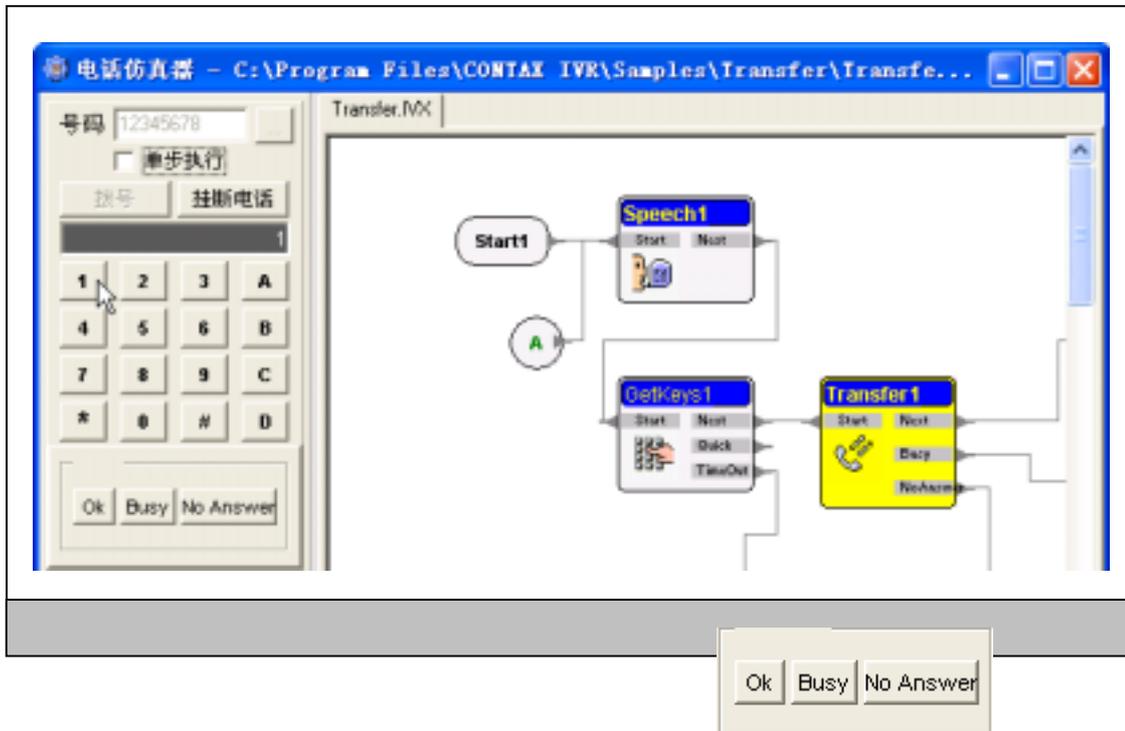


现在单击**执行**按钮--顶端工具栏上的绿色三角按钮。**电话仿真器**程序将被打开(如下图)。



按拨号按钮，执行 Transfer.IVX 电话流程，检测到一个呼叫从一个通道中拨打进来。你会听到问候语：“Welcome to Eletech. Please dial the extension number.” 注意，当消息正在播放时，Speech1 模块变成黄色，说明是当前活动模块。当一个模块变成黄色时，你就可以知道此模块正在执行中。

如果你什么也不做，问候语将重复。按拨号盘中的任意一个键，例如数字 1。



如果你按了拨号盘中的 **Busy** 按钮，你会听到下面的问候语：“Sorry, line is busy now.” 重复最开始的话，这次按数字 1 之后按 **Ok** 按钮，呼叫结束，表明呼叫已被转接到所要的号码。

这就是 CONTAX IVR 电话流程处理呼叫的方式。在以下章节中你将学会如何根据特殊的需要使用 Flow Designer 制作电话流程。

第二部分 CONTAX IVR 流程设计器

第三章：主窗体

单击 开始 -> 程序 -> CONTAX IVR->FlowDesigner。出现 Flow Designer 主窗体。你会看到两个工具条。一条沿着顶端，另一条沿着左边界。

主窗体内共有三个子窗体：设计窗体，属性表窗体，变量表窗体。

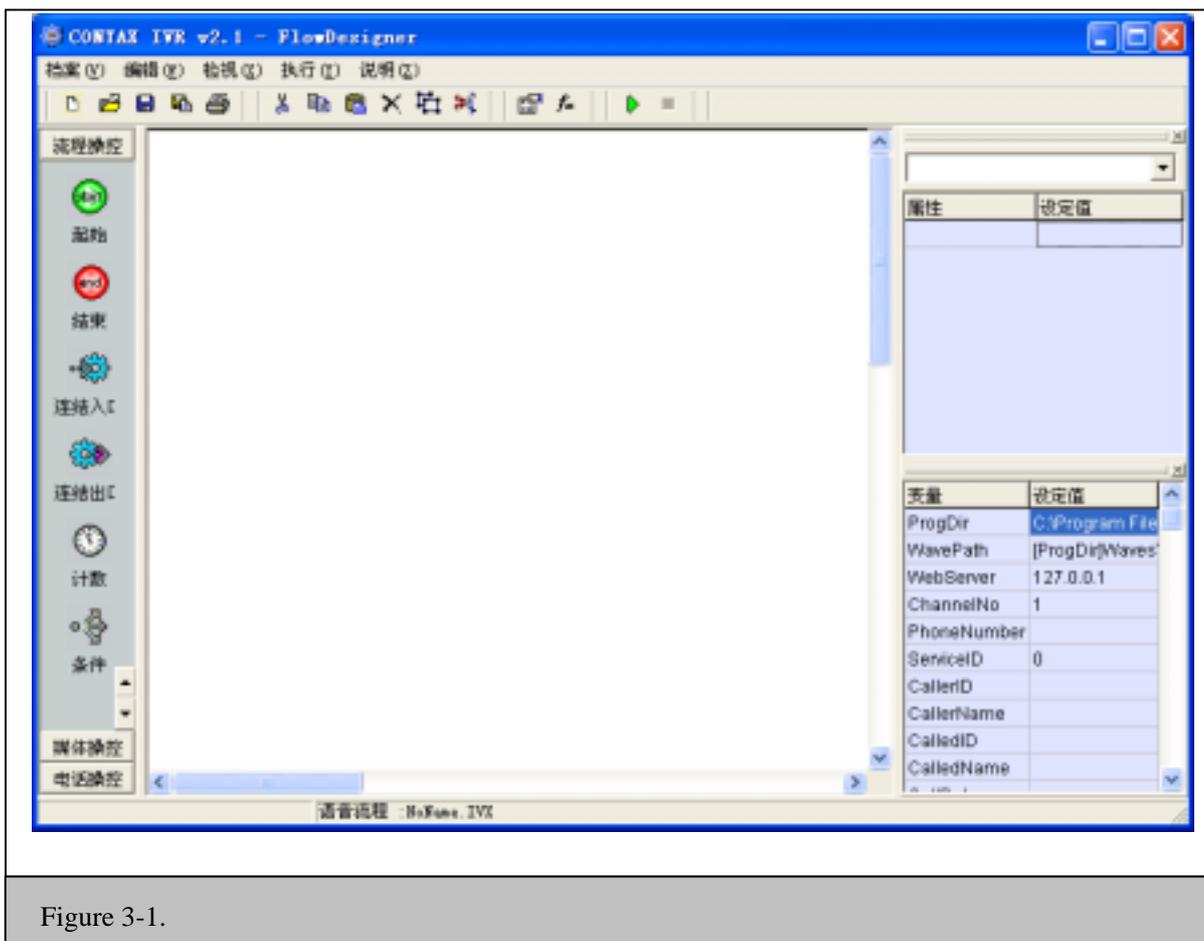
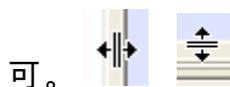


Figure 3-1.

每一个窗体的大小都可以调整，你只要移动鼠标到两个窗体的边界处，光标会变成如下之一的状态，拖动边界线到所要的位置就可。



3.1 单元格、控件、属性、和变量

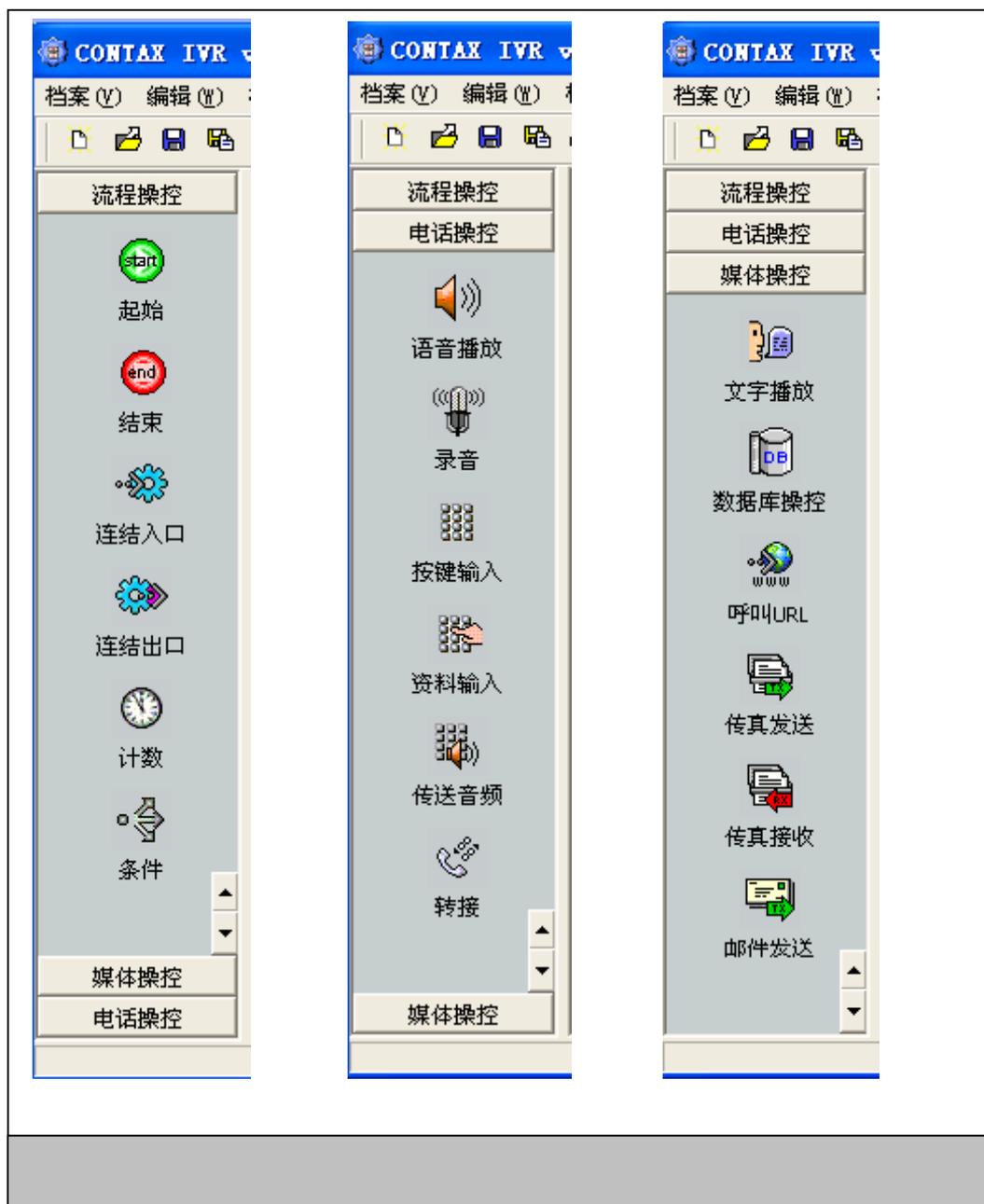
如果你已经使用过 Visual Basic，那么就会感到 IVR 非常地熟悉。在 CONTAX IVR 中，每一个**单元格**都用一个单独盒子表示，它是一组待定义的指令。CONTAX IVR 提供了大量的单元格，可以完成从播放 .WAV 文件，接受用户按键输入，到发送传真和电子邮件的所有功能。单元格也被称为控件，这两个术语将会交替出现在本手册中。

每个单元格包含许多属性，属性是指单元格的特征，例如单元格的名称、播放的 .WAV 文件名等。通过属性设置你可以根据你的特殊需求定义相关的特征值，其它特征值保持不变，这样大大简化 IVR 应用程序的创建过程。如此，创建 IVR 应用程序的主要工作就是选择单元格，连结单元格图标，根据需要改变属性值。

变量是存放信息的容器，在后面的章节“**变量及表达式**”中你会对变量了解得更多。

3.2 旁边的工具条

主窗体左边的工具条为设计者提供了可选的工具单元格。



单元格分为**流程控件**、**电话控件**及**媒体控件**三组。流程控件单元格用于决定 IVR 应用程序从哪里开始执行。电话控件单元格用于与电话的交互。媒体控件单元格可以在你的 IVR 应用程序中加入语音、传真、简讯 SMS、电子邮件及数据库功能。主窗体中默认显

示 7 个 **流程控件** 单元格，如上图左边所示。

单击 **电话控件** 按钮可切换到 **电话控件** 单元格，要看到更多的电话控件单元格，只要单击工具条右下角的向下箭头即可，如上图中间所示。

单击 **媒体控件** 按钮可切换到 **媒体控件** 单元格，如上图右边所示。

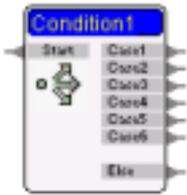
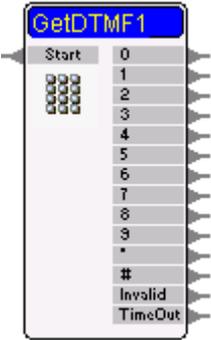
单击 **流程控件** 或 **电话控件** 按钮可切换回 **流程控件** 单元格或 **电话控件** 单元格。

你可以在工具条中单击选择其中一个单元格，移动鼠标指针到设计窗体中，再次单击鼠标将单元格放置于设计窗体内，出现单元格图标，其左上角为接入节点位置。

下表列出了工具条中所有的单元格按钮、相应的单元格图标及功能描述：

流程控件单元格

单元格	按钮	图标	功能描述
起始			开始一个电话流程
结束			结束一个电话流程
连结入口			同一流程中跳转到相同名字的 连结出口
连结出口			同一流程中连结来自相同名字的 连结入口
计数			对循环计数

条件			按条件执行
子流程			同一通道线路上跳转到子流程(即另一个电话流程)
电话控件单元格			
语音播放			播放一段语音
录音			将语音信息录成 .wav 文件
按键输入			接收来自来话者的单个电话按键输入(DTMF).
资料输入			接收来自来话者的一组电话按键输入(DTMF)
传送音频			传送 DTMF 音频信号

转接			通过 PBX 将电话转接到另一分机
多方会谈			通过 PBX 将电话联接到第二路分机
打电话			通过在一新线路上拨打电话执行新的电话流程
线路交换			联接两个通道线路

媒体控件单元格

文字播放			将文字转为语音并向来话者播放
数据库控件			执行数据库查询操作
呼叫 URL			根据已设置的 URL 连结至 WEB 页面
传真发送			发送一个传真

传真接收			接收一个传真
邮件发送			发送一个电子邮件
邮件接收			接收一个电子邮件
发送简讯			传送一个简短讯息(文字)

3.3 顶端工具条



主窗体顶端的工作条中有许多的按钮。

以下按钮与文件操作有关：

	新建一个 IVR 文件
	打开一个已存在的 IVR 文件
	保存当前 IVR 文件
	以新的文件名将当前的 IVR 文件另存
	将 IVR 电话流程设计图作为文档打印输出

以下按钮用于对当前设计的应用程序进行编辑：

	剪切所选单元格图标及其属性、连结路径
---	--------------------



复制所选单元格图标及其属性、连结路径



在原单元格图标右下 20 像素的位置粘贴所复制单元格图标及其连结路径，并同时粘贴单元格的属性表。



删除所选单元格图标及其连结路径，属性表值。



全部选择 (在编辑下拉菜单内)



删除两个单元格之间的路径连线

下面的按钮用来切换检视，即显示或隐藏列表：



显示属性表



显示全局变量表

下面的按钮用于应用程序执行和停止



执行当前的电话流程



中止当前的电话流程

3.4 着手开始：探索 Flow Designer

与任何其它学习过程一样，最好的学习方法就是实践。让我们从设计普通任务开始吧。

3.4.1 语音播放单元格图标

在左边的工具条上，单击一个单元格按钮，移动鼠标到设计窗体内，你会注意到鼠标指针变成了一个小的单元格符号，如下图：



移动鼠标指针到设计窗体内所要的位置(即插入点)，再次单击鼠标，所要的单元格图标就会出现在插入点的左上角位置。

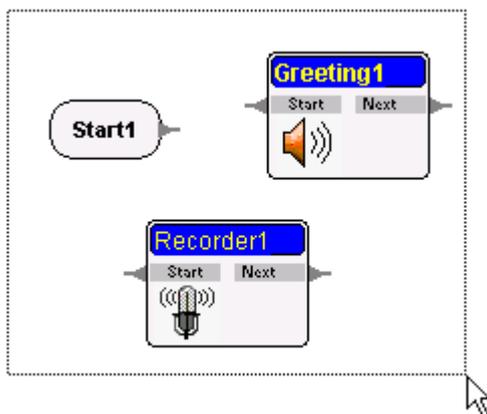


3.4.2 单元格图标的选取与取消选取

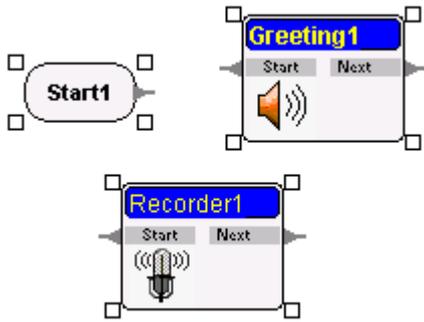
当单元格图标被放置于设计窗体内，或是单击选定它时，你可以看到它的四个角上有四个小方块(如上图)。取消单元格图标的选取，你需要单击设计窗体内除了四个小方框组成的矩形区域以外的任何一点即可。

你也可以通过单击属性表窗体下拉菜单里的标题名，从电话流程的多个单元格图标中寻找、选定一个单元格图标。要选定多个单元格图标，你可以用以下两种方法中的任意一种：

1. 单击单元格图标以外的一点，拖动指针形成一个矩形框，将你要选定的单元格图标包含在矩形框内，然后释放鼠标键。



2. 单击选中其中一个单元格图标，出现四个小方框。移动鼠标指针到另一个单元格图标，按住<Ctrl>键再单击，继续移动鼠标指针然后按住<Ctrl>键+单击下一个单元格图标，直到全部需要选定的单元格图标都被选定为止。



3.4.3 单元格图标的删除

1. 选定你想要删除的单元格图标。
2. 单击顶端工具条中的删除按钮。

单元格图标及其在属性表中的属性内容将被删除，所有子连结到这个单元格图标的路径线也将被删除。

3.4.4 在两个单元格之间放置连结路径

一个单元格图标代表一组待定义的指令。大多数单元格图标的左右两边都有指针。“起始”和“连结出口”只有右侧的指针。“结束”和“连结入口”只有左侧的指针。左侧的指针是单元格图标的入口，右侧的指针是单元格图标的出口。根据设定的条件，运行中的程序将会从右侧的指针中的其中一个退出。

连结一个单元格图标的右侧指针到另一个单元格的左侧指针表示连结两个程序段，右边的单元格接着左边的单元格之后执行。

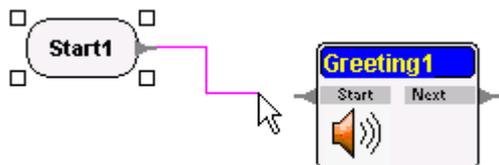
***** 2 个图标先执行第一个，再执行第二个

每一个单元格图标都有一个或多个指针用于连结到其它的单元格。如下所示，指针区域包括指针本身及暗灰色的指针文字区域。注意：部分单元格图标指针区域只有指针而没有指针文字区域 (e.g. 起始，结束，连结入口，连结出口，子流程)。

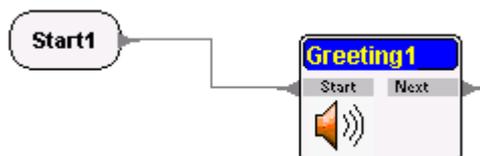


连结两个单元格图标的指针：单击某个单元格图标右侧指针区域，移动鼠标指针到另一个单元格图标左侧指针区域，一条粉红色曲

折线会跟随着鼠标指针的轨迹。当鼠标指针触到另一个单元格图标的指针区域时，再次单击鼠标。



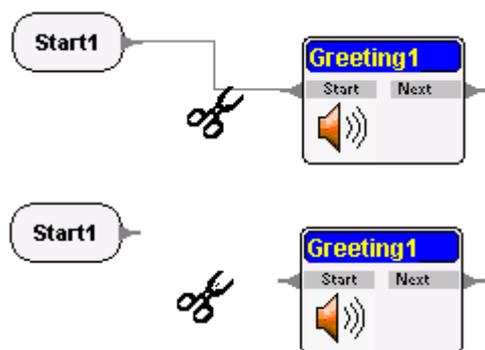
如下图所示，你会看到粉红色连结线被固定下来，并且变成了灰色。这样，两个单元格图标通过各自特定的指针连结了起来。



如果粉红色线出现了(可能由于你无意中的点击)，而你又不想将它连到任何别的单元格图标，在它变成固定灰色之前，单击除设计窗体内任何左侧指针区域之外的其它任何一点，粉红色线就会消失。

3.4.5 删除两个单元格之间的连接线

单击顶端工具条上的删除连结线按钮，你会看到鼠标指针在设计窗体中变成了一把小剪刀。移动剪刀形的鼠标指针到你想要删除的连结线上，在连结线上单击，连结线将被删除。如下所示：



3.4.6 移动选定的图标

1. 选定你要移动的单元格图标，将鼠标指针置于此单元格的矩形区域之内。
2. 拖动选定的单元格图标到你想要的位置，然后放下。请注意，避免拖动单元格图标之前将鼠标指针置于任何右侧指针区域上。因为这样会产生一条欲连接到另一个单元格图标的粉红色线。因此将鼠标指针置于单元格名称(标题)区域就可以轻松地移动选定的单元格图标。

移动单个单元格图标将会改变它与其余图标间的相对位置。所有连结到此单元格图标上的连结线也会随之移动。

如果你想要同时拖动或移动多个单元格图标，将鼠标指针置于所选定单元格中的任何一个图标名称(标题)上即可。同时移动多个单元格图标时，它们彼此之间的相对位置及连结线保持不变。

3.4.7 剪切/复制单元格图标及连结线

1. 选定要剪切/复制的单元格图标。
2. 单击顶端工具条上的剪切或复制按钮。

所有被剪切或复制的单元格图标及其在属性表中的属性内容(请参考“**属性表窗体**”部分)都将被粘贴，同时它们之间所有的连结线也将被同时粘贴。如果你将剪切/复制的单元格图标粘贴到另一个电话流程，它们就会如同是在原来的电话流程中一样出现在设计窗体中相同的相对位置。单元格图标的复制节省了你的时间，因为你不必手工去填写属性表及变量表。简单地编辑或者改变属性表中的标头和/或变量表中的变量，就可以生成一个相似的电话流程。

3.4.8 粘贴单元格图标及连结线

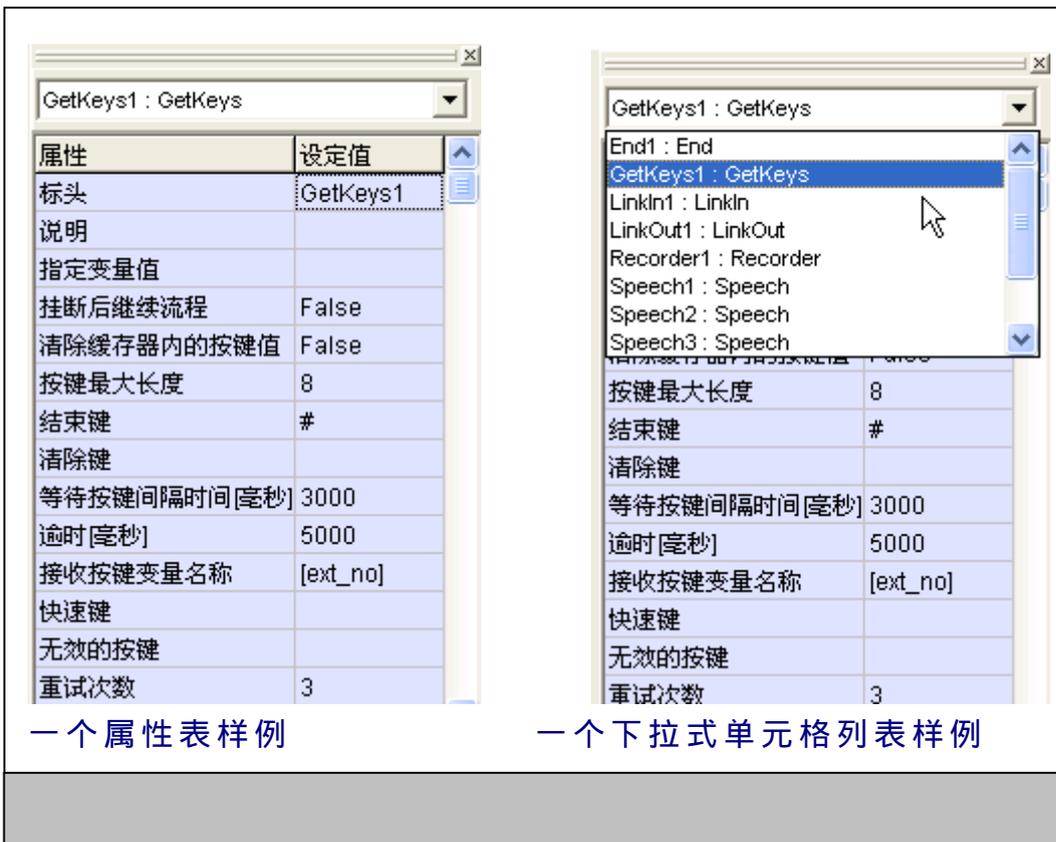
在剪切或复制了所选单元格图标之后，单击顶端工具条上的粘贴按钮，所剪切或复制的单元格图标将出现在原单元格图标右下 20 个像素的位置。你可以在粘贴之后将它们及其连结线拖动/移动到

想要的位置上。

所有被剪切或复制的单元格图标及其在属性表中的属性内容(请参考“**属性表窗体**”部分)都将被粘贴，同时它们之间所有的连接线也将被同时粘贴。如果你将剪切/复制的单元格图标粘贴到另一个电话流程，它们就会如同是在原来的电话流程中一样出现在设计窗体中相同的相对位置。单元格图标的复制节省了你的时间，因为你不必手工去填写属性表及变量表。简单地编辑或者改变属性表中的标头和/或变量表中的变量，就可以生成一个相似的电话流程。

3.5 属性表窗体

属性表窗体紧连着设计窗体，位于 Flow Designer 主窗体的右上位置。设计窗体内的每一个单元格都有相应的属性表，如果你单击选中一个单元格图标，相应的属性表就会出现在属性表窗体中。



每个单元格的属性表记录此单元格的属性设定值及参数。不同类型的单元格的属性/参数数量各不相同。不同类型的单元格具有不同的属性及参数条目。相同类型的不同单元格具有相同的属性条目和相同的属性数量，但属性值不同(至少表的名称(标头)不同)。属性条目包括**标头**(表的名称)、**挂断后继续流程**、**逾时**、**按键最大长度**、**快速键**、**无效的按键**等。

所有单元格都具有以下四个属性条目：**标头**、**说明**、**指定变量值**和**挂断后继续流程**。

标头是你可以分配给电话流程单元格控件的标题。默认名称是单元格控件的名称加上一个数字，此数字自 1 开始递增，表示同一电话流程中相同单元格的数目。比如：“Condition1”。

说明和标头相似，它们都为电话流程单元格控件提供了一个标签。此外，说明条目可以让你解释使用此单元格控件的意图，以使你的应用程序有更好的文档编制。这样，如果另一个人想使用你设计的应用程序，或者对它进行修改以适应新需求的出现，他可以了解你为何使用了该电话流程控件。

从编程方面看，说明属性类似于程序的注释语句-并不真正成为程序的一部分，但被包含在程序的语句里，使得编程者可以知道前后程序或步骤的意图。

指定变量值允许你指定一个变量值，它既可以被用于此电话流程单元格控件，也可以用于后继的单元格控件。你可以为它设定一个初始值，之后再调用它以决定电话流程将要执行的步骤。

挂断后继续流程决定在电话流程执行过程中，电话被挂断时此流程是否继续。程序会检查**挂断后继续流程**的当前值，如果当前值是 false，则电话流程相应地会中止。**挂断后继续流程**的默认值是 false。

如果电话处于挂断状态，并且当前正在执行的单元格**挂断后继续流程**的当前值为 true，那么程序不会中止，直到执行遇到下一个**挂断后继续流程**当前值为 false 的单元格。

挂断后继续流程的特性可被用于与数据库之间的操作，避免电话流程在电话挂断之后立即中止，这样电话流程就可以执行一次计算，或者其它的功能。比如在电话挂断后记录未成功的来电次数。

你可以设定和编辑属性表，通过单击属性表窗体上的设定值域，然后在域中输入相应的值即可。关于属性的细节以及它们如何影响一个电话流程，我们将在之后的章节中讨论。

如果域中一个条目太长不能完全被显示，你可以调整列的宽度。将鼠标指针移到域的顶端，鼠标指针会变成如下所示的图形，这时拖放分界线到想要的位置即可。

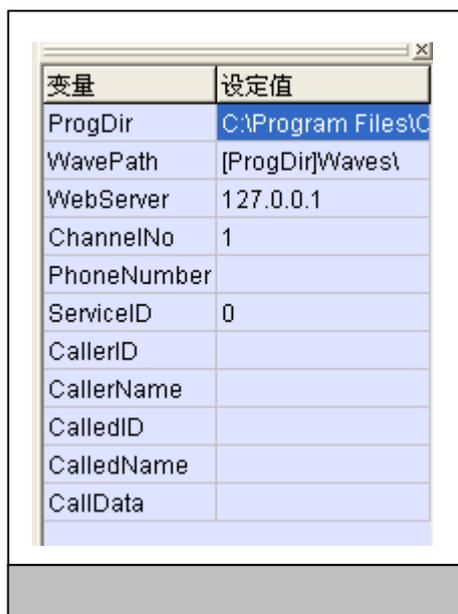


3.5.1 下拉式单元格列表

在属性表窗体的顶部有一个下拉式单元格列表(如上所示)。如果一个电话流程中有许多个单元格，你也可以通过单击下拉列表中的单元格属性表名称来切换到一个特定的单元格属性表。

3.6 变量表窗体

变量表窗体在属性表窗体的下方，位于 Flow Designer 主窗体的右下位置。它列出了电话流程的变量表。通常情况，每一个 IVR 文件都包含一个单独的电话流程，其中含有对应的变量表。一个电话流程的变量表中的值在此电话流程及其子流程中有效。你可以通过单击变量表窗体中的设定值域，然后在域中输入相应的值来定义或编辑变量表。



The image shows a screenshot of a software application window with a table titled '变量' (Variables) and '设定值' (Settings). The table contains the following data:

变量	设定值
ProgDir	C:\Program Files\C
WavePath	[ProgDir]Waves\
WebServer	127.0.0.1
ChannelNo	1
PhoneNumber	
ServiceID	0
CallerID	
CallerName	
CalledID	
CalledName	
CallData	

对变量表的操作，例如增加变量、分配变量及类似的话题我们将在以后的章节中讨论(即在我们学习了开发简单的 CONTAX IVR 应用程序的基础知识之后)。

第三部分 简单应用程序开发

到目前为止，你已经尝试了去使用 CONTAX IVR Flow Designer，并做好了应对特定需求开发流程控制应用程序的准备。在下面的章节中你将开始创建 CONTAX IVR 应用程序，从最简单的到相对比较复杂的。

第四章：你的第一个应用程序

当你使用 CONTAX IVR Flow Designer 开发应用程序的时候--事实上--是在进行计算机的程序编写。

开发计算机应用程序，其他的程序员使用“编程语言”，比如 C++、Python、Visual Basic 或 Java。大多数编程语言要编码或写成数条文本行，然后编译或解释成计算机可识别并执行的语言。

但其他编程语言如 Visual Basic，为开发者提供了一个图形用户界面（GUI），允许通过使用单击及拖放的方式开发应用程序，简化了开发过程。CONTAX IVR Flow Designer 就像第二种类型，为你的应用程序开发提供了基于图形用户界面的工具。

如果你已经具备了一些编程背景，你可以跳过下面的讨论，直接到第一个应用程序例子。通过阅读一些例子，弄明白我们是如何通过连结各种不同单元格控件组成应用程序，仍然会使我们受益非浅。

4.1 把它分割成细小的部分

对于没有任何编程背景的人士，编程可能会是一个让人惧怕的词。但不要担心，编程是你早已具备的一种天生能力。它只是简单地将你想要完成的事情分割成一连串步骤或行动的过程。例如，如果你想要喝速溶咖啡，你就要完成下面的步骤：

把水煮沸。
将热水倒入杯子内。
准备好速溶咖啡。
用茶匙将所需量的咖啡放入水中。
加入糖并搅拌。
如果需要，加入乳酪或牛奶。
享用你的热咖啡。

对于编程，或者为 CONTAX IVR 设计应用程序，你遵循类似的步骤：将任务分割成一连串的逻辑步骤。如果是使用计算机编程语言实际制作应用程序，就是将每一个步骤用适当的代码写下来。

对于 CONTAX IVR，创建应用程序只不过是选择每个步骤相应的控件图标，再把它们连结起来即可。当然，为了使应用程序完成你所要求的特定任务，你也许得去调整每一个单元格控件的默认设置。这个问题我们将会在以后的章节中进行讨论。

4.2 投入到设计中

与其它技能一样，比如游泳，只有一种方法才能使我们学会它：跳入水中开始游！----也就是去做。这就是我们目前将要进行的工作。不用担心你还没有熟悉所用单元格控件的特性，遵循指导，在需要时可以参考 CONTAX IVR 用户指南 (CONTAX IVR User's Guide)，你可以从中找到所有单元格控件的属性及如何调整以得到所要结果的更多细节。

4.2.1 应用一：问候

你的第一个编程任务，让我们从简单的开始。创建一个“问候”电话流程，自动接听电话并致问候语。

打开 Flow Designer，你会看到一个空白的设计窗体，表示你已经准备好开发一个新的应用程序了。单击流程操控单元格中的**起始**控件。

移动鼠标光标到设计窗体内，再次单击鼠标。**起始**图标将会出现在设计窗体内。

现在单击电话操控按钮，面板上会列出各种电话控件图标，单击**语音播放**。

再次移动鼠标光标到设计窗体**起始**图标的右边，单击一下。**语音播放**图标就会出现在设计窗体内，标识为“Greeting1”。

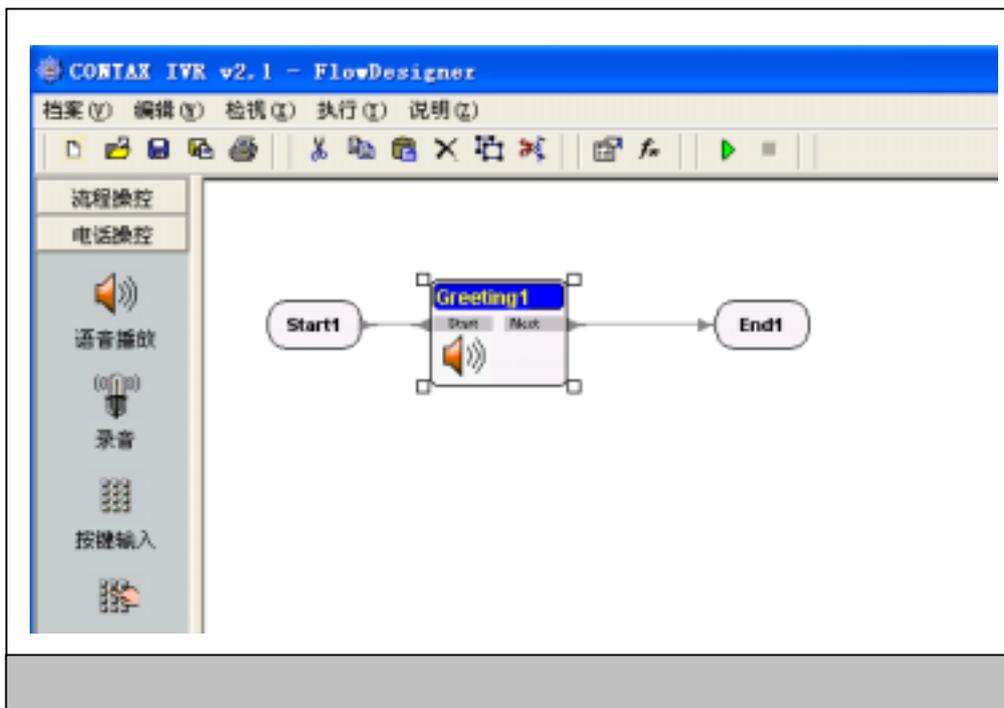
单击流程操控按钮，单击**结束**单元格。移动鼠标光标到设计窗体内，单击放置**结束**图标。

单击**起始**图标的指针，将鼠标向右移。你会发现无论怎样移动鼠标，它后面都会有一条紫色线跟随。将鼠标光标移动到“Greeting1”的入口指针处单击，一条灰色线会将两个图标连结起来。

技巧：你可能会发现点击指针的尖端来连结图标会很困难，请试着点击指针体。

连结“Greeting1”的另一个指针到结束图标。

现在应该可以看到类似下面的状态：

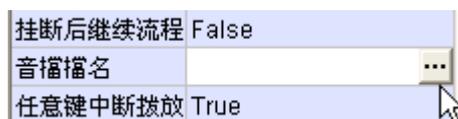


好了，现在你将要提供电话拨入时所要播放的问候语。但首先要注意语音播放图标所用的是 WAVE 文件(.wav)。你可以借助 CONTAX IVR 中的录音软件将自己的问候语保存为 .wav 文件。我们会在本章的稍后部分加以讨论。

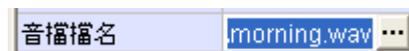
现在保持简单的原则，我们使用 CONTAX IVR 安装时生成的电话流程例子中的 .wav 文件。单击“ Greeting1 ”图标选定它，设计窗体右边上方-属性表窗体上会出现“ Greeting1 ”的属性。



这里最重要的属性是“音档档名”。点击其右边的空白域方格，再次点击就会出现一个白色的空白区域，右边有省略号(三点)按钮。



点击省略号会弹出一个“打开”对话框，默认为 CONTAX IVR 的安装目录。在对话框中打开“Samples”文件夹，接着打开“Greeting”文件夹，选择“morning.wav”并单击打开。你会注意到“音档档名”属性这时已经包含了“morning.wav”文件所在的目录。



你刚刚创建完你的第一个电话流程。

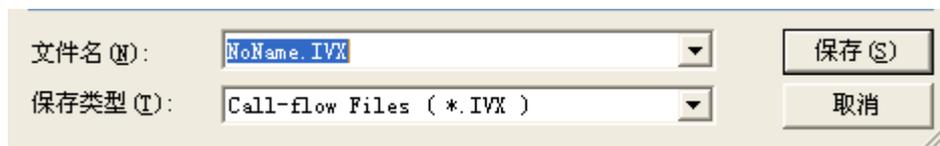
4.2.2 保存制作完毕的应用程序

但是在运行或测试你所制作的应用程序之前，你必须先保存它。记住，不管怎样其他程序员可能会在同一台计算机上开发 .IVX 应用程序。为了给你制作完成的应用程序提供详细资料，点击主窗体菜单条上的视图/档案信息，出现了一个信息对话框。输入你的 .IVX 文件的详细资料并点击确定。

选择“自动累加版本编号”复选框，你甚至可以保存电话流程制作过程中所做的修改记录。



要保存电话流程，点击 CONTAX IVR Flow Designer 顶端工具条上方的“档案”。在下拉列表中点击“另存新档”。“另存为”对话框被打开，默认的文件名是“NoName”。



但你可以将它改为你想要的任何名字。根据用途，这里我们输入“Goodmorning”并点击保存。

请注意你的应用程序保存的默认目录是 CONTAX IVR 的安装目录。你可以创建一个新的目录以便保存你制作完的应用程序，并在此目录中选择你所要运行或编辑的应用程序。

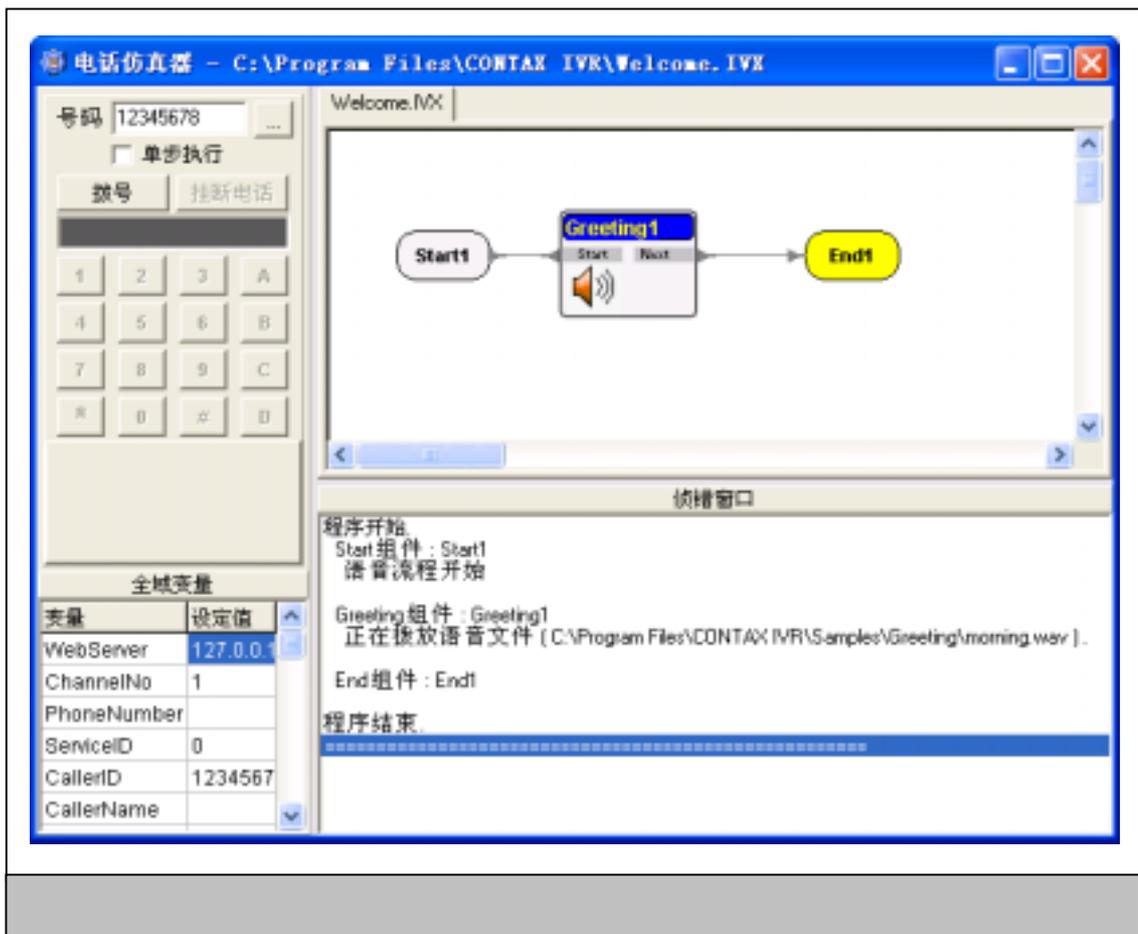
4.2.3 验证：模拟你的电话流程

一旦保存了你的应用程序，在将它交付实际使用之前，你必须对它们进行测试，以便发现可能的错误，或对它进行调试除错 (debug)。

“ Bugs ” 在计算机行话里是指程序的缺陷或错误。“ Debug ” 的意思是纠正程序的缺陷。

正如你在第一章中运行第一个例子程序时所见到的，CONTAX IVR **电话仿真器** 是一个多用途的调试工具，允许你跟踪电话流程的处理过程，使你可以看到当前活动流程控件图标是激活的。如果有任何文件正在被自动播放，你就可以听到相关的信息。

因此，让我们运行“ Goodmorning ”电话流程。打开 CONTAX IVR Flow Designer，点击主窗体菜单条上的**档案>开启旧档**，会出现一个文件打开对话框。从所在的文件中单击 **Goodmorning.ivx** 文件，单击打开对话框上的**打开**按钮。“ Goodmorning ”电话流程就会出现在主窗体中的设计窗体中。单击顶端工具条上的执行按钮，出现**电话仿真器**窗体。



单击电话仿真器上的拨号按钮开始运行电话流程。你可以看到 Start 图标变成了黄色，接着是“ Greeting1 ”图标。你会听到一个汉语问候语(因为预先录的信息是汉语)，不管怎样，他说的是“早上好”的意思。

如果你想要一步一步的执行应用程序，请选中**单步执行**复选框。当这个框被选择时，电话仿真器会在每一步或每一个断点处停下来。

要再次执行流程，你必须先中止当前的电话流程。只需单击突出的结束按钮 ，然后再次单击执行按钮即可。

电话仿真器有一个侦错窗口，你可以从中看到程序处理的代码信息。侦错窗口将会告诉你应用程序中是否含有错误，比如语法错误等。侦错窗口对应用程序开发非常有用，因为在将你所设计的应用程序交付实际使用之前，你可以对其进行测试，并找出其错误之处。

4.2.4 声音信息的录制

那么，你如何录制你自己的 .wav 文件以运用于你的应用程序呢？

CONTAX IVR 已经在它的软件包里捆绑了自己的语音录音机，但是你也可以使用其它录音程序，比如 Microsoft 公司的录音机，通常它会包含在 Microsoft Windows(Win9x ,WinME ,Win2000 ,WinXP) 的标准安装里。如果你不知道如果用录音机程序录音，请参考相关的帮助文档。CONTAX IVR 自带语音录音机的操作和 Microsoft 公司的录音机类似。

按照下面的步骤，你可以用自己录制的 .wav 文件代替问候语：

1. 打开“ Goodmorning.IVX ”文件，选择 Greeting1 图标，双击“ 音档档名 ”属性设定值域。
2. 进到你保存录音文件的目录，选择正确的 .wav 文件，单击打开。此文件路径会出现在“ 音档档名 ”的属性值里。

3. 点击顶端工具条上的档案/存储档案，保存你的修改。
4. 再次执行这个应用程序，你听到的信息将是你所录制的问候语。

4.2.5 使用文本朗读(Text-To-Speech)

你也可以通过使用文本朗读(TTS)功能来创建一个类似的“Goodmorning”应用程序，使用TTS可以代替回放一个.wav文件作为问候。

CONTAX IVR 支持 Microsoft Speech API (SAPI)引擎，Microsoft XP 已经内含了 Text-to-Speech(TTS)功能。对于其它的 Windows 操作系统，你可以从 CONTAX IVR 光盘上安装 TTS 功能(执行 **SAPI51** 目录下的 **setup.exe** 安装 Microsoft Speech SDK5.1，执行 **SAPI51LangPack** 目录下的 **setup.exe** 安装 Microsoft Speech SDK5.1 语言包)。

使用“Speech”图标，并写下你想要播放的问候语，计算机将会以话音的方式“读出”—回放这些文本。

为了创建一个类似的应用程序，用TTS代替.wav文件作为问候语，请按如下的步骤：

回到创建你的“Goodmorning”应用程序的过程中，完成步骤1和2。

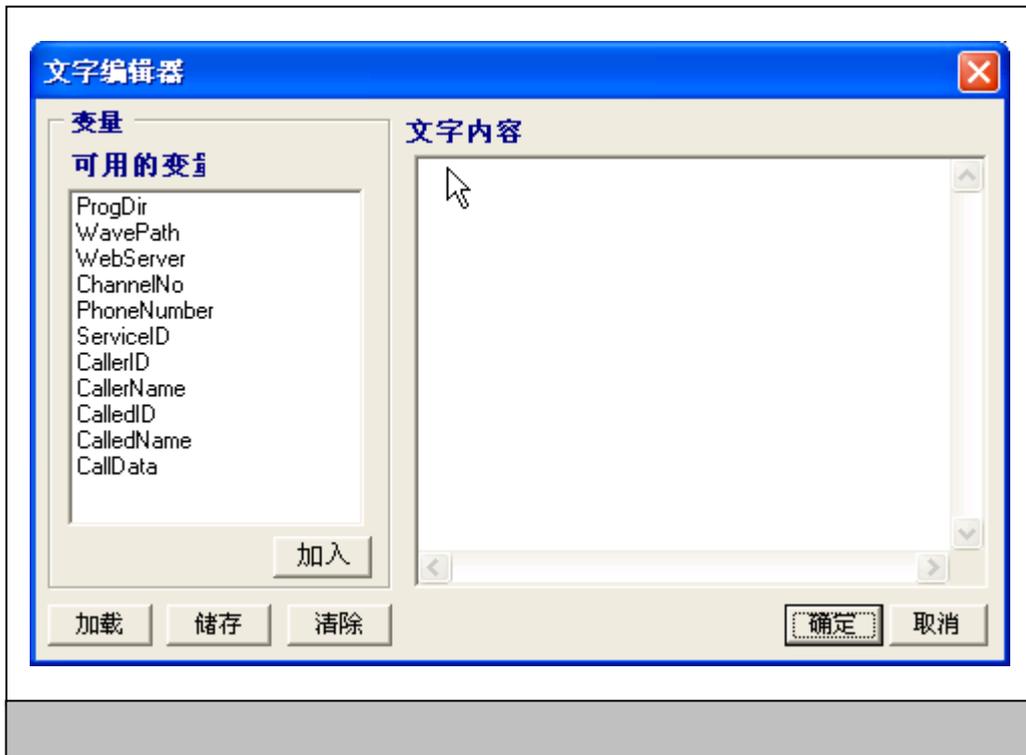
与第3步使用电话控件中的单元格不同的是，这里我们点击媒体控件并单击“文件播放”单元格。

在设计窗体上单击，出现“Speech1”单元格图标。

完成步骤5到7。

选择 Speech1 单元格，在属性表上双击**文字转语音内容**属性的设定值域。

单击出现的省略号按钮，会打开一个文字编辑器对话框。



6. 单击文字内容框并输入“Good Morning”，点击确定。

7. 将应用程序保存为“Goodmornig2”。

执行“Goodmornig2”。

你听到计算机说“Good Morning”了吗？

第五章：选择和结果

如果你所有的应用程序都是向来话者致问候语，那我们用计算机系统来做这个工作就没有什么意义了。

请记住在第一章中我们说过 CONTAX IVR 可以设计为替代过去人们常做的重复性工作。在以下的讨论中你将参与到完成一个办公室中常规通信需求的 CONTAX IVR 应用程序的创建过程中来。

在实际情况中，事情并没有如此简单与直接。因此为办公室日常通信需求所开发的 CONTAX IVR 应用程序要比“问候语”电话流程复杂得多。

在使用用户交换机的办公室里，话务员通常会问来话者想要联系的人或者所要部门的内部分机号码，然后将电话转接到相应的地方。

5.1 条件和循环

CONTAX IVR 具有自动根据选项或**条件**进行处理的功能。它可以被设计用于将流程跳回到指定的点 或者完成一次**循环**的功能。比如一个来话者按了错误的按键，就必须提示他直到键入正确的键。通过使用**条件**和**循环**，你可以设计应用程序允许来话者从多个选项之中选择一个动作。

5.1.1 应用二：来话自动转接

为了找出 CONTAX IVR 是如何借助条件和循环来处理比较复杂的情况，我们设计一个应用程序自动将来话转接至指定的本地分机号码。

基于上面的目的，我们设计一个电话流程来自动处理每一个打入 XYZ 计算机公司的电话，将这些来话转接至销售、财务、行政、技术四个部门中的任何一个部门 或者话务员。

你可以先将任务分割成一系列的逻辑步骤，从而设计出你的来话自动转接流程：

1. 接到来话呼入。
 - 向来话者致问候语。
 - 告诉来话者可以联络的相应部门的号码，或者到话务员的号码。
 - 找到来话者所拨的号码。
 - 如果来话者拨了 1，将电话转接到销售部门。
 - 如果来话者拨了 2，将电话转接到财务部门。
 - 如果来话者拨了 3，将电话转接到行政部门。
 - 如果来话者拨了 4，将电话转接到技术部门。
 - 如果来话者拨了 5，将电话转给话务员。

2. 判断如果来话者拨了其它号码，播放“对不起”并且提示输入正确的号码，然后跳回到第 2 步。
 - 如果来话者累计两次按了不正确的键，跳到电话流程的结束单元并挂机。
 - 如果所要转到的分机响铃并有被应答，跳到电话流程的结束单元。
 - 如果所要转到的分机忙线，播放“对不起”并且跳到电话流程的结束单元。

如果所要转到的分机响铃但无人接听，跳到电话流程的结束单元。

结束电话流程。

程序员通常都会以流程图的形式准备这样一个列表，它是由不同形状的方块组成的图表--每个方块代表一个特殊的动作--由线条连结标出执行的顺序。

即使你没有程序编写的背景，我们仍然推荐你使用自定义的流程图。它们会成为可视化的指导从而帮助你在 Flow Designer 中创建自己的应用程序。你可以只画一些正方形或矩形的方块来代表每一个步骤并且对其进行相应地标注。

现在你可以运行 CONTAX IVR Flow Designer，创建真实的应用程序。选择合适的控件图标并把它们连结起来。

单击起始单元格并将图标放置于设计窗体内。

点击媒体控件按钮，单击文字播放图标并将其放置于起始图标的旁边

连结两个图标。

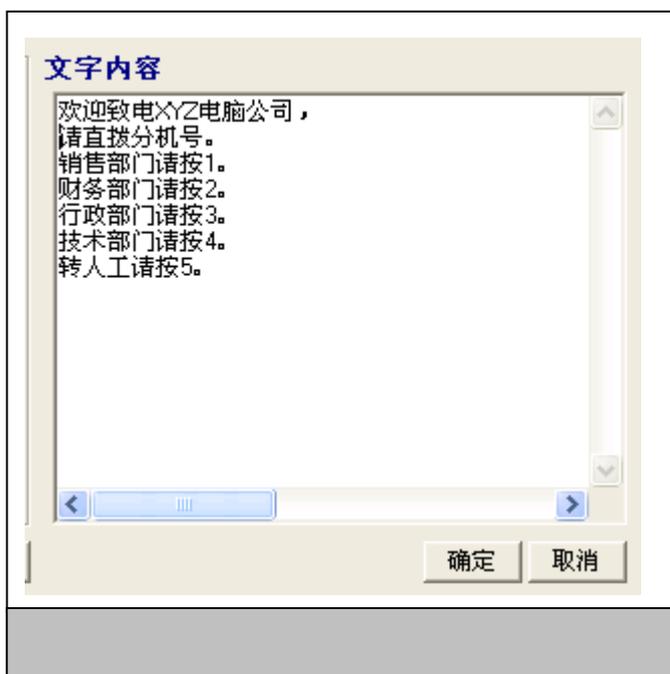
选定文字播放 (Speech) 图标。

在属性表窗体面板上点击文字转语音内容的空白设定值域，再单击省略号打开一个文字编辑器对话框。

点击文字内容框的空白部分并输入以下内容：

“ 欢迎致电 XYZ 电脑公司，请直拨分机号。销售部门请按 1。财务部门请按 2。行政部门请按 3。技术部门请按 4。转人工请按 5。”

你的文字框会是如下类似的样子。



点击 **确定**。

在电话控件面板中选择 **按钮输入** 单元格并将其放置于 **Speech1** 图标的旁边，将它们连结起来。

注意：GetDTMF(按钮输入)等待电话机端的一个按键，它的流程出口就是所输入按键的对应指针。如果你没有将其中一部分指针连结到其它单元格，输入这部分指针相应的按键会被视为无效按键而被中断，这时流程会把 Invalid 指针作为它的出口。

如果要进一步了解 **GetDTMF** 单元格的属性，请参考 CONTAX IVR 用户手册中的相应章节。

在电话控件面板中点击 **转接** 图标，并把图标放置于 **GetDTMF** 的右边。再同样生成 4 个 **转接** 图标。

注意：**转接** 单元格的任務就像它的名字一样 - 将电话转接到特定的分机或者外部电话。通过单击目的号码右边的设定值域，并输入相应的号码就可以指定所要的内部或外部电话号码。

连结 **GetDTMF** 号码 1 出口指针到第一个 **转接** 图标；连结 **GetDTMF** 号码 2 出口指针到第二个 **转接** 图标。以此类推，直到 1-5 的出口指针都连结到对应的转接图标。

将 **结束** 图标放置在第一个 **转接** 图标(Transfer1)的右边。

将第一个 **转接** 图标(Transfer1)的 Next 指针连结到 **结束** 图标(End1)。

在第一个 **转接** 图标(Transfer1)的右边，放置一个 **连结入口** (LinkIn)图标并且将其连结到 Busy 指针。

注意：一个 **连结入口** 的流程是从同一个电话流程中的相同名字的 **连结出口** 退出的。你可以改变连结入口图标的默认名字或者其中的字母，只要点击属性表中 **连结名称** 右边的设定值域并输入所要的字母即可。这里我们使用默认字母“A”。你可以选择不使用 **连**

结入口或者**连结出口**控件，但在连结应用程序中的各种图标时使用它们会特别有帮助，可以使你不至于晕头转向。

在第一个连结入口图标的下面再生成一个**连结入口**图标，并连结到**转接**图标的 *NoAnswer* 指针。请确认将连结入口的**连结名称**改为“B”。

重复步骤 12 到 14 直到 5 个**转接**图标中的每一个的 3 个指针都各自作了相同的连结。

在**按键输入**图标 (GetDTMF1) 的下方放置一个新的**文字播放**图标 (默认名称为 **Speech2**)，将前者的 *Invalid* 出口指针连结到后者的 *Start* 指针。

打开文字转语音内容属性设计值的**文字编辑器**窗体，在文字内容对话框中输入：“对不起，您的输入有误，请重新输入。”

在 **Speech2** 的右边放置一个**计数**图标 (Counter)，将 **Speech2** 的 *Next* 指针连结到 **Counter1** 图标的 *Start* 指针。

在 **Counter1** 的右边放置一个**连结入口**图标，并标为“C”。将其连结到 **Counter1** 的 *Next* 出口指针。

在 **Counter1** 的右边再放置一个**连结入口**图标，并标为“D”。将其连结到 **Counter1** 的 *Over* 出口指针

在 **Speech1** 和 **GetDTMF1** 之间的上方，放置一个**连结出口**图标，并标为“C”。将其连结到 **GetDTMF1** 的 *Start* 指针。

在 **Speech2** 的下方放置两个新的**文字播放**图标。

打开 **Speech3** 的文字编辑器，输入：“对不起，电话正忙。”

打开 **Speech4** 的文字编辑器，输入：“对不起，您拨打的分机无人接听。”

在 **Speech3** 图标的前面放置一个**连结出口**图标，并标为“**A**”。将其连结到 **Speech3** 图标的 *Start* 指针。

Speech4 作同样的操作，但连结出口的标识为“**B**”。

在 **Speech3** 和 **Speech4** 旁边各放置一个**连结入口**图标，并标为“**D**”，再把它们和 **Speech** 的 *Next* 指针相连结。

在**结束**图标(End1)的上方，放置一个**连结出口**图标，并标为“**D**”。将两个图标连结起来。

将你设计好的应用程序保存为 **Autotransfer.ivr**。

运行**电话仿真器**对程序进行测试，尝试不同的选项直到确信此电话流程已经按照你的设计要求正常工作。

好，现在你已经对如何设计及生成实际 CONTAX IVR 应用程序有了一个很好的概念。

5.1.2 应用三：用不同方式做同一件事

完成一件事情有很多种方法，程序开发也是如此。在上一个例子中，你使用了**按键输入**(GetDTMF)单元格控件来将电话转接到本地分机。这是处理来话者不同选择的一种最简单的方法，尤其是和**转接**单元格控件配合，用于将电话自动转接到所要的号码。

但是你也可以使用其它的控件来处理来话者的选择。**资料输入**(GetKeys)控件和**条件**(Condition)控件配合使用既能够处理简单的来话者选择，也能够处理比较复杂的来话者选择。因为你将要使用**资料输入**和**条件**控件模仿交通警察来决定电话流程的走向，所以有必要对**资料输入**和**按键输入**的特性进行一下比较。当设计应用程序时要处理来话者输入时，它将有助于你决定要使用哪一个控件。

5.1.3 资料输入 (GetKeys) 对比按键输入 (GetDTMF)

那么**资料输入**控件和**按键输入**控件之间有什么不同的地方呢？两个控件都是处理按键式电话端的输入，决定下一步要执行什么动作。

输入的数量

按键输入控件仅仅只能处理单个输入。即使来话者按了多个键，**按键输入**控件只会记住第一个按键，后面的按键将被忽略。

而**资料输入**控件能够处理一定数量的按键式输入。你可以通过改变属性表中的按键最大长度设定值来指定输入的按键数。它的默认值为 8。

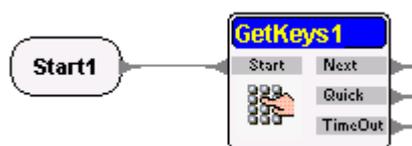
出错处理

两个控件都能判断无效的按键。正如你在第二个应用程序中看到的那样，**按键输入**控件在当对应的出口指针未连结到其它任何控件时，将其理解为无效的按键输入。当发生无效按键输入时，流程从 *Invalid* 指针退出。

资料输入控件有专门处理无效按键的内建属性项。因此，你必须指定无效的按键，它们才会生效。点击**无效的按键**属性的设定值域，输入所要指定的无效的数字即可。例如，语音菜单使用按键 1 到按键 5，你就可以把 6, 7, 8 定义为无效的按键(两个按键间用逗号分隔)。

当**资料输入**检查到无效的按键输入时，流程会循环回 *Start* 指针等待下一次输入。如果你在属性表中的**按键错误提示语音**设定值域指定了一个 .wav 文件，资料输入会播放这个文件。**按键输入**没有内建检测到无效按键后播放语音的功能。

为了更好的理解**资料输入**控件的出错处理功能，运行 Flow Designer 生成如下的电话流程：



定义 6, 7, 8 为无效的按键。

属性	设定值
无效的按键	6,7,8

录下这些内容：“对不起，您的输入有误，请重输。”将其保存为 Wrong.wav，放在 ContaxIVR\Waves 文件夹内，并在按键错误提示语音设定值域中指定此文件。

将此电话流程保存为 Experiment1。执行这个流程并输入指定的任意一个无效按键，你就会听到回放 Wrong.wav 文件的语音。

现在你已经可以设计不同的来话自动转接流程了，为了使它更有意思，你不仿加入一个语音录制特性。下面是具体的要求：

为 XYZ 电脑公司设计一个来话自动转接流程。

在第二个应用程序中指定的 4 个部门或者话务员可供来话者作选择。

我们使用下面的按键号码取代单个数字键，实现来话转接：

- 101 销售部门
- 102 财务部门
- 103 行政部门
- 104 技术部门
- 105 话务员

使用语音提示信息 (.wav 文件)。
给出按键错误提示信息并提示重新输入。

如果分机忙线或无人接听，为了能够回复来电请提示来话者录下他的名字和电话号码

为了先录下提示语音的 .wav 文件，我们打开录音机程序录下下面的内容：

第一个问候：“欢迎致电 XYZ 电脑公司，请直拨分机号。销售部门请按 1。财务部门请按 2。行政部门请按 3。技术部门请按 4。转话务员请按 5”。将此文件保存为 Transfer.wav，放在 ContaxIVR\Waves 文件夹内。

出错提示：“对不起，您输入的号码有误，请重新输入”。将其保存为 Wrong.wav 放在 Waves 文件夹内。

忙线提示：“对不起，线路正忙”。将其保存为 Busy.wav 放在 Waves 文件夹内。

无人接听提示：“对不起，此分机无人接听”。将其保存为 NoAnswer.wav 放在 Waves 文件夹内。

提示信息：“请留下您的姓名和电话号码，我们稍后会给您回复。录音结束请按任意键。谢谢”。将其保存为 RecName.wav 文件。

让我们开始创建这个应用程序的一系列过程。但和前面例子中的细节指导不同的是我们将使用实际电话流程的一个部分作为例子。当对电话流程的执行存在至关重要的地方时，我们就会对相关电话流程的重要特性进行讨论。

现在开始创建你的应用程序：

1. 将下面的控件连结起来：



使用 Transfer.wav 文件作为 Greeting1 的音档档名属性的设计值。

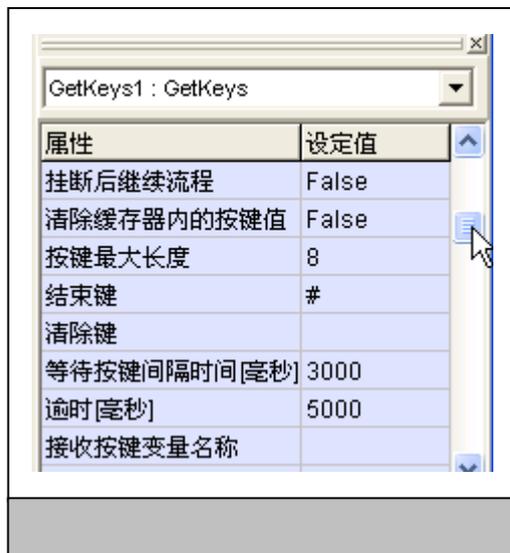
下一步，你要指定一个变量用于存放所输入按键的值。变量可以被看作是计算机内存中用于存放数据的盒子。你必须为每个变量指定一个标识或名字，这样当你需要一个变量中的数据时计算机才会知道从哪一个“盒子”中去找。在后面的章节中你将接触到更多的关于变量的内容。目前你只须将它视为存放来话者所输按键值的容器。

要指定变量，先在设计窗体中选定 GetKeys1 图标。你可以看到窗体右边出现了 GetKeys1 的属性表和变量表。列出的 9 个默认变量在所有控件中都是相同的，但它们都不能完成你现在的目的。要增加一个新的变量，你可以点击列表中最后一个变量（这里是“Call Data”）的设定值域，再按键盘上的向下键。

创建了一对新的空白单元，点击变量域并输入你的变量名。这里，我们将变量命名为“Key1”。设定值域保持空白。

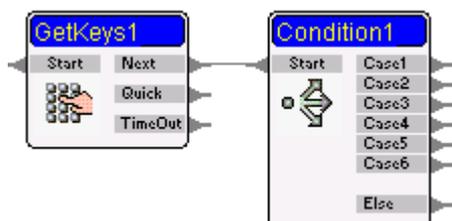
CallData	
Key1	

接下来，在属性表中点击**接收按键变量名称**的设定值域。你可能要将属性表边框往下拖才能看到**接收按键变量名称**属性项。



输入你之前创建变量的名称，用方括号括起来“ [Key1] ”并按回车键。方括号使计算机直接访问到变量 Key1 的值。

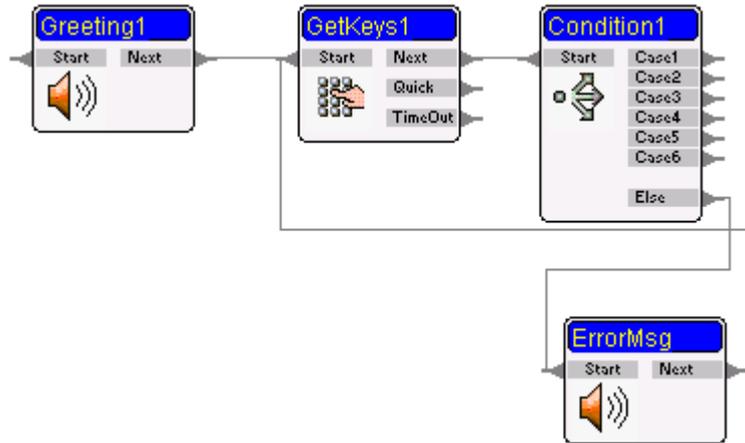
2. 下一步，将 GetKeys 控件与一个条件控件 (Condition) 连结起来。



在属性表上可以看到属性 Case1-Case6，如果需要请把边框往下拖。点击 Case1 属性的设定值域，并输入：“ [Key1]=101 ”。这是告诉计算机当来话者按了 101，Condition1 单元格的流程会把 Case1 指针作为出口。

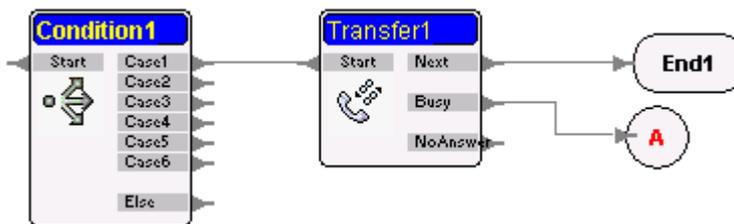
记得我们的要求，XYZ 公司有 5 个本地号码：101, 102, 103, 104 和 105。所以对于 Case2，将 Key1 的值等于 102, 103 对应于 Case3，以此类推直到 Case5。那么来话者输入了错误的号码怎么办？如果是这样，Condition1 会从 Else 指针退出。因此这里你可以连结另

一个语音播放单元格，给出错误信息并提示重新输入。将此 Greeting 单元格标注为 “ErrorMsg”，并指定音档档名为 Waves 文件夹中的 Wrong.wav。



接下来，你必须使来话者能够输入正确的本地号码。如果他输入了 101，计算机会将这个值给变量 Key1，并和你设定的接收按键变量名称属性值比较。因为你已经指定了如果 Key1 值即 [Key1] 等于 101，Case1 就为真，流程会将 Case1 指针作为出口。

类似的，如果来话者按了 102，计算机会把 Case2 指针作为出口；如果为 103 就是 Case3；以此类推。通过使用转接单元格，你可以实现电话的转接。将转接单元格连结到 Case1 指针，依此操作直到 Case5 指针。



注意：转接单元格最重要的属性是目的号码。

目的号码	101
------	-----

你可以将它的值设定为你要转接的实际电话号码，或者存有所要电话号码的变量。如果你没有指定任何电话号码或者变量，来话就不会被转接到所要的分机号码或者外部线路。

转接单元格检测所要的号码被接听、忙线、或者无人接听。如果电话被接听，本应用程序的要求-来话自动转接-已经达到，因此你可以通过连结转接单元格的 Next 指针到一个结束单元格(End)来终止本电话流程。

如果这个号码忙线中，流程会从转接单元格的 Busy 指针退出。根据程序的要求，你必须生成一个信息告知来话者此线路正忙。你可以通过连结一个语音播放单元格到 Busy 指针来实现。标注这个语音播放单元格为“Busy”，并指定音档文档为 Busy.wav。

以同样的方法，创建另一个语音播放单元格，并标为“NoAnswer”，并将它连结到转接单元格的同名指针。指定音档为 NoAnswer.wav。

连结入口，连结出口

并非一定要这样，但使用连结入口与连结出口将是一个很好的选择，它可以避免你在一堆交错的连结线面前不知所措。下面就是一个实际运用的例子：

剪去 Transfer 单元格与 Busy、NoAnswer 信息单元格的连结线。选定这两个信息单元格，并把它们拖到设计窗体的一个空白区域中。

创建一个新的连结入口单元格并标为“A”。将 5 个 Transfer 单元格的 Busy 指针连结到“A”。创建另一个连结入口单元格标注为“B”；将所有 NoAnswer 的指针连结到“B”。

创建一个连结出口单元格并标为“ A ”，将它连结到 Busy 单元格的 Start 指针。

再创建一个连结出口单元格标注为“ B ”；将所有 Transfer 单元格的 NoAnswer 指针连结到“ B ”。

当 Transfer 单元格检测到一个忙线信号，流程就会从它的 Busy 指针退出。当流程发现所连结的连结入口单元格，就会寻找和它同名的连结出口单元格并继续执行。当流程发现连结出口被连结到 Busy 或 NoAnswer 语音信息时，计算机就会播放各自指定的 .wav 文件。这样就可以照顾到电话忙线或无人接听时的提示要求了。

信息的录音

那么，你一定记得我们同样希望 CONTAX IVR 能录下来话者的名字和电话号码，以用于可能的电话回复。在进行录音之前，你希望能给来话者一个指导提示。

你可以用下面的方法来处理这个问题：

另外再连结一个信息单元格到 Busy 和 NoAnswer 单元格的 Next 指针，标注这个新的信息单元格为 RecOpt，并设 RecName.wav 文件为它的语音回放文档。

创建一个**录音**单元格，并将它的 Start 指针连结到 RecOpt 单元格的 Next 指针。

注意：录音单元格，正如它的名字，记录一个 .wav 文件并将它保存于一个指定的文件夹内。默认为 **C:\Program Files\Eletech\Contax IVR\Waves**。当然，你也可以指定你自己的文件目录，将所有录音下来的文件保存在那里。请参考用户指南中的相关内容。

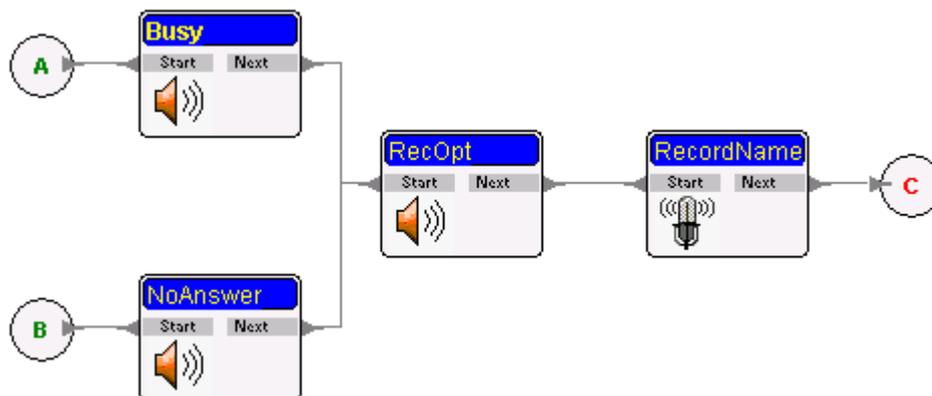
如果你将**任意键结束录音**属性值设定为 True，你就可以在任何时候按任意一个按键结束录音。如果它的值为 False，直到结尾才会

停止录音：录音长度[秒]属性的设定值。这里我们保留任意键结束录音的值为 True。

信息录音之后，程序就应该结束了。完成这个步骤，请创建一个新的**连结入口**单元格并标为“C”是。将录音单元格的 Next 指针连接到“C”。

在**结束**单元格旁边创建一个**连结出口**单元格，标为“C”，将这两个单元格连结起来。

你的控件单元格可能将会是下面的样子：



你刚刚完成了一个新的应用程序，一个新版本的来话自动转接程序，但它更为复杂，并增加了录制来话者信息的功能。保存你的应用程序，并运行电话仿真器对程序进行侦错，直到此电话流程满足你的设计要求为止。

请记住你所使用过的控件图标的特性，因为在下一章中，当我们开始制作高级电话流程时，你将会再次用到它们。

第四部分 高级电话流程设计

到目前为止你已经掌握了一些开发 CONTAX IVR 一般应用程序的基本技能，现在是接触一些更为有趣的事情的时候了。在接下来的讨论中你将会学到以下内容：

- 使用子流程
- 定义及管理变量

第六章：子流程：简化复杂的事情

要创建一个自动处理大型公司绝大部分通讯需求的电话流程，即使不提及各个控件单元格间的混乱的连线，因为它的复杂性，也可能将会是一件让人畏惧的事情。但是，有一个方法可以解决这个问题，并且维持原来的简单性：使用一个子流程。

6.1 什么是子流程？

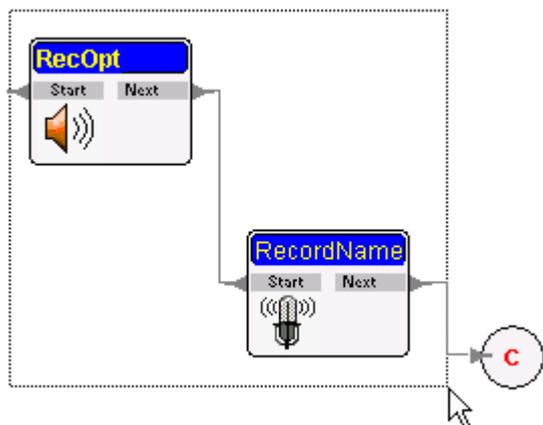
一个子流程类似于计算机编程语言中的子程序，与主程序分开但可以被主程序调用执行的一段程序。当子程序的任务执行完毕后，它就把球传回，也就是说，交给主程序。

一个子流程事实上代表一个独立的 CONTAX IVR 电话流程。当子流程执行完毕后，主流程继续执行自己余下的流程。

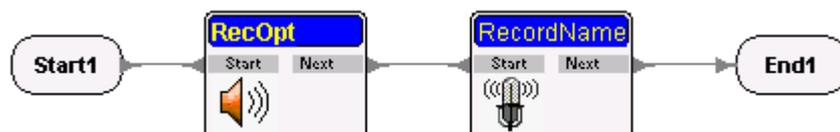
这是到目前为止子流程最重要的特性。它意味着你可以通过创建一些简单的流程，再将它们串起来如同一个单一的应用程序，以完成一个复杂的应用程序。

6.1.1 应用四：具有语音邮件功能的来话自动转接程序

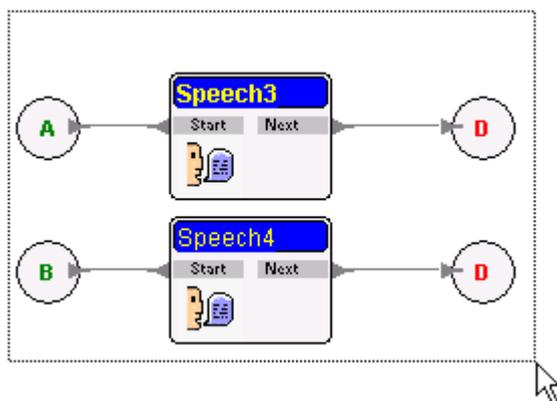
为了说明子流程是如何简化应用程序的创建，我们将要把自动来话转接应用程序与录音电话流程结合起来。事实上，你可以使用第二个应用程序中的录音部分，采用一个子流程来扩展来话自动转接电话流程的功能。



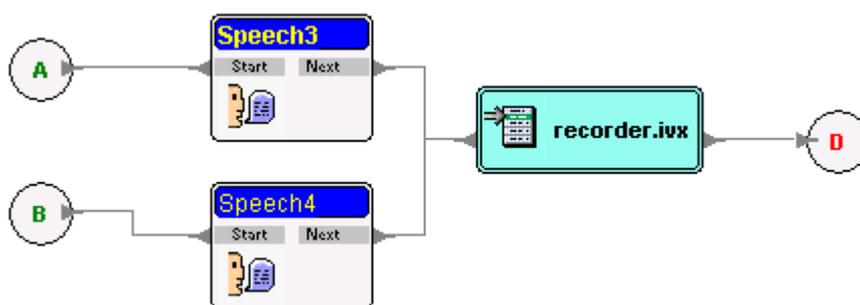
你可以将上面的录音电话流程部分定义成一个子流程，通过简单地复制粘贴生成一个新的应用程序(如下图)：



以新的文件名将这个新的应用程序保存起来。在这们的例子中你可以将上面的电话流程保存为 Recorder.ivx。然后我们转到 Autotransfer 电话流程连结入口前面的部分：



你应该可以看到下面的情况：



将这个应用程序保存为 `Sub_flow.ivx`。你是否看到通过使用子流程，生成一个新的应用程序是如此的简单？

6.1.2 保存与命名子流程

在定义子流程时，最重要的是指定恰当的文件名(.ivx)及电话流程的保存目录。

如果**使用 ProgDir** 的属性为 True，就只要填写文件名即可，默认路径为：

ProgDir = C:\Program Files\EIetech\Contax IVR\

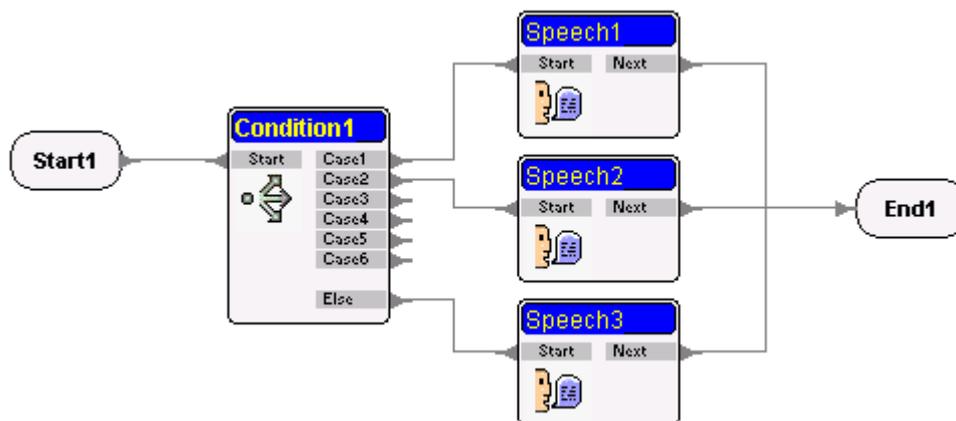
这是指我们保存 CONTAX IVR 新电话流程的路径，也是应用程序查找**子流程**控件的地方。如果你将电话流程保存在其它目录，你应该把**使用 ProgDir** 的属性值设定为 False，并指定 **ProgDir** 的值为你实际保存电话流程的文件夹。比如：

ProgDir = D:\Contax IVR\Subflows

第五个应用程序：时间检测

这里有一个小小的应用程序，你可以将它作为子程序紧随在起始单元格后面，其它信息单元格的前面。它提供一个常用的信息：欢迎致电(公司名称)。这个应用程序简单地测定一天的时间，并给出正确的问候语：“早上好”，“中午好”，“晚上好”。

通过使用这个短小的应用程序，你就可以保证当任何时间来话者呼入时都能给出正确的问候语。打开这个应用程序 `Timecheck.ivx`，你会发现它只仅仅包含如下的的控件单元格：



这个应用程序使用了变量、NOW 函数和一个表达式来得到正确的时间。从表达式返回的值将和下面的值进行比较：1200 和 1800。

如果这个值小于 1200，相应的问候语是“早上好”；如果大于 1200 但小于 1800(等于军用时间的下午 6:00)， “下午好”；如果大于 1800，“晚上好”。

在下一章中你将会学习到更多的关于变量、函数和表达式的内容。当你阅读完下一章后，再次打开 `Timecheck.ivx` 应用程序时将会更容易明白这个程序是如何工作的。

6.1.3 生成一个子流程库

当你以子流程集的方式设计你的 CONTAX IVR 应用程序时，你就不必手工涂画地去生成新的应用程序，因为已经存在针对大多数公司或特定公司的不同部门间通信需求的应用程序。你只需单单使用以前设计的流程---可能要做一些细小地修改---来开发满足要求的应用程序。你将会相应的节省开发应用程序的时间。

比如，你的子流程库可以包含如下内容：

- 来话自动转接子流程
- 发送传真子流程
- 接收传真子流程
- 发送 e-mai 子流程
- 问候子流程
- 信息录制子流程
- 发送短信息(SMS)子流程

随着应用程序的开发，你的子流程集也在不断地增大。它将不仅可以缩短你应用程序的开发过程，并且可以使你更加的自信。因为你已经测试了每一个子流程并且修正了它们的错误---使它们不再包含错误---你的应用程序中出现缺陷(bug)的可能就会更少。

第七章：变量和表达式

变量，如前面章节所学的那样，就像是一个计算机内存中的容器，或是一个虚拟的盒子，可以存放数据。变量存放两种类型的数据：数值型(数字)或子符型(字母、字母数字组合)。

变量存储数据，你可以通过这些数据来比较或判断主程序要进到哪一个分支程序，这样就简化了应用程序的设计。

7.1 变量表窗体

你可以在变量表窗体中察看每一个电话流程控件的变量，变量表位于属性表窗体的下方，主窗体的右下方部分。通常每一个 IVR 文件包含单独的一个电话流程，它带有相应的变量表。一个电话流程变量表中的值为整个电话流程及其子流程所共享。

通过点击变量表窗体上的变量设定值域，然后在这些域中输入相应的值，你就可以定义及编辑变量表。

7.1.1 在变量表中增加变量

下面是每一个电话流程的所包含的默认变量：ProgDir, WavePath, WebServer, ChannelNo, CallerID, CallerName, CalledID, CalledName, CallData and ServiceID。

ProgramDir(ProgDir)就如它的名字，只是指出你的 CONTAX IVR 应用程序所在的目录位置。例如：C:\program Files\ContaxIVR2.1。**WavePath**也是一个目录，但是是指.wav 文件所保存的目录位置。**WebServer**是指存有你要访问的 web 页面的计算机。它的默认值为 127.0.0.1，但你也可以使用一个 URL(统一资源地址 Universal Resource Locator)，例如 http://www.quickinfo.com.ph。

ChannelNo 指定运行中的 CONTAX IVR 应用程序所用的通道；默认值是 1。余下的变量 **CallerID**, **CallerName**, **CalledID**,

CalledName, **CallData** 和 **ServiceID** 的默认值为空，但在这些变量中填入相应的值对于电话的监测尤其有用。

除了默认的变量，你可以在变量表中定义新的变量，以使它们可以为整个电话流程所共享。

要增加变量，正如你在前面的章节中所做的那样，移到鼠标光标到变量表的最后一行并点击它来增加变量。指针会出现在最后一行的上面(不论是变量域上还是设定值域上)。按向下键创建一行新的变量域及设定值域。按向上键或删除键删除一个已经创建的变量域行。

在变量表中新建了一个变量后，你可以为这个变量设定一个初始值。再次点击设定值域来进行编辑。你将会注意到在空白设定值域的第一个字符位置出现了一个闪烁的光标，你就可以输入一个数字或是一个字符串作为这个变量的初始值。

通过使用一对方括号并将一个变量名括在其内，你可以设定其它变量或是字符串与其它变量的组合作为变量初始值(例如：设置变量 A 的初始值为**拨打[B]寻找[C]**，当 B=123，C=玛丽时，A=拨打 123 寻找玛丽。当 B 或 C 的内容改变时，A 的内容也会相应的改变。例如 B=1-800-7661234，C=帮助，那么 A=拨打 1-800-7661234 寻找帮助)。之后你可以使用**文字播放**(text to speech)功能来读出变量 A

7.1.2 分配一个变量

变量可以被分配到一个单元格的属性表设定值域内(位于属性表的右边部分)。当你这么做的时候，你必须在设定值中使用一对方括号“[]”将一个变量名括在其中，例如：[EmployeeID]，[UserPIN]，[ZIPCode]等。这样一来，程序就不会将这些变量错误地处理成一个文本字符串。

不论何时我们执行到一个电话流程中的某个单元格时，它的**指定变量值域**也同样会被执行一次。

7.1.3 使用表达式编辑器

你可以使用**表达式编辑器**对话框来创建一个表达式到属性表的设定值域中：

在属性表中单击选定**指定变量值**属性的设定值域。再次单击设定值域对它进行编辑。你将会看到在空白设定值域的第一个字符位置出现了一个闪烁的光标，同时一个带有省略号(三点)标记的按钮也出现在空白设定值域的右边边缘。你可以直接在此域中输入一个表达式。也可以双击空白设定值域或仅仅单击省略号按钮，就样表达式编辑器对话框就会出现。



输入一个常数

单击 **表达式编辑器** 对话框中常数区域的常数域，输入一个常数。

如果你要把数字作为字符串输入，你需要将此常数选择为字符型，而不是数值型。单击常数区域的向下箭头，从下拉列表中选择并单击 String。单击 **加入** 按钮将这个常数加到表达式窗格中。

除了使用 **表达式编辑器**，你也可以直接在属性表的设定值域输入字符型的数字。请记住要用一对单引号将这个数字括起来。（例如：' \$123 '）

如果你使用表达式编辑器输入一个数字字符串，将会自动添加一对单引号。

输入一个函数

在 **表达式编辑器** 对话框的函数区域的函数列表中单击选定一个函数，这个函数的描述信息就会出现在函数区域的下方位置。单击 **加入** 按钮将此函数添加到表达式窗格中。表达式窗体中出现了此函数，并带有几个垂直竖条(|)，由逗号(,)隔开。你可以将任何一个 | 用表达式替换，通过直接输入表达式或者再次使用 **表达式编辑器** 输入皆可。

输入一个变量

在表达式编辑器对话框中变量区域可用的变量列表中单击选定一个变量，单击加入按钮向表达式窗格中添加这个变量。

输入一个运算符

在表达式编辑器对话框中运算符区域单击一个运算符按钮向表达式窗格中添加这个运算符。

输入多个表达式到属性表

如果你希望在已经输入了一个表达式到表达式窗格之后输入更多的表达式，请按计算机键盘上的回车 (**Enter**) 键来转换到下一行，输入下一个表达式。重复上面的操作直到你想要的所有表达式全部输入为止。

当所有的表达式都已输入到表达式窗格后，单击表达式编辑器对话框中的 **确定** 按钮。你会所有的表达式都输入到了一个单元格属性表的设定值域内，并且相邻两个表达式之间用分号 (;) 隔开。

7.1.4 局部变量与全局变量

变量既可以是局部的，也可以是全局的。如果你的一个应用程序由一个主电话流程和几个子电话流程组成，你可能希望指定一个变量，使它完全应用于单个子流程中。这样的变量被称为 **局部变量**。

但是，你也可能要将变量中的内容传递给主电话流程来进行处理。这种被几个电话流程所共享的变量被称为 **全局变量**。

例如，你可能想要在一个电话流程中指定变量 `Email_doc`，让来话者选择希望通过 e-mail 发送给他的有效文档。那么当子流程退出时，你可以通过指定这个变量到主电话流程中的其它流程控件，将 `Email_doc` 变量传递过去，使得应用程序可以发送合适的文档。

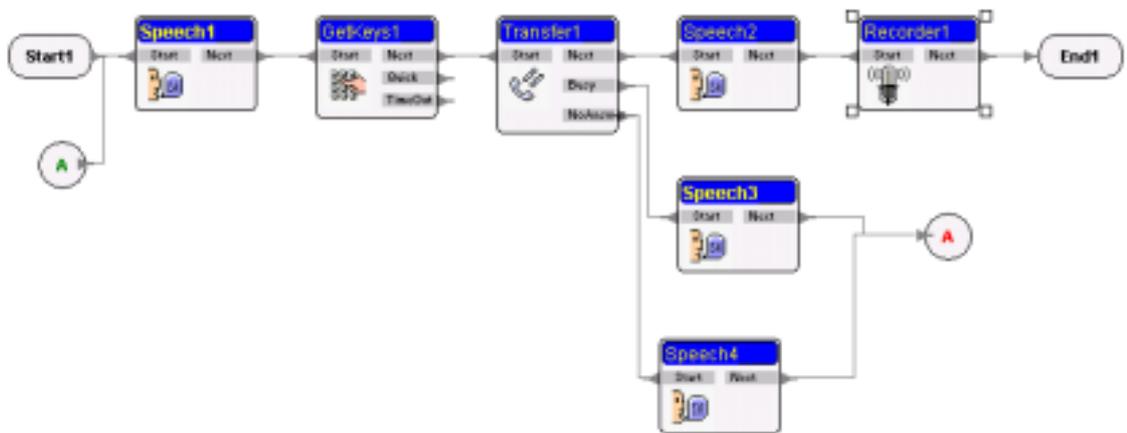
7.1.5 应用五：来话自动转接 (AutoTransfer) 中的其它变量

这个应用程序可以使你明白通过使用变量我们如何把工作变得简单。我们将再次回到你已经创建好的基本应用程序---来话自动转接程序 (AutoTransfer)。你可能已经觉得对于如此简单的任务，我们却做得很复杂。的确也是如此。但有一种更加简洁精巧的设计可以完成相同的工作。通过使用变量来保存已拨按键的内容---来话者选择的本地分机号码---将这个变量的值传递给 **转接电话**

流程的**目的号码**属性。你不必对每一个本地分机号码都用一个转接单元格，你仅仅需要一个就够了！

让我们来看看一个简单的应用程序，它实际上是我们之前做过的一个程序的变体。打开 Flow designer，单击文件(File)>打开(Open)。打开对话框出现，默认为 ContaxIVR2.1 的缺省安装目录。双击 Samples 文件夹，当出现一组文件夹时，双击 TransRec 文件夹。选择打开这个文件夹内的 TransRec.IVX 文件。

整个 TransRec 电话流程包含下面的内容：



然而，这个只含有很少电话流程控件的集合完成与来话自动转接带录音功能应用程序相同的工作。这是通过使用变量得以实现的！

让我们找出使事情变得不一样的变量。

第一站，单击 GetKeys1 控件并察看它的属性表和变量表窗体。注意，在变量表窗体上已经增加了一个 **ext_no** 变量。

CallerID	
ext_no	
rec_file	

以两样的方式，在属性表上你将看到**接收按键变量名称**属性的相应设定值为变量 **ext_no** 的值，指定为下面的内容：**[ext_no]**。这意味着所按的按键会保存于 **ext_no** 变量内。

现在选择转接单元格，再次察看属性表。注意到在**目的号码**属性的设定值域指定了[ext_no]，意味着所要的本地分机号码将等于来话者早先按的键，保存于变量 ext_no 内。所以当转接单元格执行它的操作时，它会提取**目的号码**属性的值以找出应该将电话转接到哪一条线。但是作为替代，它会直接提取 ext_no 变量并使用这个虚拟盒子内的值作为本地号码，即来话者想要联络的号码。

在电话仿真器中运行这个应用程序。仔细观察侦错窗口，你可能需要调整主窗体与侦错窗体间的分界线，以便可以更好的观察侦错窗体内的信息。

当你按下开始按钮，问候语播放完后，输入 123 作为本地号码。当你这么做时，你将会在侦错窗体中看到如下的信息：

```
侦错窗口
GetKeys 组件 : GetKeys1
使用者输入 = "123"
设定 ext_no = "123"
```

第一行是说 GetKeys1 控件单元格是当时是活动的。使用者输入是你按的键。第三行设置变量 ext_no 的值为“123”即你按的键。

当你在转接单元格为活动时按了 OK 按钮，你将会听到一个信息正被录音的提示。按另一个键录音就会结束，就像你早先制作的录音应用程序类似。

使用表达式

你可能已经注意到了 rec_file 变量。它只是简单地指定一个 .wav 文件的文件名，作为录音单元格录制完来话者的信息后的一个输出点。注意在 Speech2 属性表的**指定变量值**部分，它的设定值是一个长长的表达式

```
[rec_file]= [ext_no] + COPY( DATE, 6, 2 ) + COPY( DATE, 9, 2 ) + COPY( NOW, 12, 2 ) + COPY( NOW, 15, 2 )
```

表达式是计算机用以得到一定值的一个语句。它可以包含变量、常数及函数。你已经知道了什么是变量。常数是指任何一个固定的值。比如你设定下面的表达式：

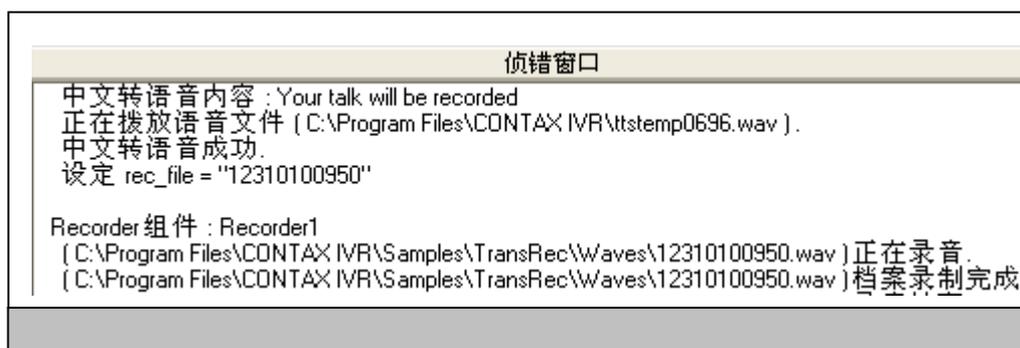
```
Emp_No=1234 , Emp_Name=Lina
```

第一个例子，1234 是一个数值型常数。第二个例子，“Lina” 是一个字符串型常数。

函数之间有 **COPY** 和 **DATE**，CONTAX IVR 已经将其内建于系统内，增强了电话流程的灵活性，使得设计者可以根据特殊要求定制应用程序。请参考用户指南中关于特殊函数的讨论。

现在我们不涉及针对具有丰富开发经验的应用程序开发者的函数使用的讨论。我们之所以要讨论在这里使用的表达式，是为了让你体会到表达式和函数是如何工作的。

当在执行应用程序时，如果你仔细观察侦错窗口，你会注意到在 TTS 信息 “Your talk will be recorded” 播放之后，程序生成如下一行：Set **rec_file**= “12310100950”。也就是将 12310100950 这个值赋给变量 **rec_file**。



然后，录音程序将文件 “12310100950.wav” 作为它的录音输出文件。

如果我们以下面的方式将这个 .wav 文件名的第一部分分解开，123-1010-0950，事情就开始变得清楚了。现在可以很明显地看出

文件名最前面的三个数字是来话者所按的分机号码。请记住表达式第一部分的内容：

[rec-file]=[ext-no]

第二部分的四个数字是日期，意思是 10 月 10 日。你看，当碰到 DATE 函数时，返回的日期是下面的格式：**yyyy/mm/dd 00:00:00**。所以你总共得到 16 个字符。**COPY** 函数返回字符串的一个部分 (Substring)，包含从第几个字符开始 (Index) 的连续几个字符 (Count)。也就是：**Copy(Substring, Index, Count)**。

这样，表达式 **COPY(DATE, 6, 2)** 将会返回 DATE 函数的一个子串。从第 6 个字符开始，只包含 2 个字符。如果你去计算 DATE 函数的格式，你会发现表达式 **COPY(DATE, 6, 2)** 仅仅返回 DATE 函数的月份部分。

第二个类似的表达式 **COPY(DATE, 9, 2)**，返回的正好是代表月中第几日的两个字符。

因此，假设今天是 2002 年 10 月 10 日，表达式 **COPY(DATE, 6, 2)+COPY(DATE, 9, 2)** 所返回的全部值是 1010。

最后部分的四个数字是电话被录音的时间。这为什么呢？NOW 函数返回一个与 DATE 函数相同的日期。然而尽管 DATE 函数在格式的最后 8 位字符提供了时间，即 00:00:00，数字保留为 0。另一方面，NOW 函数不仅仅返回实际的日期，也返回正确的时间。

这样，表达式 **COPY(NOW, 12, 2)** 返回 NOW 函数中的两个字符，从第 12 个字符开始，代表小时。

另一方面，表达式 **COPY(NOW, 15, 2)** 返回 NOW 函数中的两个字符，从第 15 个字符开始，代表分钟。以 **COPY(NOW, 12, 2)+ COPY(NOW, 15, 2)** 把两个表达式结合起来，返回确切的时间，以小时和分钟的形式。

你也许觉得非常的冗长，但这是程序员使用计算机及程序的方式，

类似 CONTAX IVR 产生和处理数据。只要有足够的想像力，逻辑思维 and 耐心，一个 CONTAX IVR 应用程序开发者可以处理绝大部分公司的电话与交互式信息系统的需求。

第八章：数据库的使用

另一个使得 CONTAX IVR 成为开发交互式信息系统应用程序强大平台的特性的是它与数据库协同工作的能力。在这一章，你将会掌握如下的内容：

数据库是什么？

如何利用 **DBQuery** 操作数据库、

8.1 什么是数据库？

简单定义，数据库是一个记录的集合。一条记录，换种说法就是一组相关的信息。可以将数据库简单地想象成系列被分割成组，类似一个表结构的数据。这个表结构由单独的数据元素组成，称为列或者域。单独的一组域被称为一个记录或者行。例如，创建一个包含雇员数据的关系型数据库，你也许会从一个名为 EMPLOYEE 的表开始，包含下面的信息条：姓名、年龄、职位。这三条信息组成了 EMPLOYEE 表的域，如下所示：

EMPLOYEE 表

姓名	年龄	职位
Will Smith	25	Electrical engineer
Dave Matthews	34	Museum curator
Jan Morrison	42	Chef
Bill Clinton	19	Student
Don Marcus	32	Game programmer
Becky Bouchard	25	Model

这 6 行就是表 EMPLOYEE 的记录。要从表中获取一个特定的记录，比如 Dave Matthews，用户应该发出指令让数据库管理系统获取姓名等于 Dave Matthews 的记录。如果指令要求获取记录中所有的域，那么将会返回给用户雇员的姓名、年龄和职位。

SQL 是一种语言，它告诉数据库获取这个数据。

注意：SQL 最初代表结构式查询语言 (Structured Query Language)，它的念法经常被我们参照“sequel”。它起初是为 IBM(International Business Machine)公司的 DB2(一个关系型数据库管理系统，或者称为 RDBMS)而开发的。

SQL 是一种**非过程语言**，与过程语言如 COBOL(Common Business Oriented Language)和 C 相区别。**非过程语言**着重**什么**而不是**如何**。例如，SQL 描述的是获取、删除或插入什么样的数据，而非如何执行这个操作。

使用数据库开发 CONTAX IVR 应用程序，并不需要你完完整整地学习 SQL 语言，你最多只需学习一些基本的 SQL 命令。同样地，你可能得通过常用程序来学习 SQL 语言的实际运用。

8.2 创建一个数据库

有许多常见的 SQL 执行程序，或是创建、管理及使用数据库的程序。每一个 SQL 执行程序都有它们自己特定的功能和局限。

其中一种广泛使用的 SQL 执行程序是 Microsoft Access，一个基于 PC 架构的数据库管理系统(DBMS)。它使用比较简单，具有图形用户界面的特性，以及向导功能以帮助用户创建和管理数据库。

另一家公司：Oracle。它有数个 SQL 执行程序---像 Oracle 9i 数据库---但是他们的产品通常被用在大公司的数据库中。

考虑到我们的目的，你可以使用 Microsoft Access 来创建数据库，使用**数据库控件 (DBQuery)**单元格访问、操作数据库。关于创建、

操作数据库的特定指导请参阅 Microsoft Access 的帮助文档。

下面是一个由 Microsoft Access 创建的简单数据库表：



ID	NAME
1	English
2	Japanese
27	Traditional Chinese
40	French
43	Thai
*	(自动编号)

创建数据库时要记住的重要方面

在设计你的数据库时，做文档记录或者制作类似如下的关键信息很重要：1)数据库的用途以及谁会用到它；2)描述数据库中每一个表的用途、内部结构，包括所有的域以及每一个域中包含数据的类型。

同样，你应该将数据分成各个独立的部分或者表，这样可以减少数据的重复。对于小型的数据库你也许会觉得这样做显得微不足道，但当你考虑到大型公司的存储空间不要求及其相应成本时，一个设计恰当的数据库不仅可以节省开支，同时可以使数据库的管理及数据获取变得简单容易。

数据源开放数据库连接---Open DataBase Connectivity (ODBC)

有那么多商家在出售自己的 SQL 执行程序，应该有一种方法可以使得在一个执行程序中创建的数据库为另一个执行程序所识别和使用。ODBC 是一个函数库，是被设计用来为数据库系统提供一个公共应用程序编程界面(API)。它通过一个库驱动来与数据库通信，就像 Windows 通过一个打印机驱动程序与打印机进行对话。

根据被使用的数据库，如果要与一个远程数据库联接，那么我们还需要一个网络驱动程序。

有了 ODBC，只需要做极少的修改，或者无需任何改动，你就可以用同样的代码对 Microsoft Access 表和 Infomix 数据库执行查询。然而，大多数厂商会在标准 SQL 上增加一些他们自己的扩展，因此当你使用一个新的数据源时，应该参考相关的文档。

ODBC 已经被许多产品所采用，你当前使用的 SQL 执行程序很可能就是与 ODBC 兼容的。

8.3 掌握数据库的功能

CONTAX IVR 可以联接数据库，扩展了原来电话系统本身---或者即使是使用了自动 PBX 系统 (PABX) 和应答机之后---所没有提供的信息的类别。联接存储在数据库中的信息的能力，使应用程序成为一个真正的交互系统---来话者可以获取他所选择的信息。

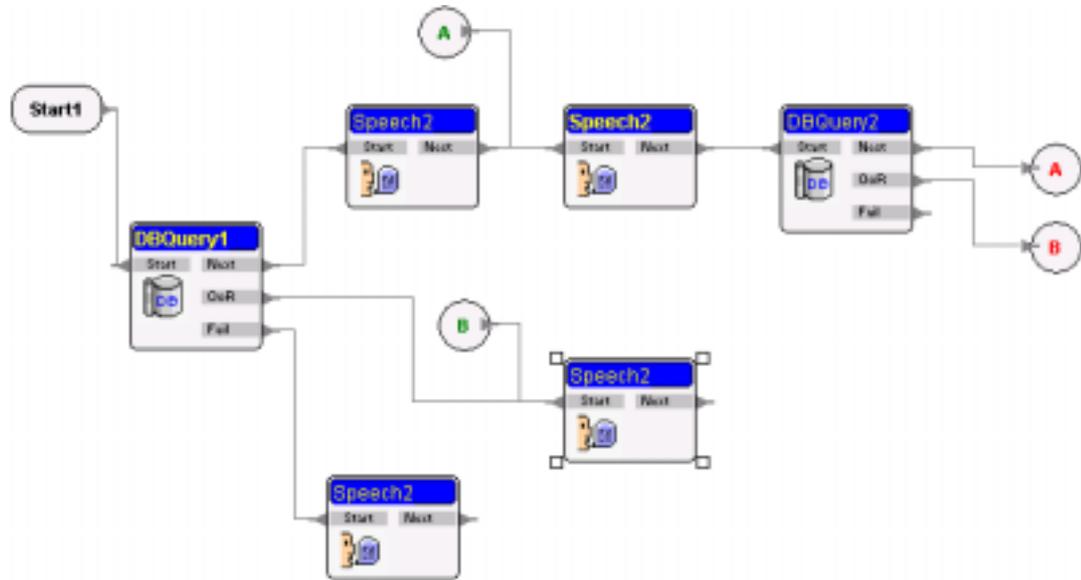
CONTAX IVR 能够通过 **数据库控件 (DBQuery)** 单元格访问到数据库内的信息。



当一个电话流程运行到这个控件单元格时，执行一个数据库查询操作，然后从它的 **Next** 指针退出。如果这个操作是以超出范围、未找到或者数据库为空，那么流程会从它的 **OoR** 指针退出。如果操作不成功，则流程会从它的 **Fail** 指针退出。

为了说明 **数据库控件** 是如何工作的，我们打开 CONTAX IVR Flow Designer，并单击文件>打开。在 ContaxIVR 2.1 文件夹中双击 Samples 文件夹。当文件夹被打开时，双击 DBConnect 文件夹并打开 DBConnect.ivx 文件。

你应该会看到下面的电话流程：



运行这个程序，它会告诉来话者数据库中的记录数，以及一个列的内容：语言的名称。

它是如何做到的呢？让我们来分析这个电话流程。

选定 DBQuery1 控件单元格，然后察看属性表窗体上的设定值。数据库控件的第一个重要属性是**连结名称** (Conn Name)。这是指要查询的数据库的名称，在本例中是 DB1。事实上，这是当**数据库控件**执行时计算机检查的第一个属性。

连结名称	Db1
------	-----

下一步，计算机寻找 SQL 命令。在本例中 SQL 命令的属性设定值如下：

SQL命令	Select * from Lang
-------	--------------------

注意： SELECT 与 FROM 一起使用是所有 SQL 命令中最基本的语句。Select...From 语句选择计算机所要查询信息的所在表。因而，**Select * from Lang** 语句表示此查询是针对 Lang 这个表进行的。

(请参考上面提供的第一个数据库例子)。语句中的符号 * 告诉数据库返回与 From 短语所描述的表相关联的所有列。

接下来，计算机会察看**操作型态(Action)**属性。我们可以看到这个属性在这里被设置成**查询(Query Open)**，意味着电话流程打开一个数据库，对它进行查询，并对它所包含的记录进行访问。

一旦在数据库中的查询成功，**数据库控件**会把所找到的记录数赋给**总笔数变量**，这里是赋给**[RCnt]**变量。在我们的例子中，计算机发现数据库中有 5 条记录，并把这个数字保存在 **RCnt** 内。

当电话流程从 DBQuery1 控件单元格的 *Next* 指针退出时，执行与它连通的 **Speech2** 单元格。这个单元格做两件事：播放一个 TTS 信息，根据变量 **RCnt** 的值，告诉用户这个表中有多少个记录，如下图所示：

文字转语音内容	There are [RCnt] records in this database.
---------	--

然后 **Speech2** 单元格将 **RecNo**(记录号)变量的值设为 1 并退出到 *Next* 指针。接着是下一个控件单元格，另一个 **Speech** 单元格，它也做两件事：1)使用 **RecNo** 变量的当前值加上 **Db1** 数据库中第一条记录的 **NAME** 列的值，播放一个信息。这是通过使用下面的表达式实现的：

文字转语音内容	Language [RecNo] is "[Db1.NAME]".
---------	-----------------------------------

此外，这个单元格的**指定变量值**属性需要增加，通过把它定义为 **[RecNo]=[RecNo]+1**，我们增加了变量 **RecNo** 的值。

接着是另一个**数据库控件(DBQuery2)**单元格，这里为什么需要它呢？答案是**操作型态**属性的设定值：**移至下一笔**。正如这个名词所暗示的，操作型态指示计算机访问下一个记录并循环回到第二个 **Speech2**，直到电话流程遍历了所有的记录为止。当计算机到达记录的结尾时位置，它自然返回了一个空记录。这时候**数据库控件**单元格会从它的 **OoR**(Out of Range) 指针退出，并且为此**文字播放**(Speech)单元格会播放相应的信息。

第九章：和传真、E-mail、短信息协同工作

此外 CONTAX IVR 还具有自动发送和接收传真、e-mail 以及使用与诺基亚移动电话相兼容的短信息系统的功能，使 CONTAX IVR 成为一个综合性的通信平台。在这一章中，你将会熟悉以下单元格控件的工作方式：

- 传真发送
- 传真接收
- 打电话
- 邮件发送
- 邮件接收
- 传送简讯

9.1 传真发送

CONTAX IVR 有了与传真功能的融合，你就可以设计出一个提供按需传真服务的应用程序：在一天中的任何时候发送传真，允许客户从可用的文档中进行选择并将选定的文档立即传送给他们或者稍后回拨。

发送一个传真

CONTAX IVR 通过它的**传真发送**模块发送一个传真文档。



然而，**传真发送**模块并不具有拨打一个传真号码的功能，这个工作是由一个前置控件单元格完成的。

在电话流程应用程序要向当前来话者发送传真时，**传真发送**模块可以被方便的用在提示来话者给出一个传真信号之后。但是，当应用程序要进行一次回拨---稍后发送一个传真到之前来话者的

号码，或者是另外一个号码时---通常通过使用**打电话**控件单元格来完成传真发送工作。



打电话单元格事实上是一个**子流程**控件单元格。当被触发后，它拨打一个本地分机号码或者一个外部线路并执行一个新的电话流程。之后，这个模块会从它的 *Next* 指针退出。

传真格式

传真发送和**传直接收**控件单元格支持下面的文件格式：

- Text (.txt)
- TIFF (.tiff)
- JPEG (.jpg)
- GIF (.gif)
- BMP (.bmp)
- PCX (.pcx)
- DCX (.dcx)

当**传真发送**控件单元被击活时，它会察看**来源档案**属性---准备要发送的图像文件---以上面特定格式中的任意一种。

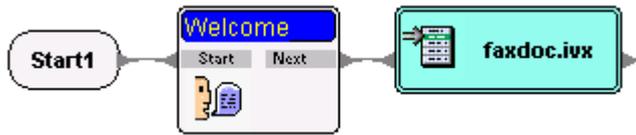
9.2 应用六 :传真自动应答(也称按需传真 Fax On Demand)

为了说明按需传真(是如果工作的，我们运行 CONTAX IVR Flow Designer 打开 Samples\FOD 文件夹，再打开 **Sendfax1.ivx**。

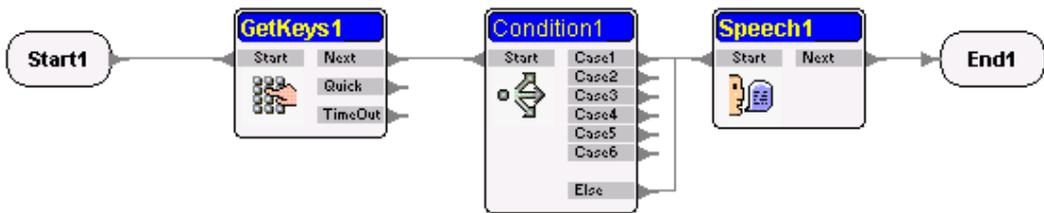
这个按需传真应用程序实际上由三个独立的部分组成。**Sendfax.ivx** 是主应用程序，它调用另外两个应用程序：其中一个是一个**子流程**控件，另一个是一个**打电话**控件。

运行这个应用程序

你会发觉在拨号及欢迎信息播放完之后，电话流程会执行名为 `faxdoc.ivx` 的子流程



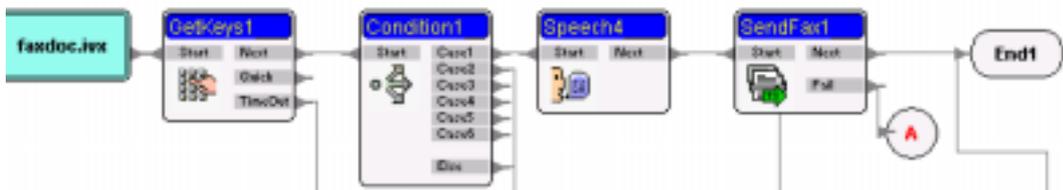
`Faxdoc.ivx` 流程包含下面的电话流程控件单元格：



这个子流程告知来话者共有哪些有效的文档，使他可以在电话按键面板输入一个特定数字来作选择，在此之后子流程结束并返回到主流程。但请留意在 `Condition1` 中，变量 `faxdoc` 被指定为所选定的文档。之后你会发觉同一个变量被传递到了主应用程序，用来确定向来话者发送正确的文档。

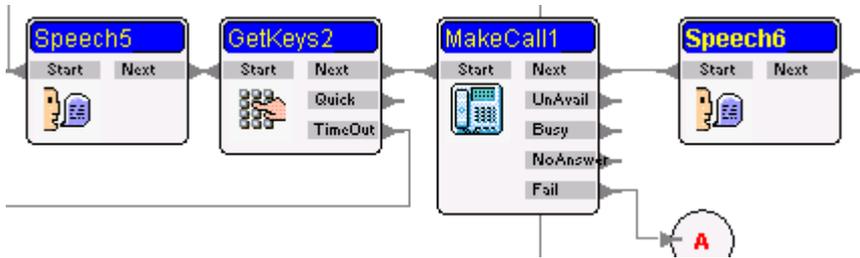
指定变量值 `[faxdoc] = IF([DocID] = 1`

当 `faxdoc.ivx` 退出后，主程序 `sendfax1.ivx` 继续执行。



`GetKeys1` 让来话者选择立即传真文档还是稍后再回拨，来话者的输入由 `Condition1` 进行处理。如果来话者想要在当前线路上立即传真，`Speech4` 模块提示他按传真机上的开始按钮送出一个传真信号，紧接着 `SendFax1` 模块将传真发送出去。注意，从 `GetKeys1`

到 SendFax1 之间的每一个控件单元格都是使用全局变量 faxdoc 来确定所要发送的传真文件。



如果来话者希望在另一个号码上接收传真，打电话控件单元格可以处理这种情况。在上面的例子中，Speech5 模块提示来话者输入他的号码(传真号)，由 GetKeys2 单元格进行处理并将它传递到 MakeCall1 控件单元格。后者拨打来话者所提供的号码并触发另一个电话流程---在语音流程文件名属性中指定的 sendfax2.ivx。

语音流程文件名	sendfax2.IVX
---------	--------------

Sendfax2.ivx 事实上是一个非常简单的电话流程，仅仅只是在 MakeCall1 所拨打的线路上发送传真。它包含以下的单元格：



9.3 接收一个传真

与自动传真发送类似，CONTAX IVR 也自动处理传真的接收，通过使用传真接收控件单元格来实现。



传真接收控件单元格接收呼入的传真并将传真到一个指定的文

件。传真文件将保存到**档案存储路径**属性所指定的目录中，比如像 C:\Recv_docs。

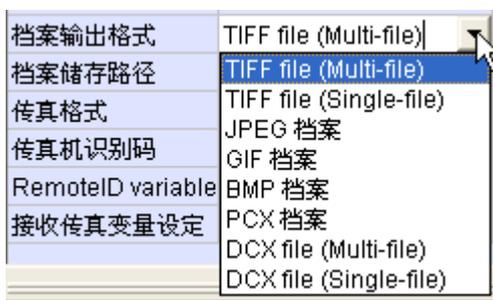
传真档案的存储文件名是由**文件名编码方式**属性决定的。你可以从中选择**信道编号-序号-页码**或者**信道编号-月份日期-页码**模式。



如果你选择了**信道编号-序号-页码**模式，每一个传真页面都会根据信道编号、序号(系统中所有传真事件的累加值)、页码作为文件名自动保存。因此一个以 5-28-1 作为议文件名的传真表示由信道 5 接收的序列号为 28 的传真的第一页。

在**信道编号-月份日期-页码**模式下，每一个传真页面将会根据所使用的信道编号、日期和时间、页码作为文件名进行保存。一个以 7-04101731-1 作为文件名的传真表示在 4 月 10 日 17:31 由信道 7 接收的传真的第一页。时间标记是在单元格接收传真完毕，保存文件之前。由于一次传真的处理过程---包括握手信号---超过一分钟，所以不会发生一个传真文件名被重复保存的情况。

接收到的传真将会以在**传真发送**模块中可使用的各种格式中的其中一种格式进行保存。你可以借助**档案输出格式**属性指定文件的类型。



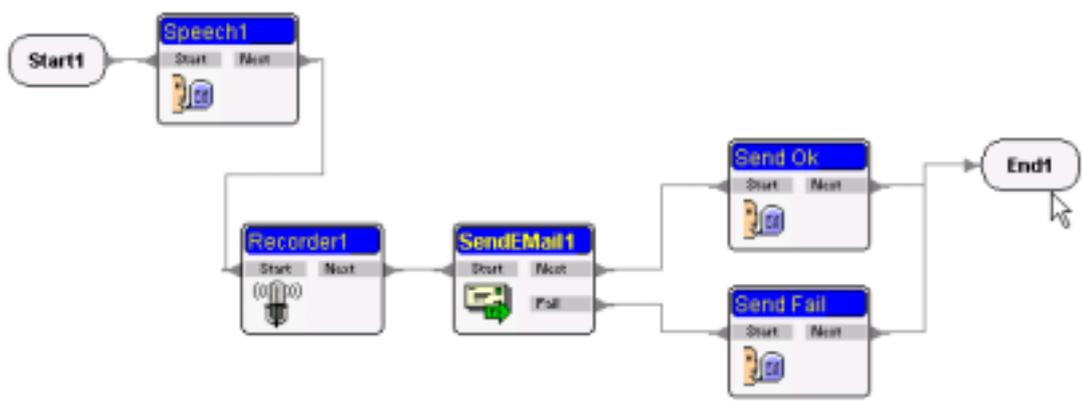
9.4 使用电子邮件(E-Mail)

除了到目前为止我们已经讨论过的功能，CONTAX IVR 此外还具有通过**邮件发送**和**邮件接收**控件单元格自动发送和接收邮件信息的便捷性。

为了更好地说明这些功能，让我们仔细分析另一个例子。

9.4.1 应用七：发送邮件

进到 CONTAX IVR 的安装目录中，打开 Samples 文件夹。双击 Email 文件夹并打开 SendEmail 文件夹，再打开 SendEmail.ivx 文件。你应该会在 CONTAX IVR Flow Designer 中看到如下的内容：



这个应用程序提示来话者（通过 Speech1）录制一条信息（Recorder1），然后，电话流程自动将信息以 e-mail 方式发送给来话者。

你已经熟悉了前面两个控件单元格，所以让我们把注意力集中到**邮件发送(SendEmail)**控件上。

这个单元格通过在属性窗体中指定的 SMTP(简单邮件传输协议 simple mail transfer protocol)服务器发送邮件。它会检测你是否已经连接到了 Internet，如果还未连接，拨打你的默认连接并发送邮件。发件人、收件人邮件地址以及抄送地址、密件抄送地址都是在属性表内各自的设定值域中指定的，如下图所示：

寄件者邮件地址	contaxivr@elettech.com
收件者	contaxivr@elettech.com
副本	
密件副本	

要给多个收件人发送同样的邮件，请指定多个邮件地址，并用逗号隔开。

注意：你可以使用一个字符串变量代替一个邮件地址或是多个邮件地址。例如，用[Sales]代表所有销售部门的邮件地址。

邮件地址变量可以在数据库查询操作中使用，创建功能强大的电子商务(e-commerce)或虚拟商务(v-commerce)工具。例如，当来话者买了一个产品后，会自动发送一封邮件给产品供应商和/或经销商。邮件的主题和/或内容(正文)可能会根据收件人的不同而有所区别。

你同样需要在属性表中指定邮件的**主题**和**内容**。要写邮件的正文，请双击内容属性的设定值域然后单击省略号按钮。打开了一个**文字编辑器**，你可以将信息输入进去，或者被你在其它邮件程序甚至是文字处理程序中准备好的信息粘贴到文字编辑器内。如果有需要，你也可以通过属性表窗体指定一个附件。

为了成功的将邮件发送出去，你必须在属性表中指定你所使用的SMTP服务器的值。CONTAX IVR 默认使用 25 作为 SMTP 服务器的端口。

9.4.2 接收电子邮件

尽管发送电子邮件相对比较简单，但是接收电子邮件相对要复杂一点。你一定记得邮件接收(RecvEMail)控件单元格有两个基本功能：1)询问服务器是否有你的邮件；2)从你的邮件箱中收取邮件。



你可以在单元格**操作型态**属性中指定你要完成的特定操作(查询邮件服务器或者收取邮件)。

邮件接收 (RecvEMail) 控件单元格从一个 POP3(Post Office Protocol)服务器(由 **POP3 服务器**属性指定)收取电子邮件。默认的 POP3 服务器端口是 110。请咨询系统管理员你的 POP3 服务器名及它所使用的端口。

你同时应该指定电子邮件的**使用者帐号**及正确的**密码**。作为可选的办法，你可以指定一个变量来得到来话者通过**资料输入** (GetKeys)单元格所输入的信息，并将它与含有使用者帐号及密码的数据库进行比较。

在属性表中，你也可以指定变量来存储发件人(发件人变量)、邮件日期(日期变量)、主题(主题变量)、邮件正文(内容变量)、附件(附件变量)---如果需要---以及你想要保存附件的目录路径(附件存储目录)。如下面的例子所示：

Retrieve content : RecvEMail	
属性	设定值
接收内容	False
邮件总数变量	
寄件者变量	[Sender]
日期变量	[EMailDate]
主旨变量	[Subject]
内容变量	[Body]
附加档案储存目录	[ProgDir]
附加档案变量	[Attachment]

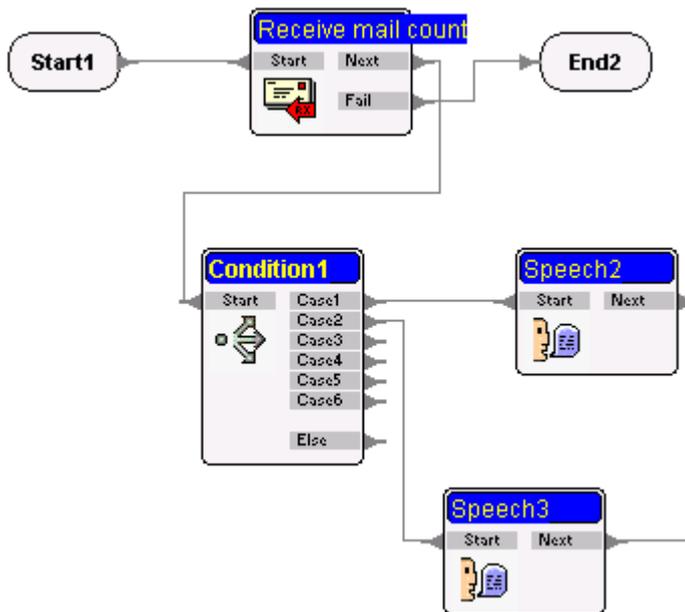
为了使你更加明白，让我们再分析一个例子。

9.4.3 应用八：邮件接收

你可能想知道，我们用 CONTAX IVR 应用程序来接收电子邮件的实际意义在哪里？举个例子来说，假设你是一个销售员并且此时在外工作。现在，你的一个客户发送了一封电子邮件给你，写明了他所要向你公司买的产品，但是你没有 PC 可供收取邮件，也没有可供你登录 POP3 服务器的来读取你的邮件信息的远程终端设备。利用邮件接收应用程序，CONTAX IVR 能够收取你的邮件。用一部按键式电话你可以拨打到你的公司，利用 TTS 功能将信息读给你。下面的电话流程样例 RecvMail.ivx 说明了这个功能。

打开 CONTAX IVR 目录中的 Samples 文件夹，进入 Email 文件夹并双击名称为“RecvMail”的文件夹，点击 RecvMail.ivx 以便在 Flow Designer 中打开这个应用程序。

你将会发现紧随着 Start1 单元格的是一个标为“Receive mail count”的邮件接收单元格。



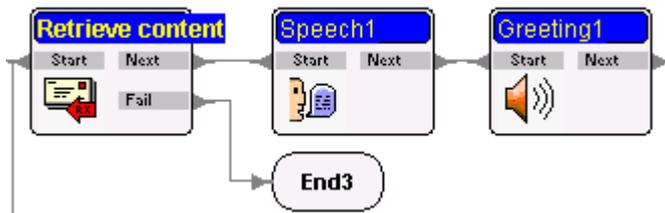
如果你单击这个单元格并察看它的属性表，你会发现它只是要查询 POP3 服务器中的邮件数量，由操作型态属性的设定值指定。

操作型态	查询邮件数目
------	--------

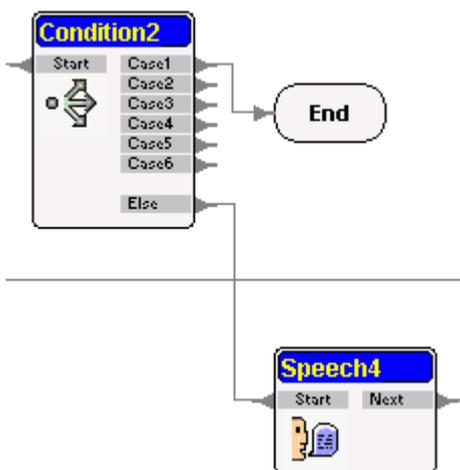
之后流程激活 Condition1，它使应用程序分成两个独立的分支。如果没有邮件，电话流程就触发 S 文字播放单元格来告诉呼入者没有他的任何邮件信息。但是如果有收到的邮件信息，Condition1 就从它的 Case2 指针退出并激活 Speech3 单元格，告诉呼入者他有多少封邮件。

另一个 **邮件接收** 控件单元格 (收取内容) 被执行 --- 它的操作型态属性被设定为 “**接收邮件**” --- 从 POP3 服务器上收取收到的邮件。注意在属性表中，变量 [a] 被指定为 **邮件索引** 属性的值。指定 **变量值** 属性的设定值为：**[a]=[a]+1**。

注意：指定的变量是用于增量 --- 依次累加 --- 已经被处理过的邮件信息，这样当所有的邮件信息都已处理之后电话流程会停止处理。稍后你会再次碰到这个变量。



Speech1 控件单元向来话者读出电子邮件信息的详细内容 (日期、收件人、主题、附件)；下一个单元格播放附带邮件中的 wave 文件。接下来，电话流程执行一个 **条件** 单元格。



请仔细留意，Condition2 单元格将[a]的值(我们正在跟踪的值)与 [MsgCount]变量的值(之前由第一个邮件接收控件单元格所获取的值)进行比较。

Case1	[a] > [MsgCount]
-------	------------------

如果[a]的值还没有超出 [MsgCount]的值，电话流程就会回到**邮件接收**控件单元格处理下一条邮件信息，一直到 POP3 服务器中特定邮件地址的所有邮件信息都被处理完毕。

9.5 使用 SMS(短消息系统 Short Messaging System)发送信息

随着移动电话的使用越来越多，使得公司能够与不在办公室内的雇员或客户取得联络。然而，拨打移动电话的花费通常会比拨打本地电话高很多。在许多情况下，公司利用移动电话短消息(SMS)功能的特点来发送信息所要支付的费用会很少。

CONTAX IVR 已经含有了发送短消息的能力，这是通过一个**发送短信 (SendSMS)**控件单元格实现的。



这个单元格发送短消息 (Short Message Service) 到一个目标移动电话，然后从它的 **Next** 指针退出。如果发送短消息失败，那么它会从 **Fail** 指针退出。

这个控件单元格可以用下面的构架发送短消息：



你可以将上面所示的基本结构作为一个**子流程**添加到你现有的应用程序中。

但是，这项特性只和使用十分普及的诺基亚(Nokia)移动电话相兼容。兼容诺基亚移动电话型号包括以下类型：

Nokia 3xxx (3210, 3310 etc.)
Nokia 51xx
Nokia 61xx
Nokia 62xx
Nokia 71xx
Nokia 8xxx

在**手机型号**属性中指定你当前所使用的移动电话的型号。移动电话安装在运行时(Runtime)服务器上并与服务器上的其中一个串行(COM)端口相联。

单击**装置连接端口**属性，选择运行时服务器(Runtime Server)上的一个串行端口，用于从这台服务器上通过移动电话发送短消息。

你总共能够连接 8 部移动电话，使用 COM1 到 COM8 端口。可以通过合适的数据线缆进行连接，也可以通过红外线端口连接。你应该在**简讯装置连接方式**中指定正确的连接类型。(请咨询移动电话制造商合适的连接方式)。

短信发送(SendSMS)控件单元格拨打你在**目的号码**属性中指定的移动电话号码。这里，你可以通过一次**数据库查询**操作使用一个变量保存想要拨打的移动电话号码。

要编排短消息的内容，你可以双击**内容**属性，再点击省略号按钮打开**文字编辑器**，在**文字内容**框中输入内容即可。通过在**可用的变量**中选择变量然后点击**加入**按钮，你可以在文本中插入变量，这样就可以得到一个可编程的内容。

你也可以从一个文件中复制内容然后再粘贴到**文字内容**框中。

你在**文字内容**框中输入字符不存在最大字符数的限制，但是绝大多数的移动电话限制信息的长度为不超过 140 到 160 个字符(取决于移动服务商)。

如果是使用双字节移动电话装置(比如：在中国大陆、台湾、香港、日本、韩国等地区)，字符数的限制大约为 70。移动电话不会发送超出长度限制的所有额外字符。

9.6 Web 集成

CONTAX IVR 也具备连接 Web 网络来扩展其功能的能力。只要通过**呼叫 URL(Call URL)**控件单元格即可。



这个单元格将电话流程连接到一个你想要呼叫的 URL(统一资源地址 Uniform Source Locator)或者说是一个 Internet 地址。

Internet 地址或者说是 Web 页面可以是一个活动服务器页面(ASP),它包含 HTML(超文本标记语言)和用 VBScript 或者 Jscript 写成的内嵌式的程序代码。当微软的 Internet Information Server(IIS)遇到由这个单元格通过浏览器发出的 ASP 页面请求,它会执行内嵌的程序。这可以使电话流程通过所呼叫的 Web 页面与数据库和其它程序进行交互。**呼叫 URL** 控件单元格使得应用程序可以获取数据或信息。除此之外,如果没有**数据库查询**控件单元格,这是不可能做到的。

为了使得控件单元格可以呼叫正确的 Internet 地址,请在**Web URL** 属性的设定值域中输入所要的地址。

学习 CONTAX IVR 中 **Samples** 目录下所提供的例子,你会明白**呼叫**

URL 控件单元格是如何工作的,以及你如何利用它为你的应用程序提供更丰富的功能。打开 **CallURL** 文件夹点击同名的 .ivx 文件。

这个应用程序调用一个 ASP 页面,计算从现在到你的生日还有多少天数。

应用程序使用**呼叫 URL** 控件是 CONTAX IVR 应用程序开发中一项比较高级的技术,对于资深的程序员来说应该会涉及到开发指南中更加广泛的内容。

9.7 其它控件单元格

除了前面已经讨论过的,CONTAX IVR 提供了其它的控件单元格,使得应用程序开发者能够处理几乎所有可以想得到的通信需求。下面的控件单元格可以被用于设计比较复杂的应用程序。现在,我们将会对它们的特征进行简单地讨论。

9.7.1 多方会谈



这个单元格接通一路分机或者一路外部电话。

你可以在**目的号码**属性中指定所要的分机号码或者外部线路号码。比如:2385 或者+86 (21) 766-1234。

如果你在 **Runtime Engine** 中设定 0 或 9 作为**外线前置码**,在这里你就不需要输入接通外线的前置码了。

如果你要拨打的是一个本地号码,但是你的**目的号码**中却包含了国家代码和地区代码,那么倘若你在 Windows 系统中(**开始 -> 设置 -> 控制面板 -> 电话和调制解调器选项 -> 拨号规则/我的位置**)已经设定了这些内容,系统将会单单为你拨出本地号码。

如果你确信这个号码是本地号码，你可以在**目的号码**属性中只输入本地号码，而不必附带国家代码和地区代码。

9.7.2 线路交换



这个单元格被用来在 T1/E1 线路中连接两个信道。由**目的信道**属性的设定值来指定所要的信道号。

在**交换型态**属性设定值中，你可以从下拉列表的 3 个选项中进行选择。

1. **Connect**: 单元格连接到另一个信道然后从 **Next** 指针退出。
2. **Monitor**: 单元格监控另一个信道然后从 **Next** 指针退出。如果另一个信道挂断了，那么单元格会从 **Close** 指针退出。
3. **Disconnect**: 单元格断开与另一个信道的连接然后从 **Next** 指针退出。

当你完成了应用程序的开发，在将它用于实际工作环境前，你必须对它进行适当的测试，以去除程序中的错误和缺陷。

第五部分 程序的运行和监控

与 .com 和 .exe 文件不同，CONTAX IVR 应用程序不是可执行文件，因此它们无法自己单独运行。在下面的章节中，你将会学习到如下内容：

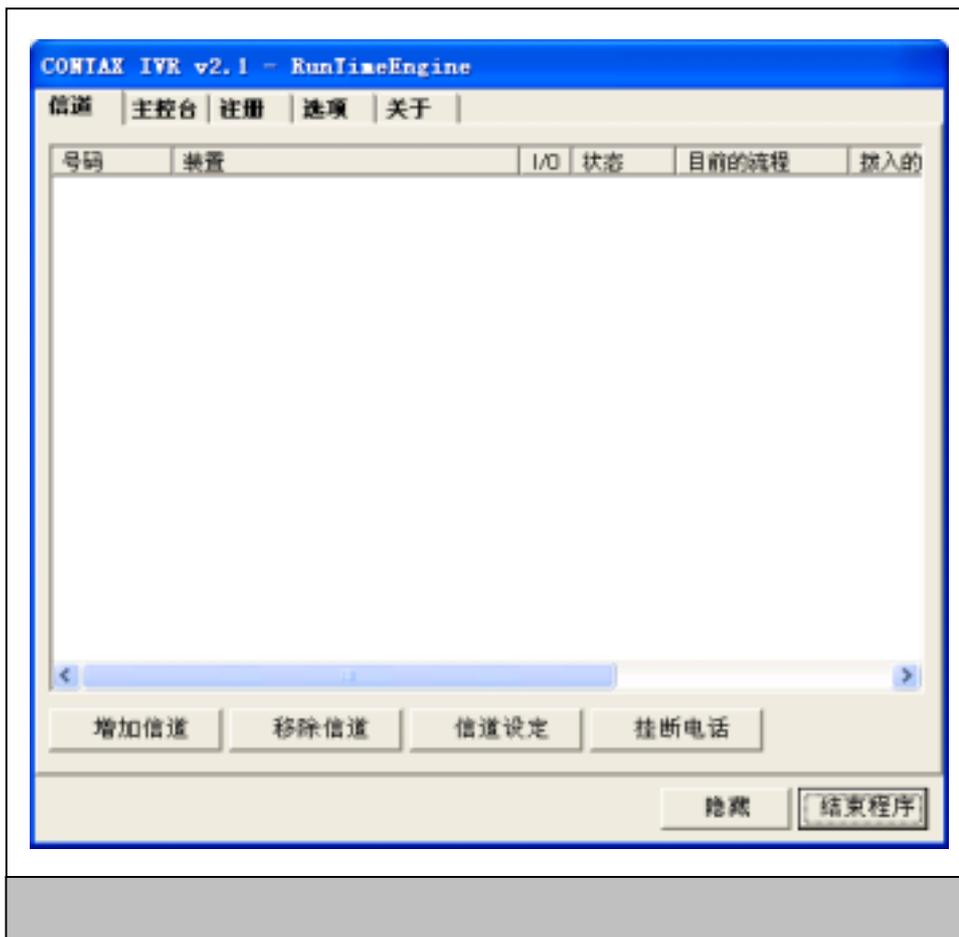
- 通过 Runtime Engine 运行程序
- 监控应用程序

第十章：程序的运行

你可以选择要在 CONTAX IVR Runtime Engine 中运行的程序。如果你点击了 Runtime Engine 的快捷方式，它会以后台方式运行，你只会在任务栏通知区域内看到一个小的电话手柄图标：



但是，如果你双击这个通知区域图标，Runtime Engine 面板就会弹出来，如下所示：



点击增加信道按钮打开对话框：



选择适当的信道，它就会添加到 Runtime Engine 面板中：



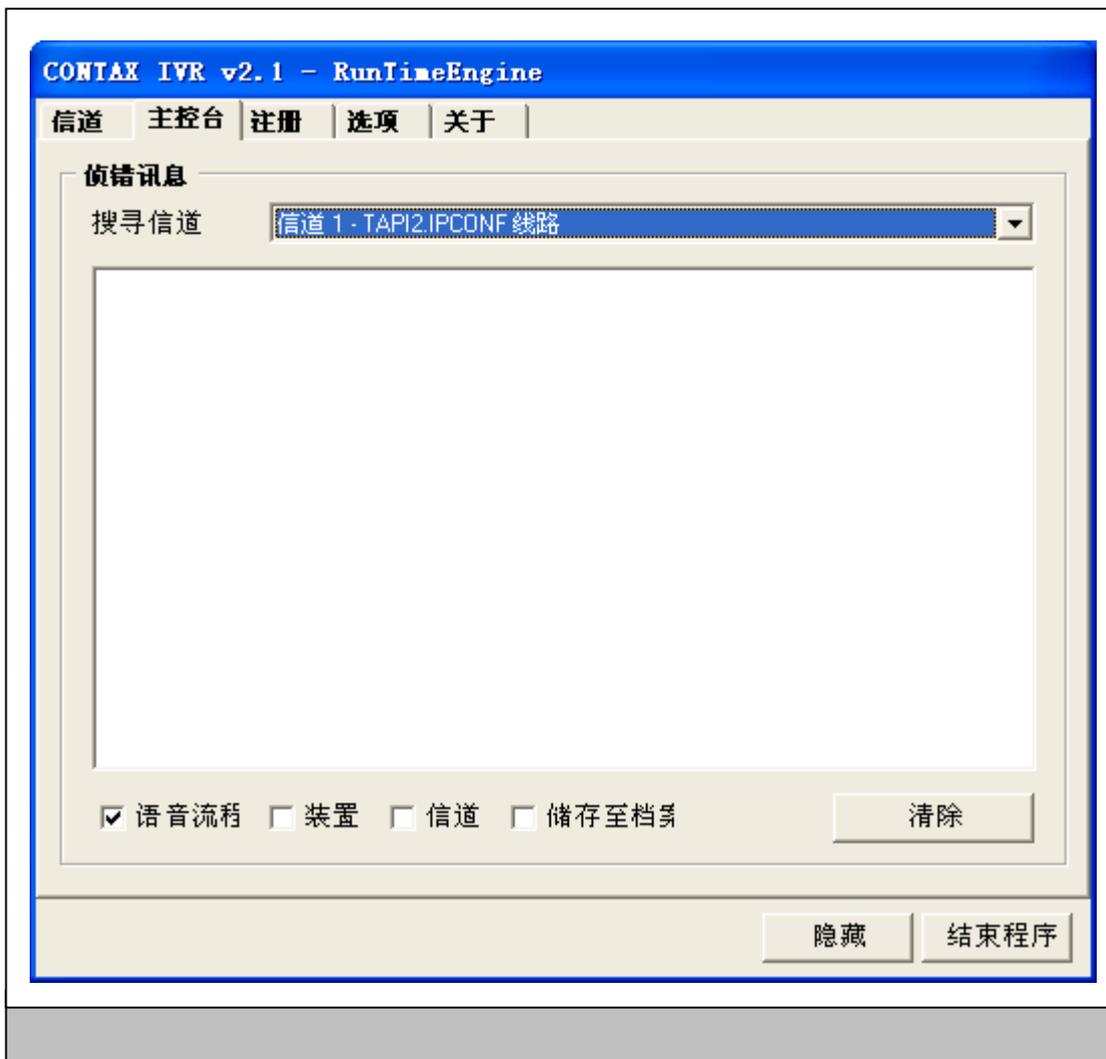
根据你取得的授权和所用的语音卡，你可以在同一时间运行多个活动信道。要运行指定的电话流程，请单击信道，然后单击**信道设定**按钮。这时，会出现下面的对话框：



单击**拨入流程档案**右边的省略号按钮(...), 打开一个文件对话框。进到你保存电话流程应用程序(.ivx)的目录中, 并单击你所要的电话流程。单击打开, 应用程序相应的目录就会添加到**拨入流程档案**文字框内。

在**电话号码**框中指定适当的本地分机号或外部线路号码(PBX/PSTN)。如果需要, 请在其它文字框中填写相应的内容(其它文字框的设置请参考 CONTAX IVR 用户指南)。单击**确定**按钮, 你的应用程序就会启动并处于运行状态!

要对运行中的应用程序进行监控和侦错, 请点击**主控台**标签。从**搜寻信道**下拉列表中单击选择一个信道, 你将会从**主控台**窗体中看到这个信道的运行时事件。



共有 4 个复选框可供选择：

- 语音流程：** 显示电话流程组件记录
- 装置：** 显示设备信息
- 信道：** 显示信道信息
- 储存至档案：** 将记录数据保存到 ChX.log 文件(文件路径与 RuntimeEngine.exe 相同)