

Notes

编程疑难详解

罗琳 陈怡之 陈斌 张宏林 等编著



人民邮电出版社

J44



高手指路系列丛书

otes 编程疑难详解

罗琳 陈怡之
陈斌 张宏林 等 编著

人民邮电出版社

图书在版编目 (CIP) 数据

Notes 编程疑难详解/罗琳等编著。—北京: 人民邮电出版社, 2000.9

ISBN 7-115-08656-7

I. N... II. 罗... III. 计算机网络—应用软件, Notes—程序设计 IV. TP393.09

中国版本图书馆 CIP 数据核字 (2000) 第 41099 号

高手指路系列丛书

Notes 编程疑难详解

◆ 编 著 罗 琳 陈怡之 陈 斌 张宏林 等
责任编辑 张立科

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ pptph.com.cn

网址 <http://www.pptph.com.cn>

北京汉魂图文设计有限公司制作

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 30

字数: 749 千字

2000 年 10 月第 1 版

印数: 1—5 000 册

2000 年 10 月北京第 1 次印刷

ISBN 7-115-08656-7/TP·1731

定价: 46.00 元

前言

Lotus Notes 是 Lotus 公司推出的群件系统。它不仅仅是一个应用软件，更是一个开发平台，丰富的 Notes 公式函数和功能强大的 LotusScript 可供设计者灵活地定制应用。Domino 系统将 Web 站点与 Notes 数据集成在一起，使开发 Web 应用变得简单易行。Lotus Notes 的内置邮件系统，文档数据库和工作流使协同工作更加容易。正是由于具有上述这些特征，该系统至今已成为一个重要的商业软件 and 开发平台。

本书针对设计者在开发应用中可能遇到的实际问题予以详细解答。问题的解答分三个步骤进行：“钥匙在此”讲述解决问题的关键技术或技巧的基本原理；“跟我来”给出切实可行的实现步骤或例程；“再送礼包”或给出这个问题的另一种实现方法，或略述相关问题的解决方法。本书分为 12 章，第一章讲述表单的设计技巧；第二章介绍 Notes 数据库的基本组成部分——文档，包括了 Notes 中典型的主文档——答复文档结构和文档集等内容；第三章讲述 Notes 的文档/视图结构以及它们的交互作用；第四章讲述了对数据库的操作，如创建副本、分析数据库、访问限制等问题；第五章介绍设计 Notes 界面的技巧以及一些特效的实现方法；第六章主要介绍 Notes 和 Windows 平台的交互，读取客户端或服务器的系统信息的方法；第七章讲述在 Notes 中使用链接和嵌入对象以及外部应用程序共享信息等技术；第八章讲解使用 ODBC 和外部数据库进行数据交换的方法；第九章介绍 Domino 服务器上的用户管理和数据库安全控制等相关的问题；第十章讲述了如何为 Notes 应用增添邮寄特性；第十一章介绍如何使用 Notes 创建 Web 应用以及将 Web 站点与 Notes 数据集成的一些问题；第十二章介绍了使用 LotusScript 和公式语言编写程序的一些其他技巧。

本书主要由罗琳、陈怡之、陈斌、张宏林编写，参加编写和资料整理的人员还有：苏超、黄昊文、高峰、吴杰、黄益文、赵立诚、吴泉、黄文歧、赖聿钒、何彬、曾俊豪、刘铎、纪锦修、李典川、沈金灿、王宝祁、傅丽萍、顾耿元、叶清婉、王志贞、卢向山、王津善、单芳、林之洋、李政、张敬、吴伯元、李明、王可、福盛海等。

由于作者水平有限，书中缺点与错误在所难免，我们诚恳希望读者批评指正。

第一章

表单与域

热点透视

表单决定了 Notes 中文档的显示方法，是用户查看和输入信息的框架，域是表单中储存数据的独立元素，两者都是设计 Notes 应用程序的基本要素。合理地设计与安排域和表单，能够为用户提供良好的操作界面，方便数据的输入、输出、查询等工作。其实除了技术性问题以外，表单和域的处理有时更需要设计者的智慧和经验，许多看上去很困难的问题，其实只要在设计上利用一些技巧就可以很容易地解决。本章针对设计者在表单和域的设计与操作中常见的编程问题进行详细解答，并涉及了有关通过域显示的文档条目(Item)，以及较特殊的数据类型即 RTF 文本类型的条目的问题。



对同一行的表单元素使用不同的隐藏公式



遇到难题

应用隐藏属性可以将不需要显示的内容暂时“藏”起来。但是，Notes 中的表单元素是以段落为单位隐藏的。如何对位于同一行的表单元素应用不同的隐藏条件呢？



钥匙在此

表单设计元素中的“隐藏”属性是一个非常实用的特性。我们不但可以编辑隐藏公式，使文档在不同情况下以不同的方式显示，一些设计巧妙的隐藏域还可以做许多仅仅通过编程做不到的事情。表 1-1 中列举了所有隐藏选项：

表 1-1

隐藏选项

选项名称	说 明
Notes 客户程序	通过 Notes 客户程序打开文档时，隐藏信息。
Web 浏览器	通过 Web 浏览器访问文档时，隐藏信息。
预览读	在预览窗格中阅读文档时，对用户隐藏信息。
预览编辑	在预览窗格中编辑文档时，对用户隐藏信息。
打开读	在读模式下打开文档时隐藏信息，隐藏的信息也无法被打印。
打开编辑	在编辑模式下打开文档时隐藏信息。
打印	打印文档时信息不可见。
拷贝到剪贴板	在剪贴板拷贝或粘贴操作中，信息不可见。
公式为真时隐藏段落	可以使用自己设计的公式判断段落的隐藏条件。如果要永远隐藏一个段落，可以选择此项，并使用@True 作为判定公式。

然而表单中的隐藏效果并不是以设计元素（如域、按钮等）为单位，而是以段落为单位应用的。因此，要令位于同一行的设计元素以不同的方式隐藏，还需要应用以下技巧：

设 A 和 B 是处于同一行的两个元素，要为 A 和 B 应用不同的隐藏公式，可以在 A 和 B 的下方创建一个和 A 相同的设计元素 A'，下一行再创建一个和 B 相同的设计元素 B'，再为这三行分别设计不同的隐藏公式。由于被隐藏的信息不会占用任何空间，所以显示时这三行将出现在同一个位置。

当 A 或 B 是域时，还需要通过计算公式等方法使这三行内容的显示同步。具体实例请看下一段。



跟我来

这个例子在域旁边创建一个在读模式下隐藏的按钮：

1. 将域和按钮所在的段落设置为“预览读”、“打开读”时隐藏。
2. 在下一行创建一个相同的“显示时计算域”，计算公式设定为上一个域的值。在隐藏附签中选择“预览编辑”和“打开编辑”，如图 1-1 所示。

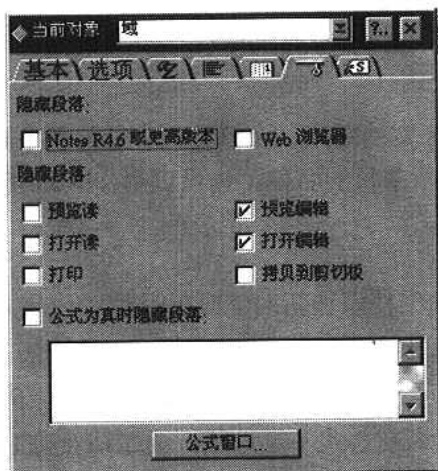


图 1-1 隐藏选项



使表单中的某些元素仅在对话框中隐藏



遇到难题

Notes 表单也可以被当作对话框来设计使用。当表单同时用作对话框和文档时，如何判断表单的用途并据此决定表单元素的隐藏条件呢？



钥匙在此

这个问题显然不是“打开读”、“打开编辑”等简单的隐藏条件能解决的，也没有现成

的属性说明表单是否正在以对话框模式打开，这就再一次需要应用文档条目来解决。

表单元素的隐藏公式可以直接访问文档条目。因此在弹出对话框时为文档创建一个临时条目，这个条目标志着当前文档正在对话框中被打开。根据这个临时条目的值编写隐藏公式，就可以实现仅在对话框中隐藏的效果。

在文档中创建域可以使用的方法和函数包括：NotesItem 类中的 New 函数、NotesDocument 的扩展类、NotesDocument 类中的 AppendItemValue 和 ReplaceItemValue 成员函数。

```
Set notesItem = New NotesItem( notesDocument, itemName$, value [, specialType% ] )
Set notesItem = notesDocument.AppendItemValue( itemName$, value )
Set notesItem = notesDocument.ReplaceItemValue( itemName$, value ) 或
notesDocument.itemName = value
```

参数：

- notesDocument：Notes 文档对象，在该文档中创建新条目。
- itemName：字符串，新条目的名称。也可以用作 NotesDocument 对象的扩展类。
- value：设置条目的值。

说明

在本例中，向文档添加新的临时条目是对后台文档（NotesDocument 对象）的操作。除非临时条目被存储并使用 NotesUIDocument 对象的 RefreshHideFormulas 成员函数刷新当前文档，该条目值不会影响到已打开的文档隐藏公式。因此，即使该文档正在被另一个应用打开，已打开文档的隐藏状态也不会改变。为了防止临时条目被保存在磁盘上，可以将这个临时条目的 SaveToDisk 属性设置为 False。

注意

弹出对话框时，Dialogbox 成员函数中的 notesDocument 参数不能省略，一定要将其指定为含有临时条目的文档对象。这样，对话框中显示的内容才是由含有临时条目的文档对象生成的，临时条目的值才会对隐藏公式起作用。



跟我来

下面的这段代码在文档中创建一个名为“IsDialogBox”的临时域，并使用这个文档对象打开对话框：

```
Dim ws As New NotesUIWorkspace
Dim uidoc As NotesUIDocument
Dim doc As NotesDocument
Set uidoc = ws.CurrentDocument
Set doc = uidoc.Document

Set item = New notesitem(doc, "IsDialogBox", True)
```

```
item.SaveToDisk = False
```

```
Call ws.dialogbox("FormName", False, False, True, True, True, True, "Title", doc)
```



通过程序改变表单背景色



遇到难题

Notes 提供了 240 种表单背景色供设计者创建一个色彩缤纷的数据库。如果指定了“允许用户设置表单属性”选项，用户也可以依照自己的喜好随意改变文档的背景色。怎样用程序控制表单的背景颜色呢？



钥匙在此

文档的背景颜色由名为“\$PaperColor”的域确定。Notes 用“\$”符号标识由系统创建的条目。按照下面的步骤可以看到“\$PaperColor”域值的变化：

在表单设计中选中“允许用户设置表单属性”选项，打开一个使用该表单的文档并进入编辑模式，执行菜单命令“文件-文档属性”，在“背景”签中更改文档的背景颜色。保存文档后观察“域”附签中的域列表，就会发现文档中多了一个名为“\$PaperColor”的域（如图 1-2 所示）。

\$PaperColor 是数字型域，取值范围为从 0 到 239 的整数。图 1-3 所示的是表单背景颜色选择对话框，其中小色块排列顺序表示域值和颜色之间的对应关系：0 代表黑色，1 为白色，以此类推。通过程序更改 \$PaperColor 域值，就可以实现动态变换表单背景色的效果。

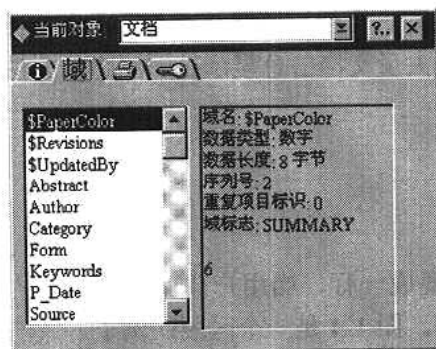


图 1-2 文档域列表

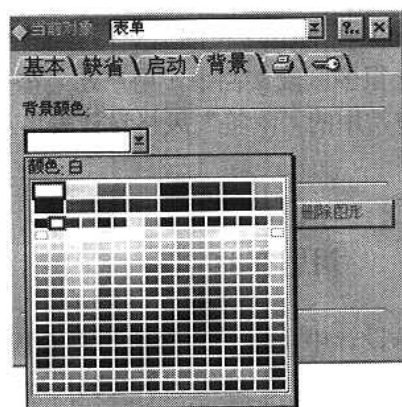


图 1-3 颜色选择框



跟我来

下面一段代码放在视图的 Queryopendocument 事件中，每次以不同的背景色打开文档。

```
Sub Queryopendocument(Source As NotesUIView, Continue As Variant)
    Dim ws As New NotesUIWorkspace
    Dim docs As NotesDocumentCollection
    Dim doc As NotesDocument

    Randomize
    Set docs=Source.Documents
    Set doc=docs.GetFirstDocument()
    Call doc.ReplaceItemValue("$PaperColor", Int(Rnd(1)*240))
    Call ws.EditDocument(False,doc)
End Sub
```

说明

NotesUIView 的 Documents 属性的值是一个 NotesDocumentCollection 对象。在 QueryOpenDocument 事件中，Documents 属性包含了用户想要打开的文档。



使用“附签”风格代替过多的区段



遇到难题

区段使文档的阅读和浏览变得容易。使用可折叠的区段可以更好地组织文档的信息。但是在包含很多区段文档中，用户还是需要不断翻页才能一览文档全貌。本节介绍一种使用 Notes UI 中常用的“附签”风格设计表单从而将大量文档信息显示在一屏中的方法。



钥匙在此

在表单设计中可以使用区段将多个段落折叠成一行，当用户想阅读其中的内容时再将它展开。这是一种有效的组织长文档内容的方法。图 1-4 是一个区段的例子。

要在表单中创建区段，选择菜单命令“创建-区段”。在 RTF 文本域中也可以创建区段。

创建了区段以后，通过区段属性框来格式化这个区段。区段属性框中可执行的操作包括：

➤ 改变区段标题：缺省情况下，如果在创建区段时选中了一部分文本的话，文本的第一段将成为区段的标题。要改变区段标题，应编辑区段属性框的“标题”附签。除了指定静态文本为区段标题以外，还可通过公式计算区段标题。在“标题”附签中还可以指定边框样式和颜色选项。

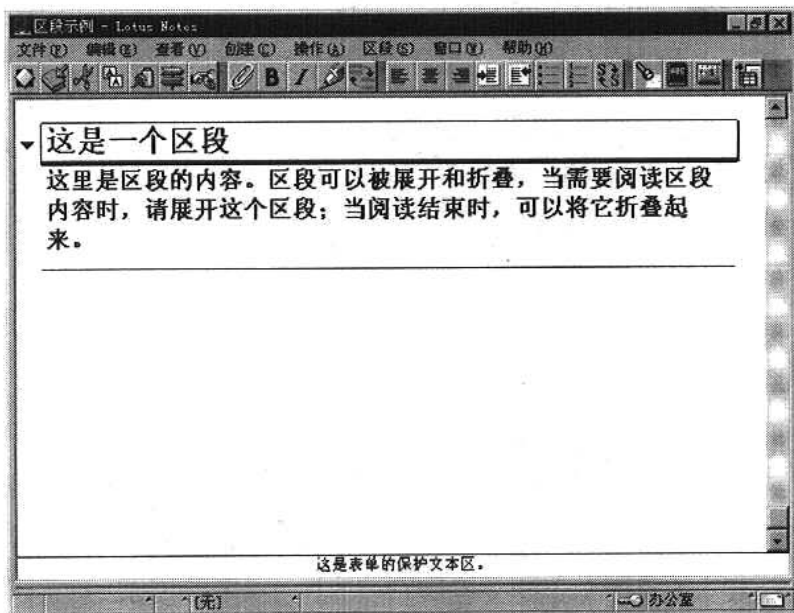


图 1-4 区段示例

➤ 控制区段的展开与折叠：在区段的“展开/折叠”附签中可以指定区段自动折叠或展开的条件，包括预览、阅读、编辑或打印等状态。

➤ 控制区段的隐藏：要隐藏区段标题，选择“展开/折叠”附签中的“展开时隐藏标题”选项。要隐藏区段内容，请在“隐藏”附签中指定。

区段虽然有效，但当文档很长时，即使使用区段也无法将文档内容显示在一页中。当用户翻页翻到文档底端时，可能已经忘记前面区段的内容。在表单设计中使用“附签”风格就可以避免这一弊端。

“附签”风格是通过表单上端的一行或多行热点按钮实现的，每一个按钮对应一个区域布局。在按下了一个热点按钮时，只显示和它相关的布局，其他区域布局都处于隐藏状态。这样的效果可以通过布局隐藏公式实现。

下一段内容包括了具体的实现范例和运行效果。



跟我来

图 1-5 所示的是一个应用示例的运行效果，这个“用户信息”表单由若干子表组成，分别对应第一行的“用户信息”等按钮。下一行按钮对应下一级子表，如“用户信息”又分为“基本信息”、“附加信息”等若干部分。当按下其中一个按钮时，屏幕上将显示相应的子表。设计这样的一个表单需要执行下面的步骤：

1. 为每一个子表创建一个相应的控制按钮。
2. 为每一个子表创建一个区域布局，布局中包含子表内容（本例的表单设计如图 1-6 所示）。
3. 为控制按钮指定下面的代码：

图 1-5 “附签”风格应用示例

图 1-6 “附签”风格的表单设计

Sub Click(Source As Button)

Dim session As New NotesSession

Dim workspace As New NotesUIWorkspace

```
Dim uidoc As NotesUIDocument
```

```
Set uidoc = workspace.CurrentDocument
```

```
Call session.SetEnvironmentVar( "LeadMainSection", "Preferences" )
```

```
Call session.SetEnvironmentVar( "LeadSubSection", "Residence" )
```

```
Call uidoc.RefreshHideFormulas
```

```
End Sub
```

上面的代码将被激活的子表名称写入环境变量。条目 LeadMainSection 代表第一级字表，即选中了第一行中的哪个按钮；条目 LeadSubSection 代表下一级字表，对应第二行中的按钮。

在布局属性框中为每个区域布局指定如下的隐藏公式：

```
@Environment("LeadSubSection")!="Residence"
```

4. 为每个按钮和布局指定相应的代码和布局隐藏公式，并将它们一一联系起来。



再送礼包

下面提供一些使用区段的技巧。

利用区段隐藏 RTF 文本域的全文

在 RTF 文本域中，只有文字是可以隐藏的，表格和附件无法隐藏。但是可以利用区段隐藏整个 RTF 文本域，只让被授权的用户看到 RTF 域中的内容。

1. 为 RTF 文本域创建一个控制存取区段。
2. 在区段属性框中指定如下选项：
 - 在“标题”附签中，将边框风格和颜色与表单颜色统一。
 - 在“编辑器”附签中，为每一栏指定“自动展开区段”选项。
 - 在“非编辑器”附签中，为每一栏指定“自动折叠区段”选项。
 - 指定“展开时隐藏区段标题”选项。
 - 在“公式”附签中，为被授权用户编写公式。
 - 在“隐藏”附签中，选择所有隐藏选项。



使用不同的表单查看文档



遇到难题

同一个文档，在阅读、编辑和打印时却需要分别使用不同的表单打开，应该怎样做呢？



钥匙在此

使用公式和 Script 都可以做到这一点。公式语言中的 ViewSwitchForm 命令改变显示当前文档的表单。语法为：

```
@Command( [ViewSwitchForm] ; formName )
```

参数 formName 为文本，是要切换的表单名称。缺省时显示一个对话框，列出当前数据库中所有可用的表单。

无论在 @Command 还是 @PostedCommand 中，ViewSwitchForm 命令都必须在整个公式计算完成后才执行。因此有时需要通过 Script 实现这一功能。

一般情况下（不使用“在文档中存储表单”时），文档是和表单分开存储的，在显示时才被组合到一起。显示文档的表单名称存储在“Form”域中。因此，通过 Script 改变显示文档用的表单，只需改变文档的“Form”域即可。下面的两行 Script 使用名为“Form1”的表单显示 doc 文档：

```
doc.Form = "Form1"
Set uidoc = ws.EditDocument(True,doc)
```

说明

NotesUIWorkspace 的 EditDocument 成员函数不仅仅能打开当前选定文档，还可以打开后台文档（即 NotesDocument 对象）。

```
Set notesUIDocument = notesUIWorkspace.EditDocument( [ editMode [, notesDocument [, notesDocumentReadOnly]] ] )
```

如果指定了第二个参数 notesDocument，函数将打开指定的文档，而不是当前选定的文档。



跟我来

下面的程序是一个通用的更换表单 Script，由 InitSwitch 和 SwitchForm 两个子过程组成。

```
Sub InitSwitch (FormName As String)
Dim session As New NotesSession
Dim ws As New NotesUIWorkspace
Dim uidoc As NotesUIDocument

Set uidoc = ws.CurrentDocument
```

'将表单更换的参数写入环境变量。

'PFormSwitch 是要更换的表单名称

'FormSwitch 为 1 时，通知后面的程序需要进行表单更换，否则为 0

Call session.SetEnvironmentVar("PFormSwitch",FormName)

Call session.SetEnvironmentVar("FormSwitch","1")

'如果需要保存文档，在此调用 uidoc.Save

Call uidoc.Close

End Sub

Sub SwitchForm

Dim session As New notessession

Dim db As NotesDatabase

Dim ws As New NotesUIWorkspace

Dim uidoc As NotesUIDocument

Dim doc As NotesDocument

'从环境变量中取出表单并更换参数

Switch\$ = session.Getenvironmentstring("FormSwitch")

If Switch\$ = "1" Then

Set uidoc = ws.CurrentDocument

Set doc = uidoc.Document

doc.Form = session.Getenvironmentstring("PFormSwitch")

Set uidoc = ws.EditDocument(True,doc)

'重置 FormSwitch 参数

Call session.Setenvironmentvar("FormSwitch","0")

End If

End Sub

说明

1. 使用方法:

在表单事件中的“Options”处引入上面两个外部过程。更换表单时，先调用 Call InitSwitch(formName)，然后在旧表单的 Terminate 事件中调用 Call SwitchForm。InitSwitch 子过程中的 Call uidoc.Close 语句会触发表单 Terminate 事件中的 SwitchForm 过程。

2. 如果希望文档以固定的表单存储，可以在编辑此文档的表单中增加一个名为“Form”的隐藏计算域，在该域的计算公式中输入相应的表单名。这样，无论这个文档是通过哪个表单打开的，都不会影响存储时使用的表单。



更新存储在文档中的表单



遇到难题

一些特殊情况下，在文档中存储表单可以保证文档的正确显示。但这样做不但占用了更多空间，而且如何更新存储在文档中的表单也是一道难题。



钥匙在此

在文档中存储表单的特性通常用于以下几种情况：

- 数据库中的文档将被邮递或粘贴到不包含原始表单拷贝的数据库中。
- 数据库中的文档将被邮递或粘贴，而原始表单没有共享同名。
- 包含嵌入式 OLE 对象或子程序的表单，希望使文档反映出对象的任意改变。
- 选择了“包含在搜索方式构造器”选项，并且希望使表单中的静态文本可以被搜索到。
- 使用该表单创建的文档保存在压缩数据库中并邮递给 cc:Mail 用户。

要指定在每个文档中存储表单，执行下面的步骤：

- 在设计模式下打开表单。
- 单击属性快捷图标打开表单属性框。
- 在“缺省”附签中选择“在文档中存储表单”（如图 1-7 所示）。

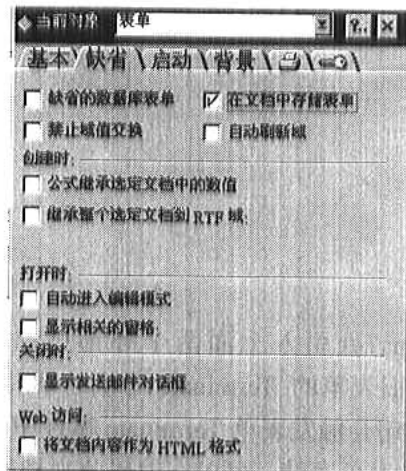


图 1-7 指定在文档中存储表单

- 在属性框中的“当前对象”中选择“数据库”。

➤ 在“基本”附签中选择“允许在此数据库中使用预先存储的表单”（如图 1-8 所示）。

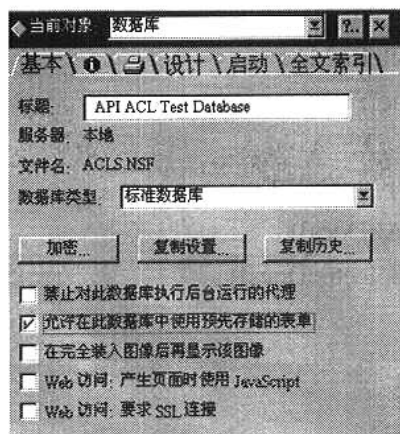


图 1-8 激活在数据库中使用预先存储的表单

为表单指定了这个属性以后，所有由这个表单创建的文档中都将存储这个表单中的内容。通过程序邮寄文档时，如果将 NotesDocument 的 Send 成员函数中 attachForm 参数设置为 True 的话，表单同样会随文档保存，不需要特别指定表单属性，并且该表单也不会保存在其他文档中。这一特性对正确显示被邮寄到其他数据库中的文档很有帮助。

说明

NotesDocument 的 Send 成员函数的语法：

Call notesDocument.Send(attachForm [, recipients])

在存储表单的文档中，表单名被存储在名为 \$Title 的域中，其他信息分别存储在名为 \$Info、\$WindowTitle 和 \$Body 等域中。更新原表单后要正确显示所有存储这个表单的文档，请执行以下操作：

- 创建一个代理，在代理中使用 SELECT 关键字选取 \$Title 条目值为该表单名的文档。
- 从文档中删除所有存储表单信息的条目。要使用公式删除一个条目，用 FIELD 关键字将这个条目赋值为 @DeleteField。
- 将文档的 Form 域赋值为该表单的名称。
- 调用 @Command([ToolsRefreshAllDocs]) 或 @Command([ToolsRefreshSelectedDocs]) 刷新文档。



跟我来

删除和文档一起存储的表单信息：

```
SELECT $TITLE="Form Name";
FIELD $TITLE:=@DeleteField;
FIELD $INFO:=@DeleteField;
```