

---

# 第 13 章 复合应用和扩展应用开发 (节选)

---

复合应用开发是 Lotus Domino Designer 8 中最新推出的开发新功能之一，也是版本 8 发布中开发部分最亮丽的一道风景。除此以外，基于 Eclipse 的开放结构下的 Lotus Notes 8 客户端还为所有开发人员提供了强大的二次开发接口。本章将围绕这两个最新开发设计理念做详细介绍。

## 13.1 基于 Notes 的复合应用开发

复合应用程序将一个画面分割成不同的区域或窗口，每个区域对应于一个独立的组件。但是一个应用程序仅仅具有分割成不同区域的画面并不能称之为复合应用程序。只有这些不同区域之间互相发生作用，比如在一个组件中的操作触发另一个组件执行相应的操作，才能称之为复合应用程序。

如图 13-1 所示是一个复合应用程序示例。其中包括三个组件：屏幕顶端的 Notes 讨论数据库中“所有文档”视图，屏幕左下方为同一个 Notes 讨论数据库中“类别文档表单”表单，屏幕右下方为 Eclipse 视图“TopicView”。当选中组件“所有文档”视图中的一个文档时，左下方组件会显示此文档所属类别的所有文档，右下方组件会显示此文档所属类别信息。

复合应用程序是指为了满足特定用户群的商业需要，可以从一个屏幕来访问的两个或多个独立应用程序组成的集合。复合应用程序由不同的组件组成，允许用户执行有关的任务而无须启动新的窗口或应用程序。

复合应用程序的支持及其应用概念的引入是整个 Lotus Notes 8 中主要的新特性之一。复合应用程序用于创建企业聚合 (Enterprise Mashups)，很方便地将行业方案及数据集成到新的类型的应用程序中。

复合应用程序是 SOA (面向服务架构, service-oriented architecture) 及上下文协作策略的一个重要组成部分，其可以存取访问多个数据源的信息，例如：Lotus Notes 数据库、Java 应用程序、Web 应用程序或者 CRM (客户关系管理 Customer Relationship Management) 应用程序。

应用程序各组件相互可以发送信息，这样当一个应用程序组件中信息发生变化时，其他应用中相对应的视图或者信息也会随之改变。

通过复合应用程序，可以设计可重用的组件，然后混合及搭配它们，用极少的甚至于无须任何额外的代码即可以创建广泛的应用程序。无论是联机还是脱机方式下，复合应用程序都能够很容易地进行自助操作。使用 Lotus Notes 8 软件的复合应用程序编辑器功能，用户及行业管理者可以很容易地将开发的组件混合并搭配到其定制的应用程序中去。

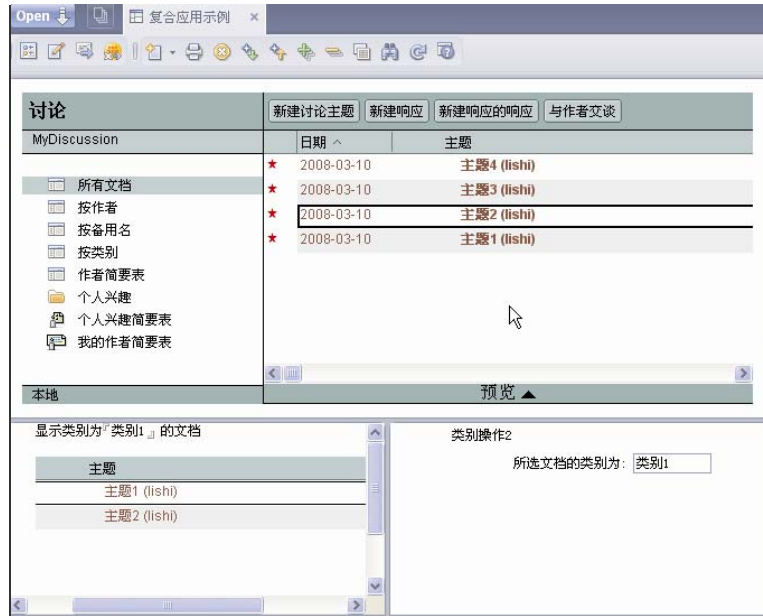


图 13-1 复合应用示例

复合应用程序能帮助利用现有的技术来促进投资获益，如 IBM WebSphere Portal 及 Lotus Domino。可以重用先前开发的基于 Eclipse 组件到基于 Domino 8 软件的复合应用程序中，帮助增加应用程序开发工具及技能上的投资获益。复合应用程序既可以位于 Lotus Domino 8 服务器环境中，也可以位于 WebSphere Portal 6 服务器环境中。

了解了复合应用程序的概念及其好处之后，接下来将介绍 Lotus Notes 8 中最主要的新特性之一——复合应用程序，并将针对如何在 Lotus Domino Designer 的环境下创建复合应用程序做详细介绍，其中包括创建流程、组件设计、组件构建、装配及连接、部署及测试等。最后将介绍如何创建示例应用程序。

在开始创建复合应用程序之前，假定已经知道如何创建新 Notes 数据库、修改 Notes 视图、创建简单的 Notes 表单、在编程窗格中输入 Notes 公式及 LotusScript。

### 13.1.1 Lotus Notes 8 中的复合应用程序

复合应用程序是 SOA 及上下文协作策略的重要组成部分，复合应用程序的创建及编辑可以不费力地集成不同类型的组件和技术。通过创建复合应用程序，可以松散地耦合 NSF 组件、Eclipse 组件、Lotus Component Designer 组件，成为用户体验的一部分。各组件可以连接在一起，

这样用户在一个组件中的操作就能够触发另一个组件中的操作。

Lotus Notes 8 软件同时结合了 Eclipse 应用程序开发架构的开放式标准，以及基于组件的 SOA 架构。这就为方便地联合、访问、部署多个软件程序的功能提供了基础。利用“属性”、“操作”，以及“属性代理程序”（属性代理程序）的支持，复合应用程序中的组件连接在一起，用户在一个组件中执行的操作能触发另一个组件中已安排好的操作。

IBM WebSphere Portal 复合应用程序模型现在也适用于 Lotus Notes 8 客户端，Lotus Domino Designer 将“属性代理程序”扩展到 NSF 组件，从而使 Notes 客户端能够提供更有价值的用户体验。

我们既可以使用 Domino 作为复合应用程序的服务器，也可以使用 WebSphere Portal 作为复合应用程序的服务器，还可以脱机使用基于 NSF 的复合应用程序。位于 WebSphere Portal 的复合应用程序可以在复合应用程序编辑器（Composite Application Editor, CAE）或 Portal 应用程序模板编辑器中进行定义，位于 Domino 的复合应用程序可以在 CAE 中进行定义，但需要特定权限才能在 WebSphere Portal 服务器上创建应用程序。对于在 Domino 上的复合应用程序定义，必须要在 Domino 服务器上新建数据库，并且要有“管理者”权限来更改数据库访问控制列表。

除了满足 Lotus Notes 8 的系统前提条件以外，还需要安装下列软件来开发和组装复合应用程序：

- 带 CAE 功能的 IBM Lotus Notes 8 客户端；
- IBM Lotus Domino Designer 8 客户端。

**!** 特别提示：Lotus Domino Designer 当前不支持 Linux 平台。

## 1. 属性代理程序

Notes 应用程序需要 WSDL 设计部分才能利用属性代理程序。在 IBM Lotus Domino Designer 中，使用所提供的属性代理程序编辑器功能，很容易创建 WSDL 部分。

属性代理程序（与 WSDL 有关的）主要内容如下。

1) 属性：属性是组件可产生的带类型的可交换数据条目。一个复合应用中所有组件都使用一个属性代理程序来传递消息。消费一个属性意味着接收另一个组件发布的信息，并相应地做一些处理。属性可以在任何时候由 LotusScript 代码发布，或者通过关联某一视图列与一个可发布的属性，当选中视图中一行时自动发布，随着不同行的选择，所关联列的数据就发布出来。

2) 操作：此外的操作是消费属性的逻辑，仅 LotusScript 操作能消费属性。

**!** 特别提示：此处的“操作”一般用于讨论消费属性的组件中的“plug”（而不是服务或用户界面服务）。请勿混淆此处的操作和 Notes 视图及表单中用到的操作按钮。所有组件都可以有操作，不管是否是 Notes 应用。尽管不同，但组件触发的 Notes 操作与严格意义上的“Notes

操作”还是有联系的。使用 Domino Designer ,可以关联一个 Notes 操作和属性代理程序调用的操作 ,这样当接收到信息时 ,Notes 客户端就知道具体是哪一个 Notes 操作 ( 严格意义的 ) 要执行。

3) 组件: 每一个组件定义自己的属性及操作, 可以与其他组件的不同。

4) 一个外部的连接: 将一个组件中的属性与其他组件中的操作联系起来, 这样属性的发布就能触发操作。这有助于在不同应用程序中重用组件。

**!** 特别提示: 连接必须在复合应用程序级别而非组件级别进行定义, 这样确保了组件的松散耦合。

5) WSDL 文件: 用 WSDL 文件定义属性及操作。对于 Notes 组件, WSDL 文件已导入到 Domino Designer 8 新的设计元素“组合应用程序”——“连线属性”中。对于 Eclipse 组件, WSDL 文件包括在 plugin 中, 通过 plugin.xml 文件中的扩展点来引用。这两种情况下, 所提供的属性代理程序编辑器功能可以用于创建 WSDL 文件。如果已熟悉 WSDL 语法, 也可以不用该工具, 手工创建文件。

## 2. 复合应用程序支持的元素

复合应用程序是为了满足商业目的, 由多个组件组成的集合。这些组件可以是不同类型, 由不同工具开发。复合应用程序及组件既可以运行在富客户端 (Lotus Notes 或 Lotus Expeditor), 也可以通过 WebSphere Portal 在浏览器上展现。复合应用程序能包含下列组件类型:

- NSF 组件;
- Eclipse 组件;
- 本地 portlet 组件;
- Portlet 组件;
- Lotus Component Designer 组件;
- WebSphere Portlet Factory 组件。

不同客户端支持的组件及应用程序类型如表 13-1 和表 13-2 所示。

表 13-1 富客户端支持的组件及应用程序类型

组件类型/后台服务器情况	基于 NSF 应用程序	Portal 应用程序
NSF 组件	支持	支持
Eclipse 组件	支持	支持
本地 portlet 组件	支持	支持
Lotus Component Designer 组件	不支持	不支持

WebSphere Portlet Factory 组件	不支持	不支持
------------------------------	-----	-----

表 13-2 浏览器客户端支持的组件及应用程序类型

组件类型/后台服务器情况	基于 NSF 应用程序	Portal 应用程序
NSF 组件	不支持	通过 DAP 支持 <sup>①</sup>
Eclipse 组件	不支持	通过定制 portlets 支持 <sup>②</sup>
Portlet 组件	不支持	支持
Lotus Component Designer 组件	不支持	支持
WebSphere Portlet Factory 组件	不支持	支持

以上介绍了 Lotus Notes 8 中复合应用程序的实现及支持的组件类型。接下来将详细介绍在 Lotus Notes 8 中如何创建基于 NSF 的复合应用程序。

### 13.1.2 复合应用程序创建流程

首先，创建用于 Lotus Notes 8 的复合应用程序可分为 4 个步骤：组件设计、组件构建、装配及连接复合应用程序、部署与测试复合应用程序。

#### 1. 组件设计

在 Domino Designer 中创建复合应用程序的第一步是组件设计，Domino Designer 中支持两种类型组件：NSF 组件与 Eclipse 组件。

关于如何设计 Lotus Notes 组件及 Eclipse 组件，下面介绍一些相关的指导。

##### (1) NSF 组件：

- 表单——可用做组件。如果要创建一个表单用做组件，请注意该表单可能会位于屏幕上不同的区域，与多个其他组件联合。
- 域——如果要创建在组件中使用的域，请注意它或许与多个其他组件联合。
- 页面——可用做组件。如果要创建一个页面用做组件，请注意该页面可能会位于屏幕上不同的区域，与多个其他的组件联合。

<sup>①</sup> 可以用 DAP (Domino Application Portlet) 在复合应用程序中嵌入 NSF 组件。这样的话，就可以在 portlet 中呈现 Domino server 上原始的 HTML。DAP 不支持属性代理程序，这意味着能将 NSF 组件和其他组件放在一个页面上，但组件间无法交互。将 NSF 数据放到复合应用程序中的另一个方法是使用自己的 portlet，一个 Lotus Component Designer 组件或 WebSphere Portlet Factory 组件。这样将可以支持复合应用程序，但是得重新实现用户界面及 NSF 应用的商业逻辑。

<sup>②</sup> 需要通过 portlet，才能在基于 Web 的复合应用程序中使用 Eclipse 组件。

讨论复合应用程序时注意区分以下术语：

- Notes 应用——指以前所谓的“Notes 数据库”，包括视图、框架等；
- NSF 组件——用于复合应用程序的 NSF 数据库；
- NSF 组件的内容——例如，Notes 视图、Notes 表单等；
- 基于 NSF 的应用——NSF 文件，其中存储了一到多个复合应用程序的 XML 文件。

- 文件夹、视图——可用做组件。
- 导航器——可用做组件；
- 框架——可用做组件；
- 属性——提前确定组件将发布及共享的属性；
- 操作——提前确定当其他组件属性变化时关联的组件要执行的操作。

## ( 2 ) Eclipse 组件 :

- 属性——提前确定组件将发布及共享的属性。
- 操作——提前确定当其他组件属性变化时，关联的组件要执行的操作。

## 2. 组件构建

基于 NSF 的复合应用程序中 NSF 组件和 Eclipse 组件的构建是如何实现的呢？

如果需要构建 NSF 组件及应用程序(Notes 数据库),使用 Lotus Domino Designer 应用软件。如果需要构建 Eclipse 组件及应用,使用 Eclipse 集成开发环境 (IDE) 或者 Rational<sup>®</sup> Application Developer (RAD)。

### ( 1 ) NSF 组件构建

一个 Notes 组件可以理解成一个链接,链接到 Notes 数据库中包含的特定的视图、框架、文档,或者是其他设计组成部分。同一个 Notes 数据库可以包含多个组件,这些组件可以用于多个复合应用程序。

在同一个 Notes 数据库中的所有组件共享同一套类型、属性以及操作(在一个或多个“组合应用程序”——“连线属性”,以及“应用程序”设计元素中进行定义)。组件不需要发布所有的输出属性或者实现所有操作。组件要参与到复合应用程序,必须引用复合应用程序属性 WSDL 文件中的属性及操作。

一个组件可以仅发布属性,可以仅消费属性,也可以既发布属性,又消费属性。

组件发布属性有两种方式:

- 使用 LotusScript API。可以在任何 LotusScript 实现的地方使用,包括在代理中使用;
- 如果组件是视图或文件夹,可以将某一系列与 WSDL 文件中定义的输出实现相关联。通过列属性的高级签中的选项来实现。因为一系列仅包含单一的值,所有仅能与“简单(simple)”类型的属性进行关联。

Notes 组件要消费另一个组件提供的属性只有一个办法。Notes 组件必须包括一个“WSDL 操作”相对应的“Notes 操作”。要在 Notes 操作和 WSDL 操作间创建关联,使用 Action 属性框的高级签。

当一个属性更改事件发生时,如果属性已经与 Notes 组件中的一个 WSDL 操作相关联,并且如果 Notes 组件包含一个 Notes 操作与 WSDL 操作关联,那么 Notes 操作就会执行。通常,Notes 操作会使用下面提到的 LotusScript APIs 来发现所发布的值,并用该值做些操作(例如,用该值重新配置当前视图以显示匹配文档)。

只有当包含 Notes 操作的表单、视图或者页面由组件显示的时候,此 Notes 操作会随着 WSDL 操作的触发而被执行。可以将一个 Notes “共享操作”与 WSDL 操作进行关联,但是这并不能

使它在整个应用中一直执行，仍然要将此共享操作放到它将要被执行的表单或视图中。

属性更改事件调用的所有 Notes 操作，一般都应当是一直被隐藏的。这些操作一般使用 LotusScript API 来从属性代理程序读取输入属性。并且如果是由用户激活/触发的话，会因为没有用于读取的输入属性而引发错误。

可以按照如下所述的步骤构建用于复合应用程序的 NSF 组件（数据库、视图、表单等）：

- ① 在 Domino Designer 8 中完成标准的 Notes 数据库设计；
- ② 确定组件将要发布的属性；
- ③ 确定与其他组件属性关联的组件要执行的操作；
- ④ 在 WSDL 文件中定义属性及操作，通过使用属性代理程序编辑器特性来实现；
- ⑤ 在 Domino Designer 8 中将该 WSDL 文件导入到 Notes 数据库中；
- ⑥ 修改 NSF 设计元素以与先前定义的属性和操作关联起来。通过使用 Domino Designer 8

新特性实现，包括新的 LotusScript 类、方法及属性。比如，为了实现新的复合应用程序，可能会用到下列 LotusScript APIs 来有计划地发布属性：

- NotesSession.GetPropertyBroker;
- NotesPropertyBroker.SetPropertyValue;
- NotesPropertyBroker.Publish。

Notes 数据库中的 LotusScript 代码能发布属性到属性代理程序。Notes 视图或表单中的 Notes 操作中 LotusScript 代码能被属性代理程序触发。以下是两个 API 例子。

#### 例 1：发布属性的新 LotusScript API

```
Sub Click(Source As Button)
  Dim s As New NotesSession
  ...
  Dim pb As NotesPropertyBroker
  Set pb = s.GetPropertyBroker()

  Call pb.setPropertyValue("Track", newTrack$)
  Call pb.Publish()
  ...
End Sub
```

#### 例 2：获取属性的新 LotusScript API

```
Sub Click(Source As Button)
  Dim s As New notessession
  ...
  Dim pb As NotesPropertyBroker
  Set pb = s.GetPropertyBroker()
  Dim pbInputProperty As NotesProperty
  Dim pbContext As Variant
  pbContext = pb.InputPropertyContext
  Set pbInputProperty = pbContext(0)
  Dim InputPropertyName As String
  Dim NameSpace As String
  InputPropertyName = pbInputProperty.Name
  NameSpace = pbInputProperty.NameSpace
```

```

Dim pbValue As Variant
pbValue = pb.getPropertyValue(InputPropertyName,NameSpace)
cName$ = pbValue(0)
...
End Sub

```

Lotus Domino Designer 8 中，添加了如下所示的新的复合应用程序有关的 LotusScript APIs。

#### ① NotesSession

- NotesSession.GetPropertyBroker()——返回 NotesPropertyBroker。

#### ② NotesPropertyBroker

- NotesPropertyBroker.InputPropertyContext (属性) ——返回一个包括输入 Notes 属性的数组。对于版本 8，仅数组的第一个成员被装配上。
- NotesPropertyBroker.HasProperty(Name) ——布尔值。指出新属性是否存在。
- NotesPropertyBroker.GetProperty(Name) ——返回 Notes 属性。
- NotesPropertyBroker.GetProperty Value(Name) ——返回指定属性的值。
- NotesPropertyBroker.SetPropertyValue(Name, Value) ——返回 Notes 属性。它创建一个后台对象并赋值。如果属性名称未在 WSDL 中定义的话，此方法会抛出一个错误。
- NotesPropertyBroker.ClearProperty ——新方法用于删除任一新的或修改过的属性，不用于输入属性。
- NotesPropertyBroker.Publish ——新方法用于发布所有新的及修改过的属性。

#### ③ NotesProperty

- NotesProperty.Values (属性) ——简单的属性读/写，返回类型为字符串的可变数组。单一值属性包含的数据存在数组第一个元素中。作为输入，接受类型为字符串的变量或数组。数组是其次的，这里的数组仅能包含一个字符串条目，无法设置输入属性。
- NotesProperty.Name (属性) ——只读，返回字符串方式的属性名称。
- NotesProperty.TypeName (属性) ——只读，返回字符串方式的类型名称。
- NotesProperty.NameSpace (属性) ——只读，返回字符串方式的 namespace。
- NotesProperty.Title (属性) ——只读，返回字符串方式的标题。
- NotesProperty.isInput (属性) ——只读，布尔值。如果是输入属性，则返回真值。否则返回假值。
- NotesProperty.Description (属性) ——只读，返回字符串方式的描述。
- NotesProperty.Publish() ——新方法用于发布修改过的属性。
- NotesProperty.Clear() ——新方法用于删除新的或修改过的属性。

## ( 2 ) Eclipse 组件构建

接下来将对如何构建 Eclipse 组件做简单介绍。更加详细的信息，请参考 IBM Lotus Expeditor Toolkit 提供的文档。

可以使用 Eclipse 集成开发环境 (IDE)、Lotus Expeditor Toolkit，或者 Rational Application Developer 来构建 Eclipse 组件。Notes 8 客户端能访问的复合应用程序可以包括 NSF 组件及 Eclipse 组件的组合，这些组件通过属性代理程序所支持的组件内交流来交互。

首先必须创建一个 Java 插件组件，然后在一个 Eclipse feature project 中关联之前创建的插



件。

这步做完后，创建或更新一个更新站点项目是很重要的。这是一个 Eclipse 项目，其输出可以放置在 HTTP 服务器上。在 IBM Lotus Notes 8 环境中，features 及插件 s 的更新及供应都离不开更新站点。

可以按照下列步骤来开发用于复合应用程序的 Eclipse 组件：

- ① 使用 Eclipse IDE；
- ② 创建一个 Java 插件组件；
- ③ 确定组件要发布的属性；
- ④ 确定当其他组件属性变化时，关联的组件要执行的操作；
- ⑤ 创建 WSDL 文件，在其中定义属性及操作；
- ⑥ 定义插件中的扩展点（通过 plugin.xml 文件），用于引用上述 WSDL。如下面加黑文字所示：

```
<plugin>
  <extension
    id="loadURLAction"
    name="Load A URL into Our Window"
    point="com.ibm.rcp.propertybroker.PropertyBrokerAction">
    <action
      active="true"
      class=
        "com.ibm.pvc.samples.propertybroker.browser.LoadURLAction"
      file="loadurlaction.wsdl" />
    </extension>
  ...
</plugin>
```

- ⑦ 使用属性及操作方面的新属性代理程序 Java APIs；
- ⑧ 编写操作以实现 plugin.xml 中定义好的接口，实现操作的类必须事先在 plugin.xml 中扩展点进行定义；
- ⑨ 在一个 Eclipse 功能（feature）项目中关联之前创建的插件；
- ⑩ 通过创建或更新一个更新站点项目来部署组件。这是一个 Eclipse 项目，其输出既可以位于 HTTP 服务器也可以在本地的文件夹；

**!** 特别提示：一个本地更新站点可以用于组件开发及测试。在应用产品的阶段，

推荐按照下列指导使用远程更新站点。

插件必须使用更新站点项目来打包。

更新站点的位置：

更新站点既可以位于 HTTP 服务器（例如 <http://test.ibm.com/casample/site.xml>），也可以

在本地访问（例如 file:/C:/test/casample/site.xml）；

在本地访问的情况下，更新站点文件夹必须与用户机器上的位置相符合。这种情况也常用于开发人员本地调试。

⑪ URL 必须在 CA.xml 中属性 url.feature 中进行设置。在复合应用程序编辑器中从 Eclipse 更新站点添加组件时，这一步会自动完成；

⑫ 给定 URL 的组件供应会在第一次在 Lotus Notes 8 客户端加载复合应用程序时进行。所需的插件自动被下载到 shared/eclipse 文件夹。

构建 Eclipse 组件时，必须使用 Lotus Notes 8 提供的新的属性代理程序 APIs。例如：

```
import org.eclipse.core.commands.AbstractHandler;
import org.eclipse.core.commands.ExecutionEvent;
import com.ibm.rcp.propertybroker.event.*;
import com.ibm.rcp.propertybroker.property.*;
public class YourAction extends AbstractHandler {
    public Object execute(ExecutionEvent arg0) throws ExecutionException {
        Object event = arg0.getTrigger();
        if (event instanceof PropertyChangeEvent) {
            PropertyChangeEvent e = (PropertyChangeEvent)event;
            PropertyValue propValue = e.getPropertyValue();
            if(propValue.getProperty().getType().equalsIgnoreCase("employeeName")) {
                String employee = (String)propValue.getValue();
                // do something
            }
        }
        return null;
    }
}
```

下面是另一个使用新属性代理程序 APIs 的例子。

```
import com.ibm.rcp.propertybroker.service.*;
...
String selectedEntry = ...;
PropertyBroker broker = PropertyBrokerFactory.getBroker();
PropertyValue[] values = new PropertyValue[1];
SimpleProperty sProp = new SimpleProperty(selectedEntry);
Property property = broker.getProperty( "http://www.ibm.com/notes/pb/sample",
"Subject");
values[0] = PropertyFactory.createPropertyValue(property, sProp);
broker.changedProperties(values);
```

### 3. 装配和连接

可以使用 Lotus Notes 8 中的复合应用程序编辑器来装配及连接基于 NSF 的复合应用程序。

组件构建好之后，需要新建一个基于 NSF 的复合应用程序用以装配及连接组件。这步做完

之后，再在 Lotus Notes 8 中，单击菜单文件>打开>Lotus Notes 应用先前新建的 NSF 复合应用程序。接下来，单击操作>编辑应用程序进入复合应用程序编辑器，从而开始添加组件。

可以使用下列方法新建.NSF 的复合应用程序。

① 基于 Lotus Notes 8 应用程序模板新建复合应用程序。

基于新的空白复合应用程序模板新建应用。要打开新应用，可以使用“文件>打开>Lotus Notes”应用，会显示新应用中空白页面。接着，可以使用“操作>编辑”应用程序在复合应用程序编辑器中开始添加组件。

② 在现有 Lotus Notes 8 应用程序中创建新的复合应用程序。

在 Domino Designer 8 中打开一个现有 Lotus Notes 应用。导航到视图复合应用程序——应用。单击按钮“新建空白 CA”。打开数据库属性框，选择应用程序启动选项，设为以复合应用方式启动。保存并退出 Domino Designer 8。现在可以使用复合应用程序编辑器来添加组件了。

③ 在 Lotus Notes8 应用程序中引入已有复合应用程序 XML 文件。

在 Domino Designer 8 中打开现有 Lotus Notes 应用，导航到视图复合应用程序——应用。单击按钮“引入 XML”。选择应用程序启动选项，设为以复合应用程序方式启动。保存并退出。接下来使用 Notes 8 中复合应用程序编辑器来添加组件。

(以上为图书第十三章内容节选，感谢阅读!)