

Lotus Domino/Notes R5 丛书



R5

Lotus Notes

R5 编程指南

北京义驰美迪技术开发有限责任公司 编



海洋出版社

Lotus Domino/Notes R5 丛书

Lotus Notes R5 编程指南

北京义驰美迪技术开发有限责任公司 编

海洋出版社

2000 年 · 北京

图书在版编目 (CIP) 数据

Lotus Notes R5 编程指南/北京义驰美迪技术开发有限
责任公司编. -北京: 海洋出版社, 2000.1

(Lotus Domino/Notes R5 丛书)

ISBN 7-5027-4876-8

I.L… II.北… III.计算机网络-应用软件, LotusNotes
R5-程序设计 IV.TP393.09

中国版本图书馆 CIP 数据核字 (1999) 第 74603 号

海洋出版社 出版发行

(100081 北京市海淀区大慧寺路 8 号)

北京四季青印刷厂印刷 新华书店发行所经销

2000 年 1 月第 1 版 2000 年 1 月北京第 1 次印刷

开本: 787×1092 1/16 印张: 14

字数: 312 千字 印数: 1~5000 册

定价: 28.00 元

海洋版图书印、装错误可随时退换

序 言

我们首先祝贺莲花公司隆重推出了 Notes R5。10 年前，Lotus Notes 的推出，极大地推动了公司之间、人与人之间的环球通讯、协同工作和协调一致的进程。10 年间，在不断的完善和创新中，Notes 对它的定义发出了挑战，从一种产品扩展演变为人们工作和生活必不可少的商务环境。

北京义驰美迪技术开发有限责任公司是莲花（中国）公司的重要合作伙伴，本地化工作是我们的主要业务之一，同时，我们也是 Lotus Notes 的忠实用户。在使用 Lotus Notes 的过程中，我们充分利用 Notes 的卓越功能，实现了公司内部的协调工作，并深深感受到 Notes 带来的方便和快捷。更重要的是莲花公司以“知识管理”为目标，将“知识管理”引入 Notes 产品中。这一点在数据库管理和网络使用方面最为突出，相信在亲身体验之后，您会同意我们的观点。在本地化过程中，我们与莲花（中国）公司紧密合作，共同努力，继 Notes R4.6 之后，又向您全面介绍经过不断改进、创新的 Notes R5。

为了使您更好地了解 Notes R5 性能，我们编写了这套丛书。您可以从中学到更多的技术与应用，Notes R5 中最激动人心的改进可以概括为以下五个方面：

新的用户界面——Notes 发布了一个引人入胜的新界面，这个界面新包括了“欢迎”页面、书签、导航器和任务按钮。从而您可以充分利用 Notes 的强大功能，无论是浏览数据库、向讨论组投稿、阅读 ISP 邮件还是网上冲浪。还有一点是极为重要的，那就是尽管界面已经改变，但是您仍可以使用旧版本的 Notes 工作台，这省去了很多麻烦。

邮件和日历的新功能——您可以使用邮件的新功能，例如：消息追踪、邮件路由控制、系统监控（Notes Minder 可以在不打开 Notes 的情况下监控新邮件）和简单的 Internet 邮件地址查询等等。同时还可以使用加强的“日历”和“预约”功能（如任务，它可以随用户从一个场所到另一场所），以及改进的“日历”管理和群组日历，来组织工作。

Internet 标准支持——Notes 包含 Internet 消息传输标准，因此您可以浏览 Internet 邮件消息、Web 网页和完全保真的新闻组（由 MINE 和 HTML 支持）和安全性（使用 SSL、S/MIME 和 X.509 验证字）。您还可以向任何 LMAP 或 POP3 服务器（如 Internet 服务供应商）发送消息。可以阅读和发送消息给 NNTP 新闻组，或搜索任何 LDAP 目录。相信这在您给电子邮件消息写地址和快速查找电子邮件时是非常有帮助的。

数据库（与服务器）性能的提升——Notes R5 的 Domino 服务器，采用最新的数据库存储结构，以获得更好的性能，并对指针结构做了调整；可以联机和执行同地数据库压缩；还可以快速重启和恢复。

除了以上四项改进之外，Notes R5 相对于 4.6 版本来说，还有一个最重要、最突出

的改进，那就是：

新一代的 Web 应用平台——Notes R5 实现了安全、可靠、标准的企业级 Web 应用，满足了 Internet/Intranet/Extranet、邮件、知识管理以及企业信息的集成，并具有以下特性：

1. Domino 应用等同于 Web 应用。在 Notes R5 中，传统的 Domino 应用与 Web 应用实现了有机的结合。
2. Domino 设计客户端，增加了新的设计工具，如页面设计、大纲设计、用户界面小程序和新的编程面板。
3. 加强了对虚拟服务器的支持，每个虚拟服务器可以具有独立的配置。
4. 提供了对 MS IIS 和 CORBA/IIOP 的支持。
5. 提高了搜索引擎的性能。新的搜索工具使您可以更加有效地工作，自动建立的索引减轻了管理人员的负担，搜索的结果高亮显示，并支持可搜索的 URL。
6. PKI（公共密钥基础设施，用来验证用户身份的策略、过程、技术集合）/CDSA（基于标准的访问 PKI 的 API 接口），由此充分利用在 ID 中已经建立的用户信息，来访问企业系统的其他资源，并允许用户简单方便地实施满足其业务需要的安全机制。
7. 对 HTML 文件的访问控制。

Notes R5 的这一突出优点，完全符合国务院对政府邮件系统的要求。国务院有关部门已经决定，将启用 Notes R5 作为政府部门通用的邮件系统平台。我们确信 Notes R5 的这些新功能组合在一起，不仅会满足政府上网的需要，更会使 Notes R5 继它的前辈之后成为世界首选的群件平台。它必将为推动中国的产业信息化和网络化进程发挥更加重要的作用。

我们公司衷心希望这套 Notes R5 丛书能为您了解和使用 Notes 带来便捷，由于我们的水平有限，难免会有疏漏之处，欢迎广大读者提出意见和批评，以利于我们和 Notes 以及大家的共同进步。

北京义驰美迪技术开发有限责任公司

1999.12.2

目 次

第 1 章 编程概述	1
1.1 在 Domino 中编程	1
1.2 在何处使用 Script 和公式	1
1.3 Domino 对象表	2
1.4 便捷图标	4
1.5 复制公式	5
1.6 代 理	6
1.7 操 作	10
1.8 热 点	12
1.9 编写表单、选择和列公式	14
1.10 窗口标题、区段存取和插入子表单公式	16
1.11 区段标题和隐藏段落公式	17
1.12 事 件	18
1.13 域设计公式	20
1.14 事件描述	22
1.15 事件序列	25
第 2 章 用户界面	28
2.1 访问“编程”窗格	28
2.2 使用“编程”窗格	28
2.3 使用“编程”窗格中的 Java 界面	34
2.4 在代理中编写 Java	35
2.5 将 Java 引入到“编程”窗格	36
2.6 使用 AgentRunner	37
2.7 在“编程”窗格中编写 LotusScript	38
2.8 在“编程”窗格中编写 JavaScript	40
2.9 使用 LotusScript 调试器	41
2.10 使用 Script 库	45
2.11 在程序窗格中编写程序	46
第 3 章 LotusScript Notes 类编码指南	48
3.1 使用 Notes 类	48
3.2 在 Notes 中使用 LotusScript 特性的方法	49
3.3 访问 Notes 数据库的方法	54
3.4 访问 Notes 视图和文件夹的方法	66
3.5 访问 Notes 文档的方法	70

3.6 访问 Notes 条目的方法	87
3.7 访问 Notes 会话的方法	102
3.8 访问支持 Notes 对象的方法	107
3.9 使用 LotusScript 通过 ODBC 访问外部数据库的方法	112
3.10 连接到数据源	113
3.11 查 询 表	117
3.12 检查结果集合	121
3.13 更 新 表	128
3.14 处理 ODBC 错误	133
3.15 处理 ODBC 事件	135
3.16 处理运行错误	137
3.17 使用 OLE 定制控件	156
3.18 在 Visual Basic 中使用 Notes 类	159
第 4 章 Lotus Notes 公式语言规则	161
4.1 使用语法规则	161
4.2 使用变量	162
4.3 使用常量	164
4.4 使用运算符	166
4.5 使用@Function	175
4.6 使用关键字	178
4.7 在公式中指定表单和视图的名称	179
4.8 调试公式	179
第 5 章 Lotus Notes 公式语言编码指南	181
5.1 公 式	181
5.2 编写消息和取得用户输入	188
5.3 处理错误	192
5.4 使用@Function	194
5.5 使用@Command	196
5.6 执行字符串操作	197
5.7 执行算术运算	204
5.8 执行时间-日期计算	207
5.9 访问用户环境	209
5.10 访问当前数据库和视图	210
5.11 通过公式语言访问当前文档	212
5.12 访问当前文档和数据库以外的数据	214
5.13 使用@Function 通过 LS:DO 访问外部数据库	216

第 1 章 编程概述

本章描述如何将 Java、JavaScript、LotusScript 和公式代码附加到 Domino 对象上。以下是将讨论的主题：

- 在 Domino 中编程。
- 何处使用 Script 和公式。
- Domino 对象表。
- 事件描述。
- 事件序列。

1.1 在 Domino 中编程

公式、LotusScript、Java 和 JavaScript 代码为 Domino 设计者提供了完整的编程界面。可以根据需要将代码附加到多个对象。例如：如果在表单中创建一个计算域，则可附加公式来计算此域的值。也可向域的 onFocus 事件附加 JavaScript 代码，此代码将在用户将焦点放置在域上的任何时候执行。还可以决定创建一个公式代理、LotusScript 代理或 Java 代理来自动定期更新数据库中的所有文档。

Domino 还为 Java 应用程序和小程序提供了一个编程界面。这些应用程序和小程序可以通过访问安装的 Domino 软件在本地操作，也可以通过使用 IIOP 协议连接到一个 Domino 服务器进行远程操作。

Domino Designer 模板提供了模板和通常情况下使用代码的样例。对于这些情况，不一定要学习如何编程。除这些情况外，本书可作为在 Domino 中编程的完整的指南和参考资料。

1.2 在何处使用 Script 和公式

在编写代码之前，要确保简单操作不能执行此任务。可使用不需要编程的表单或视图中的简单操作来设计一些对象。

当特定 Domino 对象决定什么编程界面是可用的时候，请考虑以下指导方针：

- 公式是具有类似编程语言特征的表达式。例如：可将值赋予变量，并且使用限制的控制逻辑。公式语言通过调用 @Function 和 @Command 与 Domino Designer 接口。
- 通常情况下，如果用户当前正在处理对象，最好使用使用公式编程，例如：返回域的缺省值或确定视图的选择条件。此外，公式在某些场合提供更好的性能，并且对简单应用程序来说比较方便。
- JavaScript 是跨平台、面向对象的描述性语言。通过从 Objects 附签中选择

JSHeader, 可在编程窗格中编写 Headerscript, 并且在 Script 区键入 script。Script 也可被附加到诸如 onClick 的特定事件上, 或者被附加到诸如按钮的对象上。不能在代理中编写 JavaScript。Domino 监控用户 script 的编译和加载。但是不将 JavaScript 存储在已编译的表单中。

JavaScript 最好用于 Web 应用程序, 或者单个应用程序被同时用于 Notes 和 Web 环境时。

- LotusScript 是完全面向对象的编程语言。它通过预定义的类与 Domino 接口。Domino 监控用户代码的编译和加载, 并且自动包含 Domino 的类定义。访问现有的对象最好使用 LotusScript, 例如: 根据其他文档的值来更改一个文档中的值。LotusScript 提供了一些公式没有的功能, 例如: 操作数据库存取控制列表(ACL)的能力。

Java 是一种完全面向对象的编程语言, 它与 Domino 的接口是通过预定义的类实现的。在代理方面它可与 LotusScript 相比, 但是它不能附加到 DominoUI 中的事件中。Domino 监控用户的代理代码的编译和载入, 代码可以自己写也可以引入。

Java 可以用在代理、Java 应用程序和小程序中, 可以在 Domino 以外编写和编译, 并且可以通过类接口访问 Domino。

1.3 Domino 对象表

下表概括了 Domino 中的可编程对象。它指定了对象的范围, 以及对象是否支持简单操作、公式、LotusScript、Java 或 JavaScript。

表 1-1

范 围	Domino 对象	支 持
工作台 数据库	便捷图标	公式
	复制公式	公式
	代理	公式 简单操作 LotusScript Java
导航器设计	事件	LotusScript 公式
	热点	LotusScript 公式 简单操作
	表单公式	公式
视图或文件夹设计	选择公式	公式 简易式 (简单操作)
	列公式	公式 简单操作
		域

续 表

范 围	Domino 对象	支 持
表单设计	操作	公式 简单操作 LotusScript JavaScript
	隐藏操作公式	公式
	事件	公式 LotusScript
	窗口标题公式	公式
	区段标题公式	公式 文本
	区段存取公式	公式
	插入子表单公式	公式
	隐藏段落公式	公式
	操作	公式 简单操作 LotusScript JavaScript
	隐藏操作公式	公式
表单中的布局区域设计	事件	公式 LotusScript JavaScript
	热点（按钮或操作）	公式 简单操作 LotusScript JavaScript
	热点（连接或公式弹出）	公式
	热点（操作）	公式 简单操作 LotusScript JavaScript
表单中的域设计	缺省值公式	公式
	输入转换公式	公式
	输入校验公式	公式
	计算域的值公式	公式
	关键字域公式	公式

续 表

范 围	Domino 对象	支 持
文档（编辑模式）RTF 文本域	事件	LotusScript JavaScript
	区段标题公式	公式 文本
	隐藏段落公式	公式
	热点（按钮或操作）	公式 简单操作 LotusScript JavaScript
	热点（链接或弹出公式）	公式

1.4 便捷图标

单击便捷图标按钮时，它便执行一个公式。Domino 带有 100 多个预定义的便捷图标按钮，其中包含了大多数菜单命令。Domino 也提供了另外一些定制的便捷图标按钮，可以为它们编写自己的公式。便捷图标按钮只与每个用户的工作台有关，而与数据库无关，也不能在用户之间共享。

便捷图标公式在用户的工作站上运行。

样例：便捷图标

- (1) 以下便捷图标打开“DOC”服务器上的“names.nsf”数据库，并切换到“People”视图：

```
@Command([FileOpenDatabase]; "DOC" : "NAMES.NSF"; "People")
```

- (2) 以下便捷图标公式给用户数据库目录中的数据库列表，并且打开用户选定的数据库。第一个@DbColumn 将“Database by_Replica ID”视图中第四列的数值列表赋给临时变量“titles”。第二个@DbColumn 将“Database by_Replica ID”视图中第二列的数值列表赋给临时变量“servers”。第三个@DbColumn 将“Database by_Replica ID”视图中第三列的数值列表赋给临时变量“databases”。在@Prompt 中为用户显示临时变量“titles”、“servers”和“databases”的组合列表。本公式从@Prompt 的返回值中解析出服务器和数据库的名称，并在“FileOpenDatabase”命令中调用。

```
titles := @DbColumn(""); "doc":"CATALOG.NSF"; "Databases by _Replica ID"; 4);
servers := @DbColumn(""); "doc":"CATALOG.NSF"; "Databases by _Replica ID";
2);
databases := @DbColumn(""); "doc":"CATALOG.NSF"; "Databases by _Replica
ID"; 3);
list := titles + " *-* " + servers + " *-* " + databases;
member := @Prompt([OKCANCELLIST]; "Open Database"; "Select a database";
```

```
""; list);
server := @Left(@Right(member; " *-* "); " *:* ");
database := @Right(member; " *:* ");
@Command([FileOpenDatabase]; server:database)
```

1.5 复制公式

复制公式可以选择在复制过程中将哪些文档写入当前数据库。

复制公式必须以 **SELECT** 语句结尾。如果公式中最后一个语句是逻辑表达式，Domino 通过插入 **SELECT** 关键字将它转换成 **SELECT** 语句。

复制公式在包含了公式的数据库所在的服务器或工作站上运行。

样例：复制公式

- (1) 缺省的数据库复制公式是复制所有文档。

```
SELECT @All
```

- (2) 以下公式只复制那些“Region”域中包含“East”的文档。

```
SELECT Region = "East"
```

- (3) 以下公式适用于邮件数据库的副本中，不复制从“Arnold Runion”或“Mary Chen”发送的邮件文档。

```
SELECT !(From="Arnold Runion" | From="Mary Chen")
```

1.5.1 使用关键字

公式语言包括一套执行特殊功能的关键字：

表 1-2

关键字语法	描述
DEFAULT <i>fieldName</i> := <i>value</i>	将一个值与域关联。如果域在正被处理的文档中已经存在，则其当前值被使用。如果域不存在，则当作域已经存在来处理该文档，并且使用 DEFAULT 值
ENVIRONMENT <i>variable</i> := <i>textValue</i>	指定一个值为环境变量，环境变量放置在用户的 NOTES.INI 文件中 (Windows、OS/2、UNIX) 或 NotesPreferences 文件中 (Macintosh)
FIELD <i>fieldName</i> := <i>value</i>	将一个值指定给当前文档中的一个域。如果域不存在，则创建该域，如果已经存在，替换它的内容
REM ["remarks"]	在公式中加入注释而不影响它的功能
SELECT <i>logicalValue</i>	指定当前文档在视图选项、复制和代理公式中是否有效

关键字一般是语句中的第一个词。通常约定关键字用大写字母输入。可以输入小写，但是 Notes 在保存公式时将其转换为大写。

指定给关键字的 *fieldName* 和 *variable* 是名称而不是文本常量，不要将它们包括在括号中。

样例：使用关键字

- (1) (DEFAULT)。如果域 KeyThought 存在，就用该域中的值提供给计算域使用。

如果域 KeyThought 不存在，则使用 Topic 的值。

DEFAULT KeyThought := Topic;

KeyThought

本公式等价于：

@If(@IsAvailable(KeyThought); KeyThought; Topic)

- (2) (ENVIRONMENT)。把一个数字转换成文本，并保存到环境变量之中。

ENVIRONMENT OrderNumber := @Text(NewOrderNumber)

- (3) (FIELD)。本公式将 “Inc.” 添加到 Company 域的值中。

@If(@Matches(@LowerCase(Company); "*", inc*"); @Return(""); "");

FIELD Company := Company + ", Inc.";

- (4) (FIELD)。本公式创建一个名为 CompanyName 的新域来保存公司的名称并在其后加上 “Inc.”。这个域只有在加到表单设计中后才可见，但仍然可以通过公式访问它。

FIELD CompanyName := Company + ", Inc.";

- (5) (FIELD)。本公式删除域 CompanyName。

FIELD CompanyName := @DeleteField;

- (6) (REM)。本公式在代码前面包含五行注释。

REM "6/15/95";

REM "The following formula calculates the date";

REM "for the DueDate field";

REM "DueDate is the Date field + thirty days";

REM;

@Adjust(Date; 0;0;30;0;0;0)

- (7) (SELECT)。本公式选择数据库中的所有文档。

SELECT @All

- (8) (SELECT)。这个视图选择公式只选择由 Product Specification 表单创建的文档或答复文档。

SELECT Form="Product Specification" | @IsResponseDoc

- (9) (SELECT)。以下的公式将 Status 域的值改为 “Closed” (Categories 域值为 “Unsigned Contracts” 的文档除外)。

SELECT Categories != "Unsigned Contracts";

FIELD Status := "Closed"

1.6 代理

代理是可通过许多机制触发的用户过程。代理运行在：

- 用户的工作站。如果代理的触发器是：“手动选取”、“修改文档后”、“文档被粘贴时”。
- 包含此代理的服务器或工作站。如果代理的触发器是：“新邮件到达时”、“定时执行”。

可以用公式语言、LotusScript 和 Java 来编写代理，也可以使用 Domino 提供的代理。因此在编写代理之前，请查看一下 Domino 提供的代理是否可以此工作。

基于公式的代理可巡回地在数据库文档上运行。可以通过代理接口应用搜索条件来指定哪些文档将被处理。公式中的 SELECT 语句更好地限制了搜索。如果公式中没有包含一个 SELECT 语句，则 DominoDesigner 添加一个 SELECT@All 语句（SELECT@All 语句除外）。SELECT 语句必须是公式中的第一个有效语句。

LotusScript 和 Java 代理只能运行一次。可以通过语言结构提供搜索标准和巡回方式，通过代理接口应用的搜索标准只能通过 NotesDatabase(LotusScript)中的 UnprocessedDocuments 或 AgentContext(Java)中的 getUnprocessedDocuments 实现。

样例：代理

- (1) 以下 LotusScript 代码根据数据库每个文档中的“TotalSales”域的值，为“Category”域写入值。同使用 Java 和公式的样例 2 和样例 3 比较，用 Script 写的样例需要比公式方法更多行代码，但可以包含查询要处理的文档的运算法则。

Sub Initialize

```
Dim session As New NotesSession
Dim db As NotesDatabase
Dim dc As NotesDocumentCollection
Dim doc As NotesDocument
Set db = session.CurrentDatabase
Set dc = db.AllDocuments
Set doc = dc.GetFirstDocument
While Not(doc Is Nothing)
    category = doc.Category
    totalSales = doc.TotalSales
    Select Case totalSales(0)
        Case Is >= 200000 : category(0) = "Above Quota"
        Case Is >= 100000 : category(0) = "OK"
        Case Else : category(0) = "Below Quota"
    End Select
    doc.Category = category
    Call doc.Save(True, False)
    Set doc = dc.GetNextDocument(doc)

```

Wend

End Sub

- (2) 以下 Java 代理根据数据库每个文档中的“TotalSales”域的值，为“Category”域写入值。同使用 LotusScript 和公式的样例 1 和样例 3 比较，使用 LotusScript，Java 代码包含查询要处理的文档的运算法则。

```
import lotus.domino.*;
public class JavaAgent extends AgentBase {
```

```

public void NotesMain() {
    try {
        Session session = getSession();
        AgentContext agentContext = session.getAgentContext();
        // (Your code goes here)
        Database db = agentContext.getCurrentDatabase();
        DocumentCollection dc = db.getAllDocuments();
        Document doc = dc.getFirstDocument();
        while (doc != null) {
            double totalSales = doc.getItemValueDouble("TotalSales");
            if (totalSales >= 200000)
                doc.replaceItemValue("Category", "Above quota");
            else if (totalSales >= 100000)
                doc.replaceItemValue("Category", "OK");
            else
                doc.replaceItemValue("Category", "Below quota");
            doc.save(true, false);
            doc = dc.getNextDocument();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

- (3) 以下的公式根据数据库每个文档中的“TotalSales”域的值，为“Category”域写入值。以下的公式在数据库每个文档中 TotalSales 域数值的基础上，通过判断以后向 Category 域中写入数值，假设所有的文档都被选定进行处理。同使用 LotusScript 和 Java 的例 1 和例 2 比较，该公式将外部条件和 SELECT 语句相结合选择文档，并对每一个选定的文档执行一次。数据声明是隐含的，公式语法比较费解，使代码比较紧凑。

```

FIELD Category := @If(TotalSales >= 200000; "Above Quota"; TotalSales >=
100000; "OK"; "Below Quota");
SELECT @All

```

- (4) 以下公式根据“TotalSales”域的值，为选定的文档的“Category”域写入值。如果语句中不包含“SELECT @All”语句，则 SELECT 语句必须出现在公式中的应用语句之前。

```

SELECT TotalSales >= 200000;
FIELD Category := "Above Quota"

```

- (5) 以下 LotusScript 代码查找某一天数据库中所有 OrderTotal 域的总和，并且向数据库中写入一个包含日总计的新记录。数据库中每一个记录都包含

“OrderNumber”、“Date”和“OrderTotal”域。以下的 Script 查找数据库中所有的文档，并使用一个循环和一个日期比较来限定处理当天的文档。对于每一个文档，Script 把“OrderTotal”域的值累加到“dailyTotal”变量中。Script 把词语“DAILY TOTAL”放入所写文档的“OrderNumber”域中，把“dailyTotal”值放入“OrderTotal”域中。

Sub Initialize

```

    Dim session As New NotesSession
    Dim db As NotesDatabase
    Dim dc As NotesDocumentCollection
    Dim doc As NotesDocument
    Dim odate As New NotesDateTime("")
    Dim dateToday As New NotesDateTime("Today")
    Set db = session.CurrentDatabase
    Set dc = db.AllDocuments
    dailyTotal = 0
    Set doc = dc.GetFirstDocument
    While Not(doc Is Nothing)
        odate = doc.Date
        dn = Datenum(Year(odate(0)), _
            Month(odate(0)), Day(odate(0)))
        orderNumber = doc.OrderNumber
        If dn = Today _
            And orderNumber(0) <> "DAILY TOTAL" Then
            orderTotal = doc.OrderTotal
            dailyTotal = dailyTotal + orderTotal(0)
        End If
        Set doc = dc.GetNextDocument(doc)
    Wend
    Dim docNew As New NotesDocument(db)
    Set itm = _
        docNew.AppendItemValue("OrderNumber", "DAILY TOTAL")
    Set itm = _
        docNew.AppendItemValue("OrderTotal", dailyTotal)
    Set itm = _
        docNew.AppendItemValue("Date", Date$)
    Call docNew.Save(True, False)

```

End Sub

1.7 操 作

操作是与视图或表单相关联的定制处理过程。当打开视图或打开使用该表单创建的文档时，操作将显示在“操作”菜单命令中或者出现在操作条的按钮中，供用户使用。

可以使用“公式条件为真时隐藏操作”公式，有条件地限制在菜单中或操作条中操作使用。

操作在用户的工作站上运行。

样例：操作

- (1) 以下 LotusScript 操作打印运行该 Script 的计算机的 Domino 数据目录下的每一个 Domino 数据库名称。NotesDcDirectory 中的 FirstDatabase 和 NextDatabase 成员函数查询指定服务器上的所有数据库，如果未指定服务器，则缺省指向当前计算机。

```
Sub Click(Source As Button)
    Dim directory As New NotesDbDirectory("")
    Dim db As NotesDatabase
    Set db = directory.GetFirstDatabase(DATABASE)
    While Not(db Is Nothing)
        Messagebox db.Title
        Set db = directory.GetNextDatabase()
    Wend
    Messagebox "The End"
End Sub
```

- (2) 以下的公式列出“CORP1”服务器上通讯录中的姓名，让用户选取任意数目的姓名，用一个逗号和一个空格作为分隔符将选定的姓名合并为一个字符串，并且在当前域中插入该字符串。当用户的插入点位于邮件数据库中的“SendTo”域中时，使用本操作效果最突出。

```
last := @Left(@DbColumn(""); "CORP1" : "NAMES.NSF"; "People"; 1); ",";
first := @RightBack(@DbColumn(""); "CORP1" : "NAMES.NSF"; "People"; 2);
" ";
list := first + " " + last;
name := @Prompt([OKCANCELLISTMULT]; "Send To"; "Who are you sending
this memo to?"; ""; list);
@Command([EditInsertText]; @Implode(name; ", "));
```

- (3) 以下的公式是一个“公式条件为真时隐藏操作”选项中的公式。域“OrderTatol”小于或等于 100 时，当前操作显示在“操作”菜单或操作条中。（如果用户在当前文档中的“OrderTotal”域中输入新值，则在新的域值发生作用以前必须刷新文档。）

```
OrderTotal > 100
```