

IBM DB2 Alphablox



管理员指南

版本 8.4

IBM DB2 Alphablox



管理员指南

版本 8.4

注意！

在使用本资料及其支持的产品之前，请阅读第 147 页的『声明』中的信息。

第四版（2006 年 9 月）

本版本适用于 IBM DB2 Alphablox for Linux, UNIX and Windows（产品号 5724-L14）的 V8R4 及所有后续发行版和修订版，直到在新版本中另有声明为止。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何信息，有权利以任何它认为适当的方式使用或分发，而不必对您负任何责任。

© Copyright Alphablox Corporation 1996, 2006. All rights reserved.

© Copyright International Business Machines Corporation 1996, 2006. All rights reserved.

目录

第 1 章 DB2 Alphablox 概述	1
DB2 Alphablox 概述	1
J2EE 环境中的 DB2 Alphablox	1
在应用程序服务器中运行的优点	2
WebSphere 和 WebLogic Application Server 配置	2
Apache Tomcat 配置	3
DB2 Alphablox 平台的组件	3
DB2 Alphablox	3
数据适配器	3
Blox 组件	4
DHTML 客户机	5
远程管理能力	6
集中的应用程序管理	6
N 层体系结构	6
DB2 Alphablox 应用程序	6
Application Studio	7
DB2 Alphablox 体系结构	7
服务管理器	8
请求管理器	8
会话管理器	8
用户管理器	8
应用程序管理器	9
数据管理器	9
存储库管理器	9
文件管理器	10
控制台管理器	10
DB2 Alphablox Cube Manager	10
集群管理器	10
DB2 Alphablox 入门任务列表	10
第 2 章 DB2 Alphablox 应用程序	11
DB2 Alphablox 应用程序的类型	11
数据表示方式	11
转换为 PDF	11
使用 XML 来定制绘制方式	12
关系报告应用程序	12
DB2 Alphablox 应用程序的组件	12
JavaServer Pages (JSP)	12
Blox 组件	12
第 3 章 DB2 Alphablox 主页	15
DB2 Alphablox 主页概述	15
“应用程序”选项卡	15
其他应用程序	15
“管理”选项卡	15
常规	16
一般属性	16
定制属性	16
运行时管理	16
控制台	16

Portal 主题实用程序	16
应用程序迁移	17
组	17
用户	17
角色	17
应用程序	17
数据源	17
DB2 Alphablox 立方体	18
“组装”选项卡: Application Studio	18
工作台	18
示例	18
样本数据	18
右上角的链接	18
我的概要文件	18
帮助	18
第 4 章 基本管理任务	21
启动 DB2 Alphablox	21
访问 DB2 Alphablox	21
停止 DB2 Alphablox	21
在 Apache Tomcat 安装上通过关闭脚本来停止 DB2 Alphablox	21
在 Windows 系统上:	22
在 Linux 和 UNIX 系统上:	22
在 Apache Tomcat 安装上从服务控制面板中停止 DB2 Alphablox	22
使用 Essbase 客户机库实用程序	22
其他管理任务和信息	23
第 5 章 客户机管理	25
DHTML 客户机管理	25
受支持的 DHTML 客户机配置	25
DHTML 客户机问题	25
CSS	25
弹出窗口	25
第 6 章 应用程序定义	27
DB2 Alphablox 应用程序定义	27
应用程序名	28
WEB-INF 目录	28
定义新的应用程序	28
在与 WebSphere 一起运行时定义应用程序	29
在 DB2 Alphablox 创建一个应用程序, 然后在 WebSphere 中注册	30
将现有的 WebSphere 应用程序导入到 DB2 Alphablox 中	31
修改现有的应用程序定义	31
删除现有的应用程序定义	32
当使用 WebLogic 集群时定义应用程序	32
向外部 Web 服务器注册应用程序	33
在 Apache HTTP Server 上注册应用程序	33
Apache Tomcat 5.5 上的 DB2 Alphablox 8.4.1	33
Apache Tomcat 3.2.4 上的 DB2 Alphablox 8.4	33
停止并重新启动 Apache HTTP Server	33
在 Microsoft Internet Information Server (IIS) Web 服务器上注册应用程序	34
在 Alphablox 8.4 上的 Sun iPlanet Web 服务器上注册应用程序	34
添加文档目录	34
对 DB2 Alphablox 样式添加适当的指定	34
导入现有的 J2EE 应用程序	35

第 7 章 数据源定义	37
定义新的数据源	37
更改或删除现有数据源定义	38
更改现有数据源定义	38
删除现有数据源定义	38
为 Microsoft Analysis Services 数据源设置 Microsoft 认证	38
设置 Windows 用户权限	39
设置 Windows 用户的权限	39
设置 Windows 用户的权限	39
配置 Windows 服务	40
确保在 Microsoft Analysis Services 中配置了用户	40
使用 JDBC 数据源	41
为 Sybase JConnect 关系驱动程序设置环境	41
设置 JDBC 跟踪	41
将受支持的 JDBC 驱动程序更新到另一个版本	42
添加其他 JDBC 驱动程序	42
修改类路径设置	42
WebSphere	43
WebLogic	43
Tomcat	43
第 8 章 用户定义	45
创建新用户	45
更改或删除现有的组或用户	45
更改现有用户的属性	46
删除现有的用户	46
更改用户所属的组	46
第 9 章 组定义	49
创建新组	49
了解子组	49
更改或删除现有的组	50
更改现有的组	50
删除现有的组	51
第 10 章 角色定义	53
定义新角色	53
更改和删除现有的角色	53
更改用户或组所属的角色	54
删除现有的角色	54
第 11 章 安全性和认证	55
DB2 Alphablox 认证方式和安全方式	55
Alphablox 8.4.1 中的安全模型	55
基于 Java 认证和授权服务 (JAAS) 的 Alphablox 登录模块	56
Tomcat 5.5 中的 JNDI 域配置	56
在 Apache Tomcat 5.5 中添加 Alphablox JNDI 域	56
Admin 权限与用户权限	57
除去应用程序的 Guest 登录权限	57
应用程序服务器安全域和应用程序	57
Web 服务器认证与 DB2 Alphablox 认证	57
在带有 Apache Tomcat 3.2.4 的 Alphablox 8.4 上使用 Sun iPlanet Web 服务器安全选项	58
为 IIS NTLM 设置 Microsoft 安全选项	58
安装 Microsoft IIS	59
安装 DB2 Alphablox 并选择 Microsoft IIS 作为 Web 服务器	59
在 Microsoft IIS 中配置安全设置	59

为 NTLM 创建名为 admin 的 Windows 本地用户	61
在 Tomcat 5.5 中为 Alphablox 8.4.1 配置 NTLM 安全性	61
登录到 DB2 Alphablox	61
配置 DB2 Alphablox 以使用基于 Web 服务器的安全性	62
配置自动生成用户帐户	62
过滤 IP 地址	63
设置目录权限	63
禁用目录浏览	63
第 12 章 扩展 DB2 Alphablox	65
概述	65
计算扩展	65
用户管理器扩展	65
DHTML 客户机扩展	66
配置 DB2 Alphablox 以支持定制 Java 类	66
设置类路径	66
第 13 章 配置 DB2 Alphablox 属性	69
DB2 Alphablox 管理任务	69
配置启动属性	69
配置系统属性	70
指定 Telnet 端口	71
配置 DB2 Alphablox 立方体管理器	72
定制属性定义	73
定义新的用户属性	73
更改用户属性	74
删除用户属性	74
定义新的定制应用程序属性	74
更改应用程序属性	75
删除应用程序属性	75
创建和管理注释集合	75
访问“注释管理”对话框	75
定义和访问数据源	76
定义注释集合	76
使用 Microsoft SQL Server 或 Sybase 数据库时的注释集合	77
显示注释集合定义	77
删除注释集合	77
添加和显示注释	77
创建远程 PDF 处理器	78
配置远程 PDF 处理器	78
配置远程 PDF 报告管理	78
DB2 Alphablox 日志文件	79
日志文件滚动时间间隔设置	79
日志文件名	80
管理日志文件	80
第 14 章 用户管理器和个性化 (Alphablox 8.4.1)	81
个性化管理器	81
访问 JNDI 用户和组属性	81
第 15 章 用户管理器 (Alphablox 8.4)	85
DB2 Alphablox 用户管理器概述	85
可扩展用户管理器	86
基于 LDAP 的用户管理器	87
配置 DB2 Alphablox 以使用 LDAP 用户管理器	87
设置基于 LDAP 的用户管理器属性	88

访问定制用户属性	88
运行时行为	89
可扩展用户管理器 Telnet 控制台命令	89
设置缺省存储库	90
除去不再存在于外部用户存储库中的用户和组	90
可扩展用户管理器接口	90
定制安全性实现	91
单点登录	91
定制安全性示例	92
示例 1: 将 DB2 Alphablox 设置为使用外部用户管理器	92
示例 2: 将 DB2 Alphablox 设置为使用另一个用户类	93
示例 3: 将 DB2 Alphablox 设置为使用另一个组类	93
接口方法交叉引用	94
IUserManager 接口	94
findGroup().	95
findUser().	95
getExternalProperties().	95
getPrincipleUserName().	96
hasExternalEditor().	96
resume().	96
setCaseSensitiveGroups().	97
setCaseSensitiveUsers().	97
start().	97
stop().	98
suspend().	98
IUser 接口	98
authenticate().	98
authorize().	99
getEmail().	99
getFullName().	99
getName().	100
getPassword().	100
getPropertiesSubset().	100
isUserInRole().	101
refresh().	101
IGroup 接口	101
containsGroup().	101
containsUser().	102
getName().	102
getPropertiesSubset().	102
refresh().	103
第 16 章 使用数据库存储库	105
DB2 Alphablox 存储库概述	105
DB2 Alphablox 环境中的存储库	105
关系存储库的优点	105
配置 DB2 Alphablox 存储库	105
检查存储库类型	106
使用存储库转换实用程序	106
启动存储库转换实用程序	106
存储库转换实用程序交互式命令行选项	106
从文件系统转换为数据库	107
从数据库转换为文件系统	108
配置实例以使用现有的存储库	109
命令行语法	110

第 17 章 使用连接池	113
连接池 - 概述	113
MDB 连接池	113
DB2 OLAP Server 和 Hyperion Essbase 连接池	113
Microsoft Analysis Services 和连接池	113
启用连接池	114
使用连接池	114
限制连接池	114
调整连接池	115
RDB 连接池	116
DB2 Alphablox 对 RDB 连接池的使用	116
DB2 Alphablox 数据源与 RDB 连接池	116
DB2 Alphablox 存储库与 RDB 连接池	117
对 BEA WebLogic 配置连接池	117
第 18 章 使用集群环境	119
集群环境概述	119
WebSphere 集群环境	119
WebLogic 集群环境	119
在 WebLogic 集群环境中配置和安装 DB2 Alphablox	119
在 WebLogic 集群环境中创建新应用程序	120
使用 WebLogic 垂直集群	120
集群控制台命令	120
第 19 章 DB2 Alphablox 控制台命令	123
访问控制台	123
HTML 控制台	123
Telnet 控制台	123
命令语法	123
命令缩写	124
控制台命令列表	124
特定于 Essbase 的控制台命令	127
RESOLVEALIASSTOBASEMEMBERS 命令	127
SHOW OUTLINECACHE 命令	128
DELETE OUTLINECACHE 命令	128
有关控制台命令的说明	129
查看一般属性	129
消息级别	129
通过控制台运行文本文件	130
DB2 Alphablox 日志消息	130
第 20 章 管理 Alphablox FastForward 应用程序	131
概述	131
FastForward 用户的角色	131
应用程序管理员	131
模板开发者	131
用户	132
FastForward 应用程序的系统需求	132
创建 Alphablox FastForward 应用程序	132
更改管理员角色	133
管理 FastForward 应用程序	133
报告访问类别和安全性	133
已发布的报告	133
专用报告和组报告	133
布局和控件	134
导航菜单	134

管理报告	134
创建报告	134
修改报告	134
删除报告	134
移动报告	135
管理文件夹	135
创建文件夹	135
修改文件夹	135
删除文件夹	135
移动文件夹	135
管理应用程序属性	135
使用应用程序日志	136
附录. OLAP 术语和概念	137
二维分析	137
二维销售表	137
多维分析	137
数据立方体	138
多维销售矩阵	138
OLAP 数据库术语	139
词汇表	141
声明	147
商标	148
索引	151

第 1 章 DB2 Alphablox 概述

DB2 Alphablox 在 J2EE 应用程序服务器环境中运行并提供用于创建基于 Web 的分析应用程序的服务。DB2 Alphablox 可以与主要的应用程序服务器（包括 IBM® WebSphere®、BEA WebLogic 和 Apache Tomcat）集成。本章提供了对 DB2 Alphablox 的概述，描述了 DB2 Alphablox 融入到 J2EE 环境的方式，并描述了 DB2 Alphablox 的体系结构。

DB2 Alphablox 概述

DB2 Alphablox 提供了快速创建定制的基于 Web 的应用程序的能力，这些应用程序能够融入企业基础结构中并且能够供大范围的用户（既包括企业防火墙内侧的用户也包括企业防火墙外侧的用户）访问。使用 DB2 Alphablox 平台构建的应用程序在标准的 Web 浏览器中运行，这允许在 Web 浏览器中进行实时并且可高度定制的多维分析。

以下是 DB2 Alphablox 平台提供一些功能：

- 访问多维数据库及关系数据库中的数据并与之进行交互
- 创建来源于关系数据库的结构化报告
- 从各种各样的图表中进行选择以显示数据
- 创建将数据写回到数据库中的应用程序，这在“假设分析”财务计划应用程序中特别有用
- 对于多维数据源，允许用户与不同级别的数据进行交互（例如，过滤器和向下钻取等等）以便以交互方式显示所需的精确数据视图
- 用户访问直观的用户界面，这使得数据分析变得简单方便并且功能强大
- 单个应用程序可以访问多个数据源
- 能够集成到各种各样的企业基础结构组件中，这些组件包括应用程序服务器（IBM WebSphere 和 BEA WebLogic）

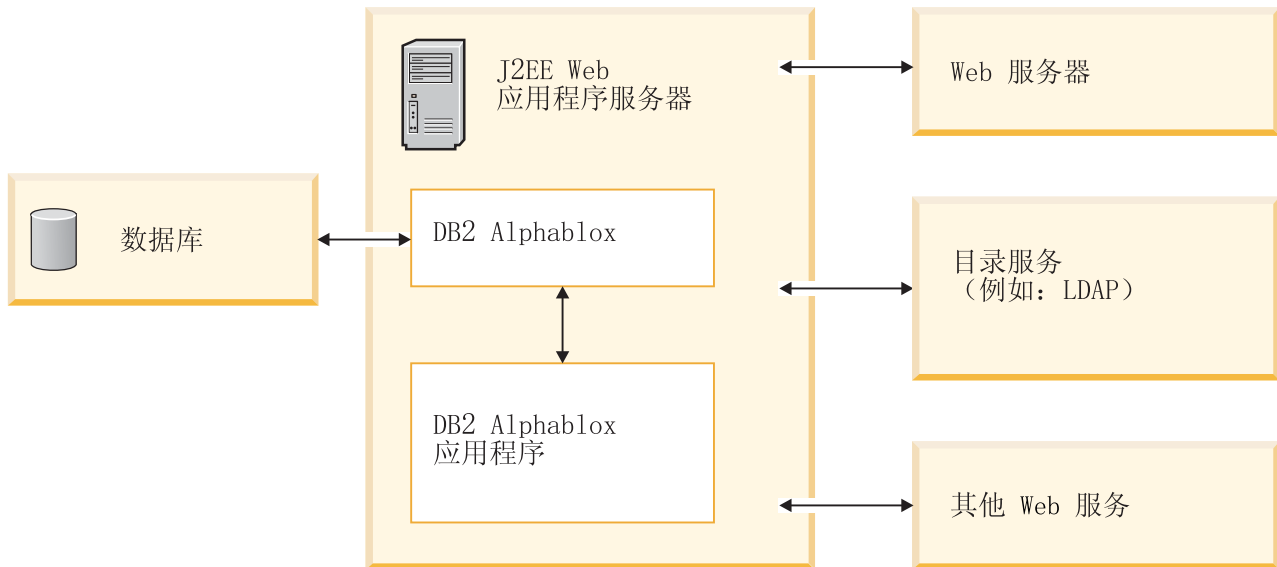
DB2 Alphablox 提供了各种各样的应用程序编程接口（API），因此开发者可以创建定制的应用程序。DB2 Alphablox API 是以 Java™ 编程语言编写的，应用程序开发者可以使用在服务器上执行的 Java 或者通过在浏览器中解释的 JavaScript™ 来访问它们。

本章的其余部分描述了 DB2 Alphablox 融入到 J2EE 环境的方式，说明了 DB2 Alphablox 的组件，并描述了 DB2 Alphablox 的体系结构以及 DB2 Alphablox 的核心组件。要了解有关创建 DB2 Alphablox 应用程序的详细信息，请参阅《*开发者指南*》。要了解有关 DB2 Alphablox API 的语法和参考信息，请参阅 *Developer's Reference*。

J2EE 环境中的 DB2 Alphablox

DB2 Alphablox 在 J2EE 应用程序服务器（IBM WebSphere 或 BEA WebLogic）或 Apache Tomcat 中运行。要获取受支持的应用程序服务器及其版本的列表，请参阅《*安装指南*》。

下图显示了 DB2 Alphablox 融入到 J2EE 环境的方式。



在应用程序服务器中运行的优点

在应用程序服务器中运行有几个优点:

- 能够方便地将基于开放标准的 J2EE 组件集成到 DB2 Alphablox 环境中
- 提供了对各种 Web Service 提供的服务的访问
- 提供了对由 Java 运行时环境提供的以及由特定应用程序服务器中任何 Java 扩展提供的所有服务的访问
- 能够使 DB2 Alphablox 专注于提供一个以分析为目标的平台, 而将诸如 Web 服务和安全性等问题留给专门投身于这些领域的供应商解决
- 为熟悉 J2EE 技术 (如 JavaServer Pages (JSP)、Java、JavaBeans™ 组件和 XML) 应用的 J2EE 开发者提供了一个平台。

与应用程序服务器的紧密集成使 DB2 Alphablox 具备所有这些优点, 同时使开发者能够访问丰富的 DB2 Alphablox API 集。

WebSphere 和 WebLogic Application Server 配置

可以将 DB2 Alphablox 配置为在 IBM WebSphere 或 BEA WebLogic 应用程序服务器中运行。无论 DB2 Alphablox 是与 WebSphere、WebLogic 还是与 Tomcat 一起运行, 其基本操作都是一样的; 只有一些不重要的管理详细信息有所不同 (例如, 在 DB2 Alphablox 中创建新应用程序之后, 必须在 WebSphere 中注册该应用程序, 并且该应用程序将在应用程序服务器启动时启动)。

当与商用应用程序服务器 (如 WebSphere 或 WebLogic) 配合运行时, 您可以访问该平台上的所有工具、可伸缩性和服务以及 DB2 Alphablox 平台提供的所有服务。

安装提供了用于配置 WebSphere、WebLogic 和 Tomcat 的选项。要了解详细信息, 请参阅《安装指南》。

因为 DB2 Alphablox 应用程序是 J2EE 应用程序, 所以应用程序能够在不同的配置中工作。因此, 可以在 Apache Tomcat 配置中开发和测试应用程序, 然后在企业应用程序

服务器配置中部署该应用程序。有时候，在不同平台上运行时会有一些不太重要的运行时差别，所以，在部署应用程序之前有必要在不同的配置中测试该应用程序，但只要应用程序未使用特定于平台的服务，从一个配置迁移到另一个配置就不会出现什么问题。

Apache Tomcat 配置

作为一个选项，DB2 Alphablox 可以使用 Apache Tomcat 应用程序服务器。在 Apache Tomcat 配置中，DB2 Alphablox 安装程序使用指定的 Apache Tomcat 服务器（仅限于 V3.2.4）。要了解有关 Apache Tomcat 的信息，请访问 <http://jakarta.apache.org/tomcat/>。要获取 Tomcat V3.2.4 的副本，请访问 <http://archive.apache.org/dist/tomcat/tomcat-3/archive/v3.2.4/>。

当将 Apache Tomcat 与 DB2 Alphablox 配合使用时，不需要执行特定于 Tomcat 的管理任务就可以使用 DB2 Alphablox 平台。Apache Tomcat 配置是一个可伸缩的解决方案，它能够使用第 105 页的第 16 章，『使用数据库存储库』中描述的集群解决方案。

Apache Tomcat 配置包含具有用户、组和角色支持的全面安全性。可以在 Apache Tomcat 配置中使用任何受支持的外部 Web 服务器（例如 Microsoft® IIS、Sun iPlanet 或 Apache Web Server），也可以让 Tomcat 处理 HTTP 请求。

技巧：Apache Tomcat 配置中包括的 HTTP 服务器非常适合于开发系统或小型的生产系统，但对于较大型的生产系统，您可能会想使用外部 Web 服务器，因为外部 Web 服务器具有许多高速缓存和页面服务功能，从而能够提高 HTTP 的性能。

DB2 Alphablox 平台的组件

DB2 Alphablox 平台由下列元素组成，本节对它们进行讨论：

- 第 3 页的『DB2 Alphablox』
- 第 6 页的『DB2 Alphablox 应用程序』
- 第 7 页的『Application Studio』

DB2 Alphablox

DB2 Alphablox 具有健壮的体系结构，该体系结构是专门为快速开发、部署和使用分析应用程序而设计的。该体系结构提供的关键功能包括：

- 『数据适配器』
- 第 4 页的『Blox 组件』
- 第 5 页的『DHTML 客户机』
- 第 6 页的『远程管理能力』
- 第 6 页的『集中的应用程序管理』
- 第 6 页的『N 层体系结构』

数据适配器

DB2 Alphablox 包括专门为了连接至各种数据库而设计的数据管理器。数据管理器负责访问、浏览、查询和检索关系数据库及多维数据库（包括 DB2 Alphablox 立方体）中的数据。对每个数据库的连接是通过使用插件适配器来实现的。对于关系数据库，插件

适配器通常是 JDBC 驱动程序。每个适配器都封装了特定于数据库的连接信息和处理，这大大减少了连接至其他数据库所需的工作。

数据管理器及相关联的数据适配器提供了对下列各项的支持：

- 浏览预先配置且指定的数据库连接（称为数据源）的集合。
- 向 DB2 Alphablox 应用程序提供对每个数据源中的可用数据库的访问。
- 发布特定数据源的兼容查询类型。
- 允许遍历数据库的元数据。
- 为用户会话管理数据库连接。
- 将查询对象转换为底层本机查询语言。
- 对数据库执行查询。
- 检查结果集的数据和模式。
- 通过显示、旋转、扩展、排序和钻取来处理结果集。
- 创建获取用户输入并将数据写回到底层数据库中的应用程序（通常用于预算应用程序的“假设分析”方案）。

Blox 组件

DB2 Alphablox 应用程序使用构建块（称为 Blox 组件）来通过标准的 Web 浏览器访问和显示实时企业数据。Blox 组件是可复用的软件组件，它们在标准的 JSP 页面上组合（即组装）到一起并生成通过 Web 浏览器访问的交互式应用程序。这些交互式应用程序既可以在企业内部网中访问，也可以通过因特网来进行访问。

这些经过测试的软件组件为应用程序提供了下列领域的功能：

- 访问数据
- 对数据进行交互分析
- 灵活地表示数据
- 访问管理详细信息（例如，用户、组和应用程序名）

例如，DataBlox 组件可以使用底层数据库中的产品类别来填充 HTML 列表。接下来，用户从列表中进行的可以选择可以驱动对底层数据库执行的查询，因此用户在选择类别 *Compact Cars* 与选择 *SUVs* 时看到的数据集不相同。

Blox 组件具有很多可以使用 JSP 文件来访问的应用程序接口（API）调用。各种 API 调用允许对所访问的数据进行大量的定制、控制提供给用户的交互性的程度并且能够定制向用户提供的数据表示。

通过将 Blox 组件组装成基于 Web 的应用程序，开发者可以快速地为用户提供对企业信息的即时访问。对各种分析应用程序使用相同的 Blox 组件不仅仅减少了创建、交付和维护应用程序所需的工作，还缩短了用户的学习过程。无论显示什么数据，Blox 都具有跨应用程序和平台一致的行为。

此外，组装 DB2 Alphablox 应用程序所需掌握的技巧比进行传统应用程序开发所需掌握的技巧略低。应用程序开发者可以快速地开始使用 HTML、JavaScript、多媒体对象和 Blox 组件，而不是必须首先掌握编程语言的复杂特征。借助这些元素，应用程序开发者可以创建合适的并且在视觉上引人注目的用户界面来向用户团体提供强大的功能。

DB2 Alphablox 提供了下列可用来组装应用程序的 Blox 组件:

Blox 组件	用途
DataBlox	<ul style="list-style-type: none"> • 提供对受支持的多维数据库的访问权，并开发数据集的客户机表示法 • 提供对受支持的关系数据库的访问权 • 接受和执行查询请求 • 将查询结果集提供给负责表示数据的 Blox 组件
ChartBlox	<ul style="list-style-type: none"> • 显示多维数据的图形视图 • 允许用户处理具有各种各样图表格式（包括饼图、条形图和折线图）的数据 • 使用户能够按顺序对分层数据执行向下钻取或者旋转数据视图
DataLayoutBlox	<ul style="list-style-type: none"> • 显示可用数据维及其当前所在的轴的分组列表 • 使用户能够在页面轴、行轴、列轴和“其他”（未使用的）轴之间移动维
GridBlox	<ul style="list-style-type: none"> • 以高级网格格式显示多维数据或关系数据 • 使用户能够分析和处理多维数据 • 使用户能够按顺序对分层数据执行向下钻取或者旋转数据视图
PageBlox	<ul style="list-style-type: none"> • 显示位于页面轴上的维的下拉列表（从而对显示在 ChartBlox 和 GridBlox 中的数据进行过滤） • 使用户能够更改数据过滤所依据的维和 / 或成员
ToolbarBlox	<ul style="list-style-type: none"> • 显示供用户访问 Blox 功能的按钮，如： <ul style="list-style-type: none"> - 在图表显示与网格表示之间进行切换 - 选择图表类型 - 保存和检索应用程序视图 - 移动工具栏以使访问更为方便 - 在单独的窗口中打开 Blox - 旋转行和列 • 允许组装人员选择出现在工具栏上的按钮，从而定制用户对 Blox 功能的访问权。
PresentBlox	<p>将上述 6 个 Blox 组件（DataBlox、DataLayoutBlox、ChartBlox、GridBlox、PageBlox 和 ToolbarBlox）的功能组合到单个 Blox 中，这简化了应用程序组装并节省了 Web 页面空间。</p>
RepositoryBlox	<p>向应用程序组装人员提供对存储的对象（包括已保存的应用程序视图以及服务器属性、应用程序属性、组属性和用户属性）的访问权。</p>

要获取 DB2 Alphablox 中的可用 Blox API 的详细描述，请参阅 *Developer's Reference*。请参阅《开发者指南》以获取有关使用 Blox 组件来创建应用程序的指南。

DHTML 客户机

因为 DB2 Alphablox 使用的是标准 DHTML 技术，所以应用程序开发者可借助它来创建用户通过 Web 浏览器（而无需其他插件）即可访问的应用程序。DB2 Alphablox 按照请求应用程序的方式来显示该应用程序，而不要求应用程序开发者进行额外的编码工作。

DHTML 客户机基于动态 HTML 技术，它利用 JavaScript 和级联样式表（CSS）来支持全面的数据分析功能，并提供了高度可用并且可定制的图形用户界面。它不需要任何插件，也不要求下载 Java 类文件。DHTML 客户机需要使用受支持版本的 Mozilla Firefox 和 Microsoft Internet Explorer。

远程管理能力

DB2 Alphablox 通过基于 Web 的管理页面、控制台命令窗口或 Telnet 窗口提供了全面的系统管理能力。通过这些管理界面，管理员可以执行下列类型的任务：

- 创建 DB2 Alphablox 控制台日志并指定所要记录的事件和消息的级别。
- 监视 DB2 Alphablox 和应用程序活动。
- 创建 DB2 Alphablox 对象，如数据源、用户、组和应用程序。
- 监视 DB2 Alphablox 对象的使用。
- 启动和停止 DB2 Alphablox 服务、会话和应用程序。

要了解更多信息，请参阅第 21 页的第 4 章，『基本管理任务』和第 69 页的第 13 章，『配置 DB2 Alphablox 属性』。

集中的应用程序管理

DB2 Alphablox 提供了负责执行集中的应用程序管理工作的应用程序管理器。应用程序管理器维护并跟踪下列各项内容：

- 可用应用程序的列表。
- 活动应用程序的列表。
- 应用程序中的活动。
- 已登录到应用程序的用户。
- 应用程序使用的 Blox 组件。
- 应用程序使用的数据库。
- 应用程序实例历史记录。

N 层体系结构

DB2 Alphablox 全面利用最新的 Java 技术来实现基于 Web 的 N 层体系结构。典型的应用程序可以使用三个层：

- 数据位于数据库服务器（关系数据库服务器或多维数据库服务器）上，并且从数据库服务器中被检索
- 应用程序 Web 页面位于应用程序服务器上的 Web 应用程序中，并且通过 DB2 Alphablox 接受服务
- Web 页面显示和用户界面表示通过 Web 浏览器发生在客户机上

根据需求，Blox 组件通过内部网或通过因特网从 Web 服务器传递到 Web 浏览器。这种基于 Java 的体系结构可使您免于安装、配置和维护客户端应用软件。

DB2 Alphablox 应用程序

对于用户来说，DB2 Alphablox 应用程序作为可以浏览的 Web 页面集合出现，就象是其他 Web 站点一样。这些 Web 页面用作下列应用程序组件的容器：

- 用于增强用户界面的标准 HTML 标记和页面元素（如徽标、文本、图像、图标、视频剪辑、声音剪辑和动画）

- 提供所需的应用程序功能和用户界面所必需的 Blox 组件
- 用于扩展应用程序和 UI 逻辑的 JavaScript 或 Java scriptlet

例如，销售分析应用程序可能会提供销售区域的图像映射。用户单击区域以查看该区域的数据图表。用户界面允许用户更改图表格式、以交互式网格格式显示数据和执行多维操作（如钻取和旋转）。根据应用程序的不同，用户还可以显示 Blox 输出以便进行打印或者导出到其他应用程序（如电子表格）。要了解有关 DB2 Alphablox 应用程序的类型和组件的更多信息，请参阅第 11 页的第 2 章，『DB2 Alphablox 应用程序』。

Application Studio

Application Studio 是 DB2 Alphablox 的姊妹产品并且是 DB2 Alphablox 的部件，它进一步缩短了向用户提供完整的定制分析应用程序所需的时间并减少了所需的工作。

Application Studio 包括下列组件：

- DHTML 查询构建器，一个工作台工具，它可以用来开发和测试针对应用程序数据源的查询，并且可以访问若干样本数据源以便创建应用程序原型和进行测试
- Blox 样本程序 - DHTML，它包含一些工作代码示例，这些示例重点说明 DB2 Alphablox 的功能并帮助开发者了解 Blox API。

Application Studio 使您能够快速创建应用程序原型。Studio 附带提供了完整的文档，这些文档提供了逐个步骤的模板配置指示信息。要了解有关 Application Studio 及其组件的更多信息，请参阅 Application Studio 中的各个模板的联机帮助。

DB2 Alphablox 体系结构

本节讨论下列 DB2 Alphablox 组件：

- 第 8 页的『服务管理器』
- 第 8 页的『请求管理器』
- 第 8 页的『会话管理器』
- 第 8 页的『用户管理器』
- 第 9 页的『应用程序管理器』
- 第 9 页的『数据管理器』
- 第 9 页的『存储库管理器』
- 第 10 页的『文件管理器』
- 第 10 页的『控制台管理器』
- 第 10 页的『DB2 Alphablox Cube Manager』
- 第 10 页的『集群管理器』

DB2 Alphablox 由若干个独立运行的离散服务组成。另外，它允许进行全面的管理工作，包括：

- 对 DB2 Alphablox 资源（如应用程序、数据源、用户和组）的管理界面。
- 可以由 Web 浏览器、Telnet 或命令窗口访问的管理控制台。
- 启动、暂停、继续和停止服务、会话和应用程序的能力。
- 监视 DB2 Alphablox 和应用程序活动的的能力。
- 用于跟踪事件和消息的日志。

作为初始安装的一部分，DB2 Alphablox 的配置自动包含其操作所需的一组缺省属性。以后，系统管理员可以更改缺省 DB2 Alphablox 属性或添加属性。要了解详细信息，请参阅第 69 页的第 13 章，『配置 DB2 Alphablox 属性』。

服务管理器

服务管理器启动和控制其他服务（包括必需的第三方服务）并提供对那些服务的访问。服务管理器处理所有服务请求并帮助分发服务组件。

请求管理器

请求管理器控制应用程序页面和同级服务的所有请求。DB2 Alphablox 为每个连接创建一个对等项。对等项跟踪连接状态。请求管理器创建、监视和管理每个请求的线程、针对它的有效会话列表来验证会话标识并且创建适当的请求对象来处理每个请求。

会话管理器

会话管理器控制并监视用户与 DB2 Alphablox 应用程序的交互。当用户请求应用程序时，会话管理器首先将该请求传递给用户管理器以进行认证。在用户成功地通过认证之后，会话管理器将创建一个新会话。单个用户可以具有多个并发会话（多个浏览器实例）。会话管理器创建和管理会话对象并跟踪用户所访问的应用程序。

会话管理器还终止静止会话（首先保存每个应用程序的当前状态）并释放会话资源。如果用户在浏览器窗口的会话到期之后尝试在该浏览器窗口中工作，则将显示重新连接消息。如果用户（通过在 Internet Explorer 中单击浏览器的刷新按钮或者在 Mozilla Firefox 中单击浏览器的重新装入按钮）请求重新连接，则会话管理器将创建新会话并将应用程序复原到它在先前会话到期时所处的状态。

用户管理器

Alphablox 中的用户管理器控制 DB2 Alphablox 服务的所有用户并负责认证应用程序和管理用户、监视他们的资源、管理用户对数据和应用程序的访问、维护活动用户列表以及维护关于每个用户的信息（如用户连接至特定应用程序的时间和持续时间）。

在 DB2® Alphablox 8.4 和更早版本中，用户管理器是在名为可扩展用户管理器的个性化引擎基础上构建的。外部用户管理器基于 Servlet 2.2，它处理安全性并提供用于其他外部存储库的定制安全性的基于 LDAP 和基于存储库的用户管理功能以及 API。从 DB2 Alphablox 8.4.1 开始，DB2 Alphablox 就支持 Servlet 2.4 规范。建议不要使用可扩展用户管理器，已将它替换为基于 Servlet 2.4 的新个性化管理器。此个性化管理器提供了一些方法，当创建、装入或删除用户或组时，这些方法就会通知 DB2 Alphablox。还可以使用此个性化管理器来指定和修改用户或组的属性，或者修改组中包含的用户和组。现在，安全性是一个单独的模型。用户认证信息是在底层应用程序服务器中配置的，而 DB2 Alphablox 提供基于 Java 认证和授权服务（JAAS）的 API 来进行用户认证。

用户管理器了解并管理下列提供 DB2 Alphablox 安全方案的对象：

- “用户”是单个的最终用户或应用程序管理员。
- “组”是用户和其他组的集合，它为管理员提供了将多个用户作为单个单元进行管理的便利机制。“组”是分层的。
- “许可权”确定特定用户在角色中具有的访问权（不具有访问权、具有读访问权和具有读/写访问权）。

- “角色”是具有一组特定许可权的用户和 / 或组的列表。

注：注意关于许可权和角色的下列事实：如果一个用户在同一个人物中出现多次（可能是既作为单个用户出现又作为组成员出现），则该用户将被授予所有条目的所有许可权的并集。例如，如果该用户具有读 / 写访问权，并且该用户的组具有读访问权，则该用户将被授予读 / 写访问权。

应用程序管理器

当组装人员或管理员从 DB2 Alphablox 主页上的**管理**选项卡的应用程序页面中创建或修改应用程序时，应用程序管理器将创建应用程序定义。

应用程序管理器还接受用户对应用程序的请求、验证用户是否具有应用程序访问权并为每个请求创建应用程序实例。应用程序管理器监视所有应用程序实例、使管理员能够关闭实例、自动地终止休眠实例并维护下列信息：

- 可用应用程序的列表
- 活动应用程序的列表
- 应用程序中的活动
- 已登录到应用程序的用户
- 应用程序使用的 Blox 组件

数据管理器

数据管理器控制对应用程序所使用的每个数据源的访问。数据管理器允许对各种关系数据源和多维数据源进行浏览、查询和检索数据，《安装指南》指定了此发行版所支持的数据源。

确切地说，数据管理器使用数据库适配器来：

- 浏览预先配置且指定的数据源的集合
- 显示每个数据源（如果有的话）中的可用立方体
- 发布特定数据源的兼容查询类型
- 允许遍历数据库的元数据
- 为用户会话管理数据库连接
- 将查询对象转换为底层本机查询语言
- 对数据库执行查询
- 检查结果集的数据和模式
- 通过旋转、扩展和钻取来修改结果集
- 创建将用户输入写回到数据库中以进行假设分析的应用程序

存储库管理器

存储库管理器控制 DB2 Alphablox 存储库及其内容，后者包括已保存的应用程序视图以及所有其他应用程序、组和用户属性。可以将 DB2 Alphablox 存储库存储在操作系统文件或关系数据库中。要了解详细信息，请参阅第 105 页的第 16 章，『使用数据库存储库』。

文件管理器

文件管理器控制临时文件以及 DB2 Alphablox 管理页面所使用的文件。

控制台管理器

控制台管理器为系统管理员提供了多个用于监视和控制 DB2 Alphablox 环境的界面（Web 浏览器、Telnet 和命令窗口）。要了解更多信息，请参阅第 123 页的第 19 章，『DB2 Alphablox 控制台命令』和第 16 页的『控制台』。

DB2 Alphablox Cube Manager

DB2 Alphablox Cube Manager 提供了用于停止和启动 DB2 Alphablox 立方体以及对其执行其他管理任务的界面。要了解有关 DB2 Alphablox Cube Server 的信息，请参阅《DB2 Alphablox Cube Server 管理员指南》。

集群管理器

当 DB2 Alphablox 在集群环境中运行时，集群管理器控制着它的不同节点之间的通信。集群配置用来提高 DB2 Alphablox 系统的可伸缩性以支持有很多用户时的情况。要了解有关将 DB2 Alphablox 设置为在集群环境中运行的详细信息，请参阅第 119 页的第 18 章，『使用集群环境』。

DB2 Alphablox 入门任务列表

下面是一个简化的列表，此列表显示了 DB2 Alphablox 入门所需执行的基本任务。要了解有关每个任务的更多详细信息，请参阅交叉引用列中列出的书籍或章节。

任务	交叉引用
1. 了解 DB2 Alphablox 的运行环境并确定配置详细信息（例如，IBM WebSphere、BEA WebLogic 还是 Apache Tomcat）。	第 1 页的第 1 章，『DB2 Alphablox 概述』
2. 安装和配置 DB2 Alphablox。	安装指南
3. 熟悉 DB2 Alphablox 主页，在此主页中，您可以执行管理任务，如创建应用程序、创建用户和设置安全策略。	第 15 页的第 3 章，『DB2 Alphablox 主页』
4. 了解数据所在的数据库环境。	您的数据库环境和信息工作者
5. 了解用于构建 DB2 Alphablox 应用程序的开发环境。	第 11 页的第 2 章，『DB2 Alphablox 应用程序』和《开发者指南》。
6. 创建应用程序定义。	第 27 页的第 6 章，『应用程序定义』
7. 使用 JSP 技术、Blox 定制标记库和 DB2 Alphablox API 来开发应用程序并进行测试。	请参阅《开发者指南》和 <i>Developer's Reference</i> 。
8. 将应用程序部署到企业。	您的用户、企业内部网和因特网。

第 2 章 DB2 Alphablox 应用程序

无论是处于企业内部网环境还是处于范围更广的因特网环境中，您都可以创建 DB2 Alphablox 应用程序来提供对数据的访问以便通过浏览器进行分析。由于 DB2 Alphablox 应用程序可高度定制并且可以集成到许多类型的企业中，因此，有许多种应用程序可以称为“DB2 Alphablox 应用程序”。本章简要描述那些类型的应用程序并描述构成 DB2 Alphablox 应用程序的组件。要了解 DB2 Alphablox 应用程序的更多信息，请参阅《开发者指南》。

DB2 Alphablox 应用程序的类型

DB2 Alphablox 应用程序是在 DB2 Alphablox 环境中运行的与 J2EE 相符的 Web 应用程序。由于 DB2 Alphablox 应用程序在 Web 浏览器中运行，并且实际上可以与任何其他 Web 应用程序技术组合，所以这些应用程序的作用以及外观的变化非常大。DB2 Alphablox 应用程序最基本的功能是提供一种在 Web 页面中以交互方式显示分析数据的方法，因此，用户可以使用该数据来获取有关基于该数据的业务的有用信息。该数据位于企业的各种数据库中。

对于这些 Web 应用程序中的 DB2 Alphablox 组件，本节简要描述下列类别的 DB2 Alphablox 应用程序：

- 『数据表示方式』
- 第 12 页的『关系报告应用程序』

还可以在单个应用程序中组合若干个不同的 DB2 Alphablox 元素。例如，可以在同一个应用程序中显示图表和电子表格。要了解这些应用程序的可能功能以及如何开发这些应用程序的更多详细信息，请参阅《开发者指南》和 *Relational Reporting Developer's Guide*。

数据表示方式

在 DB2 Alphablox 应用程序中，可以通过许多种方法来显示数据。可以在网格和 / 或图表中显示数据。还可以使用数据库中的数据来填充应用程序的元素，如下拉列表和菜单等等。

DB2 Alphablox DHTML 客户机使用最新的动态 HTML 技术（包括 HTML、JavaScript 和级联样式表（CSS））来通过高度可用并且可定制的图形用户界面（包括右键单击上下文菜单）支持全面的数据分析功能。通过将 Blox API 与 Blox UI 模型配合使用来实现可扩展性是 DHTML 客户机的主要优点。DHTML 客户机方式只能与较新的 Microsoft Internet Explorer 浏览器配合使用，但它不需要其他的 Web 浏览器插件或客户机安装，也不要求下载 Java 类文件。

转换为 PDF

您可以创建允许将报告打印至 PDF 文件（从而创建非常高质量的输出）的 DB2 Alphablox 应用程序。“打印至 PDF”功能使用一个 PDF 模板（该模板可以包含您需要的任何元素（例如，公司徽标和样板文件文本等等）），然后获取实时数据并创建

包含具有顶级质量的报告的 PDF 文件。然后，可以打印该 PDF 文件、通过电子邮件发送该 PDF 文件或者在 Web 站点上公布该 PDF 文件以进行分发。PDF 功能可以与所有客户机显示方式配合使用。

使用 XML 来定制绘制方式

DB2 Alphablox 提供了用于访问 XML 格式的数据的 API，这允许开发者以任何所需的方式来创建定制的数据布局或者使用您所需的任何定制技术来处理数据。如果您有一些无法通过其他表示方法满足的特殊需求，这就特别有用。

关系报告应用程序

可以使用 ReportBlox 组件来创建提供来源于关系数据库的报告的 DB2 Alphablox 应用程序。这些报告可高度定制并显示在 HTML 表中，并且使用级联样式表 (CSS) 技术来提供简易的定制和灵活的外观。

使用 ReportBlox 组件创建的报告可以在交互方式下使用，其目的是提供一个用户界面来定制这些报告以及提供全面的 API 来使用程序更改这些报告。要了解 ReportBlox 组件的详细信息，请参阅 *Relational Reporting Developer's Guide*。

DB2 Alphablox 应用程序的组件

由于 DB2 Alphablox 应用程序是 J2EE 应用程序，所以它们可以具有任何可供 J2EE 应用程序使用的组件。本节描述 DB2 Alphablox 应用程序的下列基本组件：

- 『JavaServer Pages (JSP) 』
- 『Blox 组件』

要获取 DB2 Alphablox 应用程序的目录结构的描述，请参阅第 27 页的『DB2 Alphablox 应用程序定义』。

JavaServer Pages (JSP)

JavaServer Pages (JSP) 技术是一种机制，应用程序开发者通常通过这种机制来向应用程序提供 DB2 Alphablox 功能。交互式 Web 页面是使用特定应用程序所需的 JSP、Blox 定制标记库、HTML 以及任何 Java 或 JavaScript 代码开发的。在运行时，JSP 页面在应用程序服务器上被编译（在大多数应用程序服务器配置中，为了提高性能，应用程序服务器仅在页面第一次被访问时才编译该页面；后续的页面请求将使用预编译的页面），接着，DB2 Alphablox 处理任何 DB2 Alphablox 请求，然后，Web 服务器将动态内容返回给用户。

Blox 组件

Blox 组件是 DB2 Alphablox 组件，它们提供诸如访问数据源以及在网格和图表中显示数据等功能。您使用 Blox 定制标记库来将 Blox 组件添加至 JSP 页面。Blox 标记库允许您控制 Blox 的许多方面。例如，您可以更改 ChartBlox 的属性以指定所显示的图表的类型。

您还可以使用 Java 或 JavaScript 来通过程序处理 Blox 组件。当您使用 JavaScript 来访问 Blox API 时，代码是在浏览器上解释的。当您使用 Java 来访问 Blox API 时，代码由 JSP 引擎（应用程序服务器）编译，然后在服务器上执行。

要了解使用 Blox 的对象类型的更多信息，请参阅第 5 页的『DHTML 客户机』。要了解更多有关在 DB2 Alphablox 应用程序中使用 Blox 组件以及有关使用 DHTML 客户机时可用的组件类型的详细信息，请参阅《开发者指南》和 *Developer's Reference*。

第 3 章 DB2 Alphablox 主页

本章描述 DB2 Alphablox 主页。要了解有关命令行界面的信息，请参阅第 123 页的第 19 章，『DB2 Alphablox 控制台命令』。

DB2 Alphablox 主页概述

在初始安装期间，将自动地配置 DB2 Alphablox。在安装之后，管理员可以设置或更改 DB2 Alphablox 启动配置、属性和端口。要管理 DB2 Alphablox，请使用 DB2 Alphablox 主页上的**管理**选项卡，或者使用控制台命令行界面。要访问 DB2 Alphablox 管理页面，请在浏览器窗口中输入下列其中一个 URL 地址：

```
http://<servername>/AlphabloxAdmin/home/  
http://<servername>/AlphabloxAdmin/  
http://<servername>/AlphabloxAdmin (iPlanet 除外)
```

其中 <servername> 表示服务器的名称以及运行 DB2 Alphablox 时使用的端口号。

“应用程序”选项卡

当您打开 DB2 Alphablox 主页时，缺省情况下将选择“应用程序”选项卡。

缺省情况下，当用户访问 DB2 Alphablox 主页时，将显示**应用程序**选项卡。**应用程序**选项卡列出用户能够访问的应用程序以及任何已保存的可用应用程序状态。某些用户可能由于不具有查看或访问其他选项卡的必需特权而只看到**应用程序**选项卡。

要启动应用程序，请单击期望的链接。缺省情况下，应用程序以 DHTML 方式启动。从下拉列表中选择**缺省应用程序状态**将以应用程序定义中定义的缺省状态装入应用程序。

其他应用程序

DB2 Alphablox 中定义的任何其他应用程序也会显示在**应用程序**选项卡下面。要启动应用程序，请单击它的链接。缺省情况下，应用程序以 DHTML 方式启动。从下拉列表中选择**缺省应用程序状态**将以应用程序定义中定义的缺省状态装入应用程序。

“管理”选项卡

仅当经过认证的用户具有管理特权时，在 DB2 Alphablox 主页上才会显示**管理**选项卡。单击此选项卡将打开**常规**页面，管理员可以在该页面中定义或修改 DB2 Alphablox 的各个方面。**管理**选项卡包含下列部分：

- 常规
- 第 17 页的『组』
- 第 17 页的『用户』
- 第 17 页的『角色』
- 第 17 页的『应用程序』
- 第 17 页的『数据源』

- 第 18 页的『DB2 Alphablox 立方体』

常规

管理选项卡下方的常规链接为管理员提供用于配置 DB2 Alphablox 核心功能的界面。

一般属性

您在一般属性部分中配置用于启动 DB2 Alphablox 和设置 DB2 Alphablox Cube Server 所允许的最大 DB2 Alphablox 立方体数目的参数。要了解有关执行这些任务的详细信息，请参阅第 69 页的『DB2 Alphablox 管理任务』。要了解有关 DB2 Alphablox Cube Server 的信息，请参阅《DB2 Alphablox Cube Server 管理员指南》。

定制属性

要了解有关定义定制属性的详细信息，请参阅第 73 页的『定制属性定义』。

运行时管理

使用运行时管理部分来管理 DB2 Alphablox 许可证、应用程序会话、注释集合、PDF 报告（仅限于 DHTML）和 DB2 Alphablox 立方体。要了解有关创建和维护 DB2 Alphablox 立方体的信息，请参阅《DB2 Alphablox Cube Server 管理员指南》。

控制台

单击启动控制台会话链接将打开控制台窗口，在该窗口中，管理员可以输入文本命令以执行大部分通常通过用户界面执行的管理功能。要了解有关使用控制台的信息，请参阅第 123 页的第 19 章，『DB2 Alphablox 控制台命令』。

注： Telnet 控制台用户（通过 DB2 Alphablox 主页 > 管理选项卡 > 常规链接 > **Telnet 控制台** 链接定义此用户）还可以通过任何 Telnet 终端软件来访问控制台。在大多数运行 Microsoft Windows® 操作系统的系统上，可以从 Windows 的开始菜单 > 所有程序文件夹 > 附件组 > **Telnet** 快捷方式访问 Telnet 软件，也可以从开始菜单中选择运行快捷方式并接着输入 **telnet** 来打开 Telnet 窗口。

Portal 主题实用程序

注： 仅当 Websphere Portal Server 上安装了 DB2 Alphablox 时，菜单中才会出现“Portal 主题实用程序”选项。

单击 **Portal 主题实用程序** 链接以创建要在包含 DB2 Alphablox Blox 组件的 WebSphere Portal 应用程序中使用的 Portal 主题。使用此实用程序，将通过合并指定的 WebSphere Portal Server 主题中的样式属性和指定的 DB2 Alphablox 主题中的样式属性来创建新的 DB2 Alphablox Portal 主题。新主题可用于优化 Blox 组件（例如，GridBlox 或 ChartBlox）在 Portal 应用程序中的显示。

提供了可选择使用的 portlet 版本的 Portal 主题实用程序，可在 Portal 应用程序中使用。可在 DB2 Alphablox 安装的 installableApps 目录中的 AlphabloxAdminPortlets.war 文件中找到此 portlet。

要获取使用 Portal 主题实用程序的帮助，请按此实用程序中的“帮助”按钮，查看可用的联机帮助。在 portlet 版本的 Portal 主题实用程序中，选择帮助方式将打开联机帮助。

应用程序迁移

注： 仅当 Apache Tomcat 服务器上安装了 DB2 Alphablox 时，菜单中才会出现“应用程序迁移实用程序”选项。

单击[应用程序迁移](#)链接以访问 Tomcat 5.5 的 Alphablox 应用程序迁移实用程序。此实用程序可帮助您将所选 DB2 Alphablox 应用程序从 Apache Tomcat 3.2.4 迁移至 Apache 5.5。对于迁移的每个应用程序，此实用程序将：

- 在需要时把应用程序复制到当前 Alphablox 安装的 *webapps* 目录中。
- 将应用程序部署至 Apache Tomcat 5.5
- 在需要时把应用程序导入到 Alphablox
- 更新当前发行版的 Alphablox 标记库描述符 (TLD) 文件
- 将应用程序的 *web.xml* 文件更新为 Servlet 2.4 规范

已迁移的应用程序应该在带有 DB2 Alphablox 的 Apache Tomcat 5.5 安装上正常运行。

组

使用[管理](#)选项卡下方的[组](#)链接来定义组以及将用户指定到组等等。要了解有关设置组的详细信息，请参阅第 49 页的第 9 章，『组定义』。

用户

使用[管理](#)选项卡下方的[用户](#)链接来定义用户、重设密码以及对用户指定组成员资格等等。要了解有关设置用户的详细信息，请参阅第 45 页的第 8 章，『用户定义』。

角色

使用[管理](#)选项卡下方的[角色](#)链接来创建和修改角色。角色允许您控制对特定应用程序具有访问权的用户和 / 或组。要了解有关设置角色的详细信息，请参阅第 53 页的第 10 章，『角色定义』。

应用程序

使用[管理](#)选项卡下方的[应用程序](#)链接来创建位于 DB2 Alphablox 上的应用程序以及修改其状态。您可以定义缺省状态、设置角色以及定义显示在“应用程序”选项卡上的图像。要了解有关定义应用程序时所需完成的任务的详细信息，请参阅第 27 页的第 6 章，『应用程序定义』。

数据源

使用[管理](#)选项卡下方的[数据源](#)链接来定义和修改可供 DB2 Alphablox 上运行的 DB2 Alphablox 应用程序使用的数据源。您可以定义数据源以访问 OLAP 数据库（例如，IBM DB2 OLAP Server™、Hyperion Essbase 和 Microsoft Analysis Services）、关系数据库（例如，IBM DB2 UDB、Oracle、Sybase 和 Microsoft SQL Server）以及 DB2 Alphablox 立方体。要了解有关定义和修改数据源的详细信息，请参阅第 37 页的第 7 章，『数据源定义』。

DB2 Alphablox 立方体

使用**管理**选项卡下方的**立方体**链接来定义或修改 DB2 Alphablox 立方体。DB2 Alphablox 立方体展示以多维格式存储在关系数据库中的数据。要了解有关 DB2 Alphablox 立方体和 DB2 Alphablox Cube Server 的更多信息，请参阅《DB2 Alphablox Cube Server 管理员指南》。

“组装”选项卡: Application Studio

单击 DB2 Alphablox 主页上的**组装**选项卡将打开 DB2 Alphablox Application Studio。打开的 Application Studio 页面带有指向包含示例和工具的页面的链接。这些页面被设计成帮助开发者快速地创建 DB2 Alphablox 应用程序。Application Studio 是为了方便您创建应用程序而提供的易于使用的工作台工具和工作代码示例的框架。要了解有关 Application Studio 的更详细的信息，请参阅《开发者指南》。

工作台

组装选项卡下方的**工作台**页面包含指向**查询构建器**的链接。查询构建器是一个应用程序，它允许开发者交互地输入对各种数据源的查询以及在网格和 / 或图表中查看结果。

示例

组装选项卡下方的**示例**页面包含一些链接，那些链接指向示例 JSP 应用程序、代码样本和其他示例。示例包括 *Blox* 样本程序，后者包含许多工作代码示例。这些代码样本对您了解如何使用 DB2 Alphablox 来开发应用程序能够起到极大的帮助作用，并且它们中的许多都适合于被剪切和直接粘贴到您的定制应用程序中。

样本数据

DB2 Alphablox 附带交付了供 *Blox* 样本程序使用的样本数据。样本数据是一个名为 Quality Chocolate Company (QCC) 的数据库，并且为多种数据库（既包括多维数据库也包括关系数据库）提供了各种数据库版本。要了解有关安装样本数据的详细信息，请参阅《安装指南》中的『装入样本数据』。

右上角的链接

DB2 Alphablox 主页的每个选项卡的页面右上角都有两个链接:

- 『我的概要文件』
- 『帮助』

我的概要文件

单击**我的概要文件**链接将打开**概要文件**页面。在此页面中，用户可以更改他们的概要文件描述、密码和电子邮件地址，并可以安装或卸载本地类 (JAR) 文件。

帮助

帮助菜单提供了指向下列各项的链接:

- **帮助**链接提供按钮被单击时所在页面的上下文相关联机帮助。联机帮助提供了有关如何使用每个页面的界面的信息。

- **联机文档**链接在新的浏览器窗口中打开 DB2 Alphablox 联机文档。联机文档包括 DB2 Alphablox 文档集的基于 Web (HTML) 的版本和 PDF 版本。
- **服务器端 API 参考**链接打开服务器端 Java API 的 Javadoc。
- **关于 DB2 Alphablox** 链接提供关于 DB2 Alphablox 的版权信息。

注: 非隶属于 AlphabloxAdministrator 角色的用户不会看到所有这些链接。

第 4 章 基本管理任务

DB2 Alphablox 包括全面的服务器环境管理设施。DB2 Alphablox 主页提供了用于管理服务器的所有各个方面的单一出发点。本节描述用于启动、访问、配置和停止 DB2 Alphablox 的步骤。

启动 DB2 Alphablox

在将 DB2 Alphablox 与诸如 IBM WebSphere 或 BEA WebLogic 之类的商用应用程序服务器一起运行时，DB2 Alphablox 将与应用程序服务器集成。在启动应用程序服务器时，DB2 Alphablox 将自动启动。对于 IBM WebSphere Portal Server，DB2 Alphablox 将在您启动 Portal Server 时启动。要了解启动 WebLogic 或 WebSphere 的信息，请参阅那些产品的文档。

访问 DB2 Alphablox

DB2 Alphablox 是通过 DB2 Alphablox 主页的**管理**选项卡管理的。

要通过浏览器访问 DB2 Alphablox，请输入以下 URL：

```
http://<serverName>/AlphabloxAdmin/home/
```

其中 <serverName> 表示服务器的名称以及运行 DB2 Alphablox 时使用的端口号。

要从 Windows 的开始菜单访问 DB2 Alphablox 主页的**管理**选项卡：

1. 打开 Windows 的**开始**菜单。
2. 选择**所有程序**文件夹。
3. 选择 **DB2 Alphablox** 程序组。
4. 选择 DB2 Alphablox 实例的实例名（缺省值是 **AlphabloxAnalytics**）。
5. 选择 **DB2 Alphablox** 主页快捷方式。

停止 DB2 Alphablox

在 WebSphere 应用程序服务器上，可以通过停止应用程序服务器来停止 DB2 Alphablox。对于 WebSphere Portal Server，停止 Portal Server 即停止 DB2 Alphablox。在 Apache Tomcat 安装上，如果 DB2 Alphablox 正在控制台窗口中运行，则可以使用关闭脚本来停止它，如果它正作为 Windows 服务运行，则从服务控制面板中停止它。

在 Apache Tomcat 安装上通过关闭脚本来停止 DB2 Alphablox

当 DB2 Alphablox 在“控制台”窗口（在 Windows 上：当从 Windows 的开始菜单 > 所有程序文件夹 > **DB2 Alphablox** > 组 > 实例名（缺省值是 **AlphabloxAnalytics**） > **关闭 DB2 Alphablox** 快捷方式启动 DB2 Alphablox 时打开的窗口；在 Sun Solaris 上：在其中启动服务器的窗口）中运行时，要停止 DB2 Alphablox，请执行下列步骤：

在 Windows 系统上:

- 运行开始菜单 > 所有程序文件夹 > **DB2 Alphablox** 组 > 实例名 (缺省值是 **AlphabloxAnalytics**) > 关闭 **DB2 Alphablox** 快捷方式。

在 Linux 和 UNIX 系统上:

- 从 `<db2alphablox_dir>/bin` 目录中运行以下命令, 其中 `<db2alphablox_dir>` 是 DB2 Alphablox 的安装目录:
`./StopAlphablox.sh`

在 Apache Tomcat 安装上从服务控制面板中停止 DB2 Alphablox

如果正在将 DB2 Alphablox 作为 Windows 服务运行, 则执行下列步骤来停止 DB2 Alphablox:

1. 从 Windows 的“开始”菜单 > 设置 > 控制面板快捷方式打开 Windows 控制面板。
2. 双击服务控制面板文档。“服务”窗口显示。
3. 对于每个正在运行的 DB2 Alphablox 服务, 选择该服务并单击**停止**按钮。
4. 按是按钮以确认要停止该服务。
5. 单击关闭按钮以关闭“服务”窗口。

使用 Essbase 客户机库实用程序

要将 DB2 Alphablox 与 DB2 OLAP Server 或 Hyperion Essbase 数据源配合使用, 必须安装合适的 Essbase 客户机库文件并使它们可用于 DB2 Alphablox。可从 Hyperion (<http://support.hyperion.com/>) 获取 Essbase 客户机库。

要更改或更新 DB2 Alphablox 上安装的 Essbase 客户机库:

1. 在 DB2 Alphablox 上, 打开以下目录:

`db2alphablox_dir/bin/`, 其中, `db2alphablox_dir` 是 DB2 Alphablox 的安装目录 (例如, `c:\alphablox\analytics\`)。

2. 找到 ChangeEssbase 批处理文件 (在 Windows 上是 `ChangeEssbase.bat` 文件, 在 Linux[®] 和 UNIX[®] 上是 `ChangeEssbase.sh` 文件), 然后运行它。会出现脚本控制台窗口。
3. 输入 DB2 Alphablox 安装目录的标准路径 (例如, `c:\alphablox\analytics\`), 然后按 **Enter** 键。
4. 将显示可用类库选项的列表。根据要使用的客户机库的 Essbase 版本的不同, 选择适当的选项, 然后按 **Enter** 键。
5. 如果操作成功, 则您将看到一条指示已重新配置系统的消息。
6. 应重新启动 DB2 Alphablox 以使更改生效。

其他管理任务和信息

要了解其他管理任务的信息，请参阅下列各节：

- 第 1 页的第 1 章，『DB2 Alphablox 概述』
- 第 123 页的第 19 章，『DB2 Alphablox 控制台命令』
- 第 15 页的第 3 章，『DB2 Alphablox 主页』
- 第 27 页的第 6 章，『应用程序定义』
- 第 73 页的『定制属性定义』
- 第 53 页的第 10 章，『角色定义』
- 第 37 页的第 7 章，『数据源定义』
- 第 45 页的第 8 章，『用户定义』
- 第 49 页的第 9 章，『组定义』

第 5 章 客户机管理

管理员需要了解一些关于将 DHTML 客户机与 DB2 Alphablox 应用程序配合使用的问题。

DHTML 客户机管理

当对 DB2 Alphablox 使用 DHTML 显示方式时，通过使用受支持的 Mozilla Firefox 和 Microsoft Internet Explorer Web 浏览器的内置能力来显示分析应用程序。因此，只需执行少量管理工作就可以支持基于 DHTML 的应用程序。

受支持的 DHTML 客户机配置

请参阅安装指南的『系统需求』一节，以了解有关 Mozilla Firefox 和 Microsoft Internet Explorer Web 浏览器的受支持 DHTML 客户机配置的信息。

DHTML 客户机问题

当用户将 Mozilla Firefox 和 Microsoft Internet Explorer 与 DB2 Alphablox 应用程序配合使用时，通常可能会遇到几个问题。以下是您可能想了解的一些潜在问题。

CSS

在极少数情况下，用户在 Mozilla Firefox 和 Microsoft Internet Explorer Web 浏览器的高级选项中设置级联样式表（CSS）的个人首选项。由于 CSS 具有级联性质并且 DHTML 客户机广泛使用了 CSS，因此这些设置的更改有可能会影响 Blox 组件在这些浏览器中的外观。

弹出窗口

在使用 DHTML 客户机时，当用户单击按钮或链接时，期望的弹出窗口将显示进一步的信息，但对话框也可能无法显示。Web 浏览器中的内置弹出窗口拦截功能可能会阻止弹出窗口的显示。另外，可选的浏览器工具（例如，Google 工具栏）和防火墙软件（例如，Zone Alarm）可能包含了某些设置，它们会影响 Web 浏览器弹出窗口的行为。要更正此问题，用户必须禁用此功能，或者通过更改他们的用户首选项来允许 DB2 Alphablox 应用程序弹出窗口。

第 6 章 应用程序定义

本章描述 J2EE 应用程序结构并提供用于在 DB2 Alphablox 中创建、修改和删除应用程序定义的过程。

DB2 Alphablox 应用程序定义

管理选项卡下的应用程序页面上显示了 DB2 Alphablox 应用程序列表。要定义新的 DB2 Alphablox 应用程序，请单击**创建**按钮。要修改或删除应用程序定义，请从列表中选择应用程序并按**编辑**或**删除**按钮。

当您使用 DB2 Alphablox 的管理页面创建应用程序时，DB2 Alphablox 将在 DB2 Alphablox 存储库中创建一个应用程序定义。DB2 Alphablox 存储库中的应用程序名由两部分组成：企业应用程序名和唯一标识（URI）。例如，如果创建了一个名为 MyApp 的应用程序，则该应用程序在存储库中将被命名为 MyApp_MyApp。对于 WebSphere 和 WebLogic 应用程序服务器，DB2 Alphablox 创建必须导入到应用程序服务器的可安装应用程序。

以下提供了导入到 WebSphere 应用程序服务器的典型 DB2 Alphablox 应用程序的目录结构：

```
<application>.ear
  META-INF/
    application.xml
    MANIFEST.MF
  <application>.war
    META-INF/
      MANIFEST.MF
    WEB-INF
      timeschema.dtd
      timeschema.xml
      web.xml
    classes/
    lib/ (for Alphablox 8.4.1 applications only)
      aastaglibs.jar
    tlds/ (for Alphablox 8.4 applications only)
      blox.tld
      bloxform.tld
      bloxlogic.tld
      bloxui.tld
```

应用程序文件（包括 JSP 文件、JavaScript 文件、HTML 文件、CSS 文件和图像文件）通常放置在 <application>.war 目录中。根据您的实际开发情况，某些文件可能会放置在 <application>.war 目录的嵌套文件夹中。定制 Java 类，包括 JavaBeans 组件放置在 classes 目录中。在 DB2 Alphablox 8.4 应用程序中，DB2 Alphablox 应用程序需要的标记库描述符（TLD）文件位于 tlds 目录中；在 DB2 Alphablox 8.4.1 应用程序中，TLD 文件打包在应用程序的 lib 目录中的 aastaglibs.jar 中。某些专用应用程序必需的与时间模式相关的文件（timeschema.dtd 和 timeschema.xml）直接位于 WEB-INF 目录中。在 DB2 Alphablox 创建可安装的应用程序时，部署描述符文件（web.xml）也会按照 DB2 Alphablox 应用程序需要的那样进行修改。

应用程序名

在 DB2 Alphablox 存储库中使用的应用程序名由企业应用程序名加上唯一标识组成。在创建应用程序后，可以转至特定应用程序的编辑视图以查看应用程序的命名信息。此列表包括 DB2 Alphablox 名称、J2EE 应用程序名、模块名、上下文路径和显示名。

WEB-INF 目录

每个 web 应用程序都包含 WEB-INF 目录。此目录是在应用程序创建期间自动创建的，它包含关于应用程序的元数据。在正常操作期间，您应该不需要修改 WEB-INF 目录下的任何内容。在创建应用程序期间，以下文件将安装在 WEB-INF 目录中：

- web.xml
- lib 目录中的 aastaglibs.jar（对于 DB2 Alphablox 8.4.1）；或各个 TLD 文件（对于 DB2 Alphablox 8.4），例如，*blox.tld*
- timeschema.dtd
- timeschema.xml

应用程序的部署描述符文件（web.xml）是所有 J2EE 应用程序中的标准 XML 文件，它包含了描述应用程序属性的标记。应用程序服务器在应用程序初始化期间读取 web.xml 文件。

要点：对 web.xml 文件的修改由 DB2 Alphablox 在产品升级期间自动执行；如果您必须修改 web.xml 文件，请务必小心谨慎，因为不正确的条目可能会导致应用程序中出现异常行为。

Alphablox 8.4.1 中打包在 *aastaglibs.jar* 中的 Alphablox 标记库描述符（TLD）文件（或 Alphablox 8.4 中的松散 TLD 文件）使开发者可以访问用于创建 Blox 组件和 DB2 Alphablox 应用程序的 DB2 Alphablox 标记库。向开发者提供了直接存储在 WEB-INF 目录中的两个与时间相关的文件（timeschema.dtd 和 timeschema.xml），它们用来创建专门处理财政和日历时间段的程序。要了解关于这些文件的更多信息，请参阅 *开发者指南* 和 *Developer's Reference*。

定义新的应用程序

执行下列操作以定义新的应用程序：

注：如果您要将 WebSphere 用作应用程序服务器，则需要执行其他步骤来创建应用程序。要了解详细信息，请参阅第 29 页的『在与 WebSphere 一起运行时定义应用程序』。

1. 作为 *admin* 用户或作为另一个隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。
3. 单击**应用程序**链接。
4. 单击**创建**按钮。将显示“**创建应用程序**”页面。
5. 输入应用程序的**唯一名称**。当在非英语语言环境中运行时，允许的字符是 A-Z、a-z、0-9、下划线以及特殊字符（例如，重音字符）。名称的显示是区分大小写的，但认证的实际名称不区分大小写。名称 *Public*、*Private* 和 *Properties* 已被保留，不能用作对象名。

6. [可选] 在**描述**字段中输入应用程序的简短描述。
7. [可选] 输入与应用程序名相对的 **URL** 地址，该 **URL** 标识了此应用程序所在的初始页面（**主 URL**）和另一个指向此应用程序的标识性图像（**图像 URL**）的相对 **URL**。这两个值用来在**应用程序**选项卡上创建一个条目，用户可以通过该条目来启动该应用程序。
8. 在**缺省保存状态**字段中，输入用于指定想要保存的状态的参数名。如果应用程序开发者希望他们的用户能访问应用程序处于保存状态的版本，则将创建参数名。如果未指定“缺省保存状态”，复原的状态将是浏览器关闭或超时时应用程序所处的状态。
9. 对于**写许可权安全角色**文本框，输入可以访问此应用程序的现有角色的名称。如果保留此文本框为空白，则将应用缺省访问权。
10. 输入**会话超时**以设置 J2EE 会话超时，这是用户会话超时之前的空闲时间。超时值以分钟计。
11. 从已启用“**复原已保存的应用程序状态**”下拉列表中选择**是**或**否**以指示是否装入最近自动保存的状态。
12. **包括在应用程序页中**选项确定用户是否将在“应用程序”页视图中看到此应用程序。缺省值为**是**。
13. [可选] 设置应用程序的**缺省绘制方式**。缺省情况下，新应用程序支持 **DHTML** 绘制方式，并将 **Alphablox** 标记库描述符（**TLD**）文件的副本安装到新应用程序的目录中。
14. [可选] 使用**头链接**功能来设置一些链接，这些链接起源于网格中显示的成员，它们用来打开 **URL** 或执行某些 **JavaScript**。在**头链接**文本框中添加唯一的成员名和 **URL** 地址或其他有效 **URL** 文本，从而在具有以下语法的每个条目之间插入换行符：
Member = URL

例如，要创建从名为 **IBM** 的成员指向 **http://www.ibm.com** 的链接，请将以下内容输入到“头链接”文本框中：
IBM = http://www.ibm.com
15. 按**保存**按钮以定义新的应用程序并返回到“应用程序”页。
16. 如果正在将 **Apache Tomcat** 配置与外部 **Web** 服务器（**Apache HTTP Server**、**Microsoft IIS** 或 **Sun iPlanet**）一起运行，则必须先执行其他步骤，然后应用程序才可用。要了解详细信息，请参阅第 33 页的『向外部 **Web** 服务器注册应用程序』。

在与 WebSphere 一起运行时定义应用程序

如果您正在使用 **IBM WebSphere Application Server**，则必须在 **DB2 Alphablox** 中定义应用程序之前或之后在 **WebSphere** 中定义该应用程序。本节描述在 **WebSphere** 中创建 **Web** 应用程序时所需执行的基本步骤。要了解有关使用 **WebSphere** 来创建应用程序的详细信息，请参阅 **WebSphere** 文档。

当与 **WebSphere** 一起运行时，可以通过两种方法来定义 **DB2 Alphablox** 应用程序：

- 第 30 页的『在 **DB2 Alphablox** 创建一个应用程序，然后在 **WebSphere** 中注册』
- 第 31 页的『将现有的 **WebSphere** 应用程序导入到 **DB2 Alphablox** 中』

这两种方法都是可接受的；请选择您使用起来最方便的方法。

在 DB2 Alphablox 创建一个应用程序，然后在 WebSphere 中注册

当与 WebSphere 一起运行时，如果您在 DB2 Alphablox 中创建了一个应用程序，则 DB2 Alphablox 将在一个企业归档（EAR）文件中创建应用程序结构。该 EAR 文件是在以下目录中创建的：

```
<db2alphablox_dir>/installableApps/
```

该文件的名称是 <application>.ear，其中 <application> 是创建该应用程序时对其指定的名称。

在可使用新创建的应用程序之前，必须执行下列步骤以在 DB2 Alphablox 中创建应用程序结构，然后用 WebSphere 对其进行注册：

1. 在 DB2 Alphablox 中，首先定义一个新的应用程序，如第 28 页的『定义新的应用程序』所述。将在 <db2alphablox_dir>/installableApps 目录中创建一个 EAR 文件。
2. 打开 WebSphere 管理控制台，然后选择 **应用程序 > 安装新应用程序**。
3. 在**为安装应用程序做准备**屏幕上，单击“浏览”按钮并选择以下路径：

```
<db2alphablox_dir>/installableApps/<newApplication>.ear
```

其中，<application> 是要安装的应用程序的名称，然后按“下一步”。

注：仅限于集群环境：您可能需要使用服务器路径设置来浏览网络才能找得到 <newApplication>.ear 文件。

4. 下一个屏幕将显示**缺省绑定选项**。除非您需要使用不同的绑定，否则请保留缺省设置，然后单击“下一步”。
5. 将显示**应用程序安全性警告**屏幕。滚动到此页面的底部，然后单击“继续”按钮。
6. 下一节『安装新应用程序』由对话框中显示的下列 5 个步骤组成：

步骤 1: 提供选项以执行安装

您可以保留现有设置并单击“下一步”。

步骤 2: 为 Web 模块映射虚拟主机

接受现有的值并单击“下一步”。

步骤 3: 将模块映射至应用程序服务器

接受现有的值并单击“下一步”。

步骤 4: 将安全角色映射至用户/组

应该会显示 AlphabloxAdministrator 和 AlphabloxUser 这两个角色。您需要为每个角色至少添加一个用户。

对于 AlphabloxAdministrator 角色，选取该角色前面的复选框，然后按**查找用户**或**查询组**按钮以添加管理用户。必须选择至少一个用户。在添加了用户或组之后，它们应该会列示在此角色的“映射的用户”或“映射的组”值中。

对于 AlphabloxUser 角色，选取此角色“所有已认证”列下面的复选框。这允许所有已认证的用户访问该应用程序。根据应用程序需求的不同，还可以使用这些设置来将访问权限制到特定的用户或组。

完成后，单击“下一步”。

步骤 5: 总结

滚动到此屏幕的底部，然后按“完成”按钮。

7. 于是将安装应用程序，并且将显示“已成功安装了应用程序 <application>”。单击**保存至主配置**链接。
8. 应该会显示**保存至主配置**对话框。单击“保存”按钮。片刻之后，您将返回到管理控制台主页。
9. 在 WebSphere 管理控制台中，打开**应用程序 > 企业应用程序**并重新启动应用程序。

现在，您可以从 DB2 Alphablox 的主页中添加内容、使用该应用程序以及修改它的任何属性。

将现有的 WebSphere 应用程序导入到 DB2 Alphablox 中

要将现有的 WebSphere 应用程序导入到 DB2 Alphablox 中，从而添加必需的 DB2 Alphablox 设置和文件，请执行第 35 页的『导入现有的 J2EE 应用程序』中的步骤。

修改现有的应用程序定义

本节描述如何修改现有的应用程序。执行下列操作以修改现有的应用程序：

1. 作为 admin 用户或作为隶属于 administrators 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。
3. 单击**应用程序**链接。
4. 从应用程序列表中选择要更改的应用程序并单击**编辑**按钮。**编辑应用程序**页面打开。
5. 更改适当的属性值。

注： WebSphere 和 WebLogic 配置：可通过修改**名称**字段来重命名应用程序，但只能在 DB2 Alphablox 存储库中更改应用程序名，而不能修改相关的 WebSphere 或 WebLogic 应用程序名或上下文路径。要修改 WebSphere 或 WebLogic 中相关联的应用程序属性，请使用应用程序服务器管理控制台来将底层应用程序上下文路径重命名以使用新名称，并将底层应用程序文档基本目录重命名。

注： Apache Tomcat 配置：可通过修改**名称**字段来重命名应用程序。当您重命名应用程序时，DB2 Alphablox 存储库中的应用程序目录名以及任何引用该应用程序的书签都将被重命名。

6. 单击**保存**按钮以更改应用程序的属性值并返回到**应用程序**页面中。
7. 如果正在将 Apache Tomcat 配置与外部 Web 服务器（Apache HTTP Server、Microsoft IIS 或 Sun iPlanet）一起运行，并且您重命名应用程序，则必须执行您的配置所需的其他步骤。要了解详细信息，请参阅第 33 页的『向外部 Web 服务器注册应用程序』。

删除现有的应用程序定义

执行下列操作以删除现有的应用程序:

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。
3. 单击**应用程序**链接。
4. 从应用程序列表中选择要删除的应用程序并单击**删除**按钮。

要点: 删除应用程序时, 将从 DB2 Alphablox 中除去应用程序定义, 因此, 应用程序名将不再出现在应用程序列表中。但是, 不会除去应用程序文件夹及其关联文件, 必须手工删除它们。

要点: 在 WebLogic 上删除 DB2 Alphablox 应用程序: 当启用了自动部署时, 要在 WebLogic 服务器上完全地删除某个 DB2 Alphablox 应用程序, 请执行下列步骤:

- a. 使用 WebLogic 管理控制台来禁用自动部署功能。
- b. 从 DB2 Alphablox 中删除应用程序定义。DB2 Alphablox 将在 WebLogic 服务器上注销该应用程序。
- c. 删除应用程序文件夹。
- d. 如果您愿意的话, 再次启用自动部署功能。

如果未启用自动部署, 则只需执行步骤 b 和 c。

5. 如果正在将 Apache Tomcat 配置与外部 Web 服务器 (Apache HTTP Server、Microsoft IIS 或 Sun iPlanet) 一起运行, 则在删除应用程序之后, 必须执行您的配置所需的步骤。要了解详细信息, 请参阅第 33 页的『向外部 Web 服务器注册应用程序』。

当使用 WebLogic 集群时定义应用程序

如果正在 BEA WebLogic 集群环境中使用 DB2 Alphablox, 则执行下列步骤来创建新的应用程序:

1. 在 DB2 Alphablox 中, 从“管理”选项卡的应用程序下定义应用程序, 如第 28 页的『定义新的应用程序』所述。
2. 将新创建的应用程序的整个目录复制到 WebLogic 管理服务器。域下方与应用程序目录处于同一级别的目录是一个理想的位置。
3. 将所有必需的 JSP 文件添加到与 *WEB-INF* 目录处于同一级别的应用程序目录中。
4. 从 WebLogic 控制台中, 配置新添加的 Web 应用程序。在 WebLogic 服务器上, 您应该能够查看新的应用程序目录, 然后按**选择**或者浏览到正确的级别。
5. 选择集群作为部署目标, 然后按**配置并部署**按钮。
6. 在部署了应用程序之后, 选择 AlphabloxServer 作为 Web 应用程序。
7. 选择“部署”选项卡并重新部署 AlphabloxServer 应用程序。
8. 重新部署操作将停止并重新启动 DB2 Alphablox。
9. 当 DB2 Alphablox 重新启动时, 您就可以访问应用程序了。

注：您不需要将 JSP 文件复制到每台受管服务器，这是因为在 WebLogic 上注册 Web 应用程序时已经完成了此操作。要了解这种情况以及有关如何修改这种情况的更多信息，请参阅 WebLogic 文档。

向外部 Web 服务器注册应用程序

如果正在将 DB2 Alphablox 的 Apache Tomcat 配置与外部 Web 服务器一起运行，则每次创建或删除应用程序时都需要执行一些步骤。

由于 Web 服务器被设计成能够高效地提供静态页面服务，而应用程序服务器被设计成提供动态页面服务，所以，为了确保 Web 服务器和应用程序服务器都具有最佳的性能，这些步骤是必需的。如果不执行这些步骤，应用程序服务器就会被迫毫无必要地处理静态页面。

描述了下列配置：

- 第 33 页的『在 Apache HTTP Server 上注册应用程序』
- 第 34 页的『在 Microsoft Internet Information Server (IIS) Web 服务器上注册应用程序』
- 第 34 页的『在 Alphablox 8.4 上的 Sun iPlanet Web 服务器上注册应用程序』

在 Apache HTTP Server 上注册应用程序

Apache Tomcat 5.5 上的 DB2 Alphablox 8.4.1

使用 Apache Tomcat 配置创建或删除 DB2 Alphablox 应用程序定义时，只有在正确停止并重新启动 Apache HTTP Server 和 Apache Tomcat 这两个服务器后，Apache HTTP Server 上才会反映出所作的更改。重新启动 DB2 Alphablox 时，它会将新的 Apache 配置文件 (mod_jk.conf) 写入 Apache HTTP Server。

要注册 DB2 Alphablox 应用程序添加或删除操作：

1. 关闭 Apache HTTP Server 和 Apache Tomcat 服务器（运行 DB2 Alphablox）
2. 重新启动 Apache Tomcat 服务器。
3. 重新启动 Apache HTTP Server。

在重新启动这两个服务器后，Apache HTTP Server 上将识别应用程序更改。

Apache Tomcat 3.2.4 上的 DB2 Alphablox 8.4

如果正在使用 Apache HTTP Server 来运行 Apache Tomcat 配置，则每次创建或删除 DB2 Alphablox 应用程序定义时都必须重新启动 Apache Web Server。为了使 Apache 能够读取当您创建或删除 DB2 Alphablox 应用程序定义时 DB2 Alphablox 对 Apache 配置文件 (mod_jk.conf-alphablox) 所作的更改，重新启动是必需的。

停止并重新启动 Apache HTTP Server

要停止 Apache，如果它正在控制台窗口中运行，则运行 shutdown 命令，或者如果它正在作为 Windows 服务运行，则从服务控制面板文档中选择该服务并单击停止按钮。

要重新启动 Apache，如果它正在控制台窗口中运行，则运行 startup 命令，或者如果它正在作为 Windows 服务运行，则从服务控制面板文档中选择该服务并单击启动按钮。

要了解有关停止和启动 Apache Web Server 的更多详细信息，请参阅 Apache 文档。

在 Microsoft Internet Information Server (IIS) Web 服务器上注册应用程序

如果正在使用 IIS Web 服务器来运行独立配置，则每次创建或删除 DB2 Alphablox 应用程序定义时都必须执行两项任务：

- 必须添加或删除 DB2 Alphablox 应用程序的虚拟目录。
- 必须重新启动 Web 服务器。

在 Alphablox 8.4 上的 Sun iPlanet Web 服务器上注册应用程序

如果正在使用 DB2 Alphablox 8.4 (不是 8.4.1) 上的 Sun iPlanet Web 服务器来运行独立配置，则每次创建或删除 DB2 Alphablox 应用程序定义时都必须执行两项任务：

- 『添加文档目录』
- 『对 DB2 Alphablox 样式添加适当的指定』

添加文档目录

对于每个 DB2 Alphablox 应用程序，必须将文档目录添加到 Web 服务器，以便该 Web 服务器可以为此目录中的页面提供服务。下列步骤描述了如何将应用程序目录添加至 iPlanet Web 服务器。

1. 打开 iPlanet Web 服务器管理页面。缺省情况下，这些页面是通过正在运行 iPlanet 的服务器上的端口 8888 访问的，但是您的系统上的端口可能有所不同。
2. 从下拉列表中选择服务器并单击**管理**按钮。
3. 单击右上角的**类管理器**链接。
4. 单击**内容管理**选项卡。
5. 单击**其他文档目录**链接。
6. 在 **URL 前缀**文本框中，输入应用程序的上下文名称并在该名称后面输入一个正斜杠 (/)，如以下示例所示：

```
MyApplication/
```

要点：不要在上下文名称前面放置正斜杠，否则可能会损坏 iPlanet 配置文件。

7. 在**映射到目录**文本框中，输入应用程序的目录，如以下示例所示：

```
d:/Alphablox4/webapps/MyApplication
```

技巧：在路径名中只使用正斜杠 (/)；某些版本的 iPlanet 不能正确地处理路径名中的反斜杠 (\)。

8. 从**应用样式**下拉列表中选择**无**。
9. 单击**确定**按钮，然后保存更改。

对 DB2 Alphablox 样式添加适当的指定

对于每个应用程序，需要对 **alphablox** 样式指定 3 个 URL 前缀通配符。这些指定将把适当的请求转发至 DB2 Alphablox。以下是将新指定添加至 **alphablox** 样式的过程。

1. 打开 iPlanet Web 服务器管理页面。缺省情况下，这些页面是通过正在运行 iPlanet 的服务器上的端口 8888 访问的，但是您的系统上的端口可能有所不同。
2. 从下拉列表中选择服务器并单击**管理**按钮。

3. 单击**样式**选项卡。
4. 单击**指定样式**链接。
5. 从下拉列表中选择 `alphablox` 样式。
6. 如果您的应用程序是标准的 DB2 Alphablox 应用程序，则添加 3 个要对 `alphablox` 样式指定的 **URL 前缀通配符**：

```
MyApplication/abx/*  
MyApplication/servlet/*  
MyApplication/*.jsp
```

其中 `MyApplication` 是应用程序的上下文名称。

7. 如果应用程序使用 3.x 兼容性，则添加下列的 **URL 前缀通配符**（而不是添加上一个步骤中的通配符）以指定 `alphablox` 样式：

```
MyApplication/*
```

其中 `MyApplication` 是应用程序的上下文名称。

8. 保存这些更改并对所有 URL 转发请求应用这些更改。
9. 如果您想要确认已正确地添加了指定，则单击**列示指定**链接并检查对 `alphablox` 样式指定的内容的列表。

导入现有的 J2EE 应用程序

可以导入现有的 J2EE 应用程序以将其定义为 DB2 Alphablox 应用程序。在将 J2EE 应用程序导入到 DB2 Alphablox 中之后，该应用程序就可以访问所有 DB2 Alphablox 应用程序服务，如 Blox 显示和书签等等。

在与 WebSphere 一起运行时，导入 J2EE 应用程序是一种常见的使应用程序可用的方法。要了解在与 WebSphere 一起运行时定义应用程序的详细信息，请参阅第 29 页的『在与 WebSphere 一起运行时定义应用程序』。

注：如果导入在安装 DB2 Alphablox 之前创建的任何现有 Apache Tomcat 应用程序，则还需要在不同位置（包括 `$CATALINA_HOME/conf/context.xml` 文件）中添加 `crossContext=true`。要了解更多信息，请访问 <http://tomcat.apache.org/tomcat-5.5-doc/config/context.html>。

要将现有 J2EE 应用程序导入到 DB2 Alphablox 中：

1. 您必须了解要导入到 DB2 Alphablox 中的现有 J2EE 应用程序的上下文名称。
2. 作为 `admin` 用户或作为隶属于 `administrators` 组的用户登录到 DB2 Alphablox 主页。
3. 单击**管理**选项卡。
4. 单击**应用程序**链接。
5. 单击**创建**按钮。
6. 将**导入 J2EE 应用程序**下拉列表设置为**是**。“导入现有 J2EE 应用程序”窗口出现。
7. 按**查找应用程序**按钮。“导入应用程序”窗口出现。
8. 使用下拉列表，选择 **J2EE 应用程序名**（Apache Tomcat 中未提供）和**模块名**。只读 **URI** 和**上下文路径**字段根据您的选择显示值。

9. 按**选择应用程序**按钮。对话框关闭，导入现有的 **J2EE** 应用程序表单将自动填充。
10. 按**保存**完成所选 J2EE 应用程序的导入过程。按“取消”返回应用程序首页，或选择**导入 J2EE 应用程序**下的**否**返回**创建应用程序**窗口。

在成功地导入 J2EE 应用程序之后，新应用程序将显示在应用程序列表中。该应用程序将具有第 27 页的『DB2 Alphablox 应用程序定义』中描述的结构。

第 7 章 数据源定义

本章包含关于多维数据源和关系数据源的管理信息，并且提供了用来在 DB2 Alphablox 中创建、修改和删除数据源定义的过程。

定义新的数据源

DB2 Alphablox 支持多维数据源和关系数据源。**管理**选项卡的**数据源**页面允许定义数据源以供一个或多个 DB2 Alphablox 应用程序访问。DB2 Alphablox 应用程序使用这些数据源来访问关系数据库和多维数据库。根据所定义的数据源类型的不同，**管理**选项卡中的**数据源**页面也会有所不同。

执行下列操作以定义新的数据源：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。
3. 单击**数据源**链接。
4. 单击**创建**按钮。**创建数据源**页面显示。
5. 在**数据源名称**文本框中，提供该数据源的唯一名称（必需）和描述。当在非英语语言环境中运行时，允许的字符是 A-Z、a-z、0-9、下划线或特殊字符（例如，重音字符）。名称的显示是区分大小写的，但认证的实际名称不区分大小写。名称 *Public*、*Private* 和 *Properties* 已被保留，不能用作对象名。
6. 从**适配器**下拉列表中，选择用来连接至该数据源的数据库适配器。

注：DB2 Alphablox 提供了一个已存数据适配器来测试客户机与服务器之间的连接。如果您怀疑这方面有问题，请与 DB2 Alphablox 客户支持联系以获取有关使用此适配器的帮助。

7. 在**服务器名**文本框中，指定数据源所在的主机的名称。（某些名称对于不同的数据适配器是不同的。）
8. 指定任何其他特定于适配器的用于访问数据源的信息，例如：**缺省目录**、**端口号**、**数据库（目录）**、**模式**、**SID**、**用户名和密码**。

注：对于 WebSphere 和 WebLogic 应用程序服务器安装，**密码**将做为**数据源密码**显示，这是因为此密码将不用于应用程序的认证。在这些情况下，用户认证是由 WebSphere 服务器或 WebLogic 服务器处理的。当对应用程序启用了使用 **DB2 Alphablox 用户名和密码**时，此密码将仅用于数据源认证。

注：Microsoft SQL Server 用户：注意，数据库的名称符合 Microsoft SQL Server 中的标识格式规则。标识分为两类：正规标识和定界标识。任何不符合正规标识格式规则的标识都是定界标识，并且必须始终由双引号或方括号定界。例如，如果 MS SQL 数据库名称为 *test-1*，则 DB2 Alphablox 数据源名中的数据库（目录）名应该是 **[test-1]**（数据源名包含方括号）或 **"test-1"**（数据源名包含引号）。

9. 如果您使用的是 Microsoft SQL Server 数据源，则必须指定被定义为 SQL Server 认证用户的数据库用户。使用 Windows 认证的 SQL Server 用户无法正确地与 DB2 Alphablox 配合工作。
10. 如果要限制此数据源返回的查询结果集的大小，则指定**最大行数**和**最大列数**。
注：不要将此值设置为小于 20。对于大部分应用程序来说，缺省值 1000 应该足够。如果大型查询被截断，则用户在浏览器中将看到一个错误对话框。
11. 如果您使用的是 Oracle 数据源，则在**行预取**字段中指定准备显示时应该检索的行数。缺省值是 100。
12. 单击**保存按钮**以定义新数据源并返回到**数据源**页面中。

更改或删除现有数据源定义

本节描述如何修改或删除现有数据源。

更改现有数据源定义

执行下列操作以修改现有数据源：

1. 作为 *admin* 用户或作为隶属于 administrators 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。
3. 单击**数据源**链接。
4. 单击**编辑按钮**。**编辑数据源**页面显示。
5. 更改适当的值。
6. 单击**保存按钮**以更改定义并返回到**数据源**页面中。

删除现有数据源定义

执行下列操作以删除现有数据源：

1. 作为 *admin* 用户或作为隶属于 administrators 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。
3. 单击**数据源**链接。**数据源**页面显示。
4. 单击**删除按钮**以删除数据源定义并返回到**数据源**页面中。

要点：删除数据源的操作将永久地从 DB2 Alphablox 中除去该数据源。

为 Microsoft Analysis Services 数据源设置 Microsoft 认证

包括 Microsoft SQL Server 2000 Analysis Services 和 Microsoft SQL Server 2005 Analysis Services 的 Microsoft Analysis Services 服务器使用基于 Windows 的认证。要在访问 Microsoft Analysis Services 的 DB2 Alphablox 应用程序中使用基于 Windows 的认证，DB2 Alphablox 传递给 Microsoft Analysis Services 的 `userName` 和 `password` 属性必须是 Windows 用户和密码。如果 DB2 Alphablox 与 Microsoft Analysis Services 位于不同的域中，则它们必须是可信 Windows 域。

为了使 DB2 Alphablox 能够正确地使用户登录到 Microsoft Analysis Services 中，运行 DB2 Alphablox 时使用的 Windows 用户（即，当 DB2 Alphablox 启动时已登录的 Windows 用户）必须具有“以操作系统方式操作”用户权限。如果正在将 DB2 Alphablox 作为 Windows 服务运行，则可以将该服务配置为在缺省 Windows 用户之下

运行。本节描述将 Windows 机器配置为使用这种形式的 Microsoft Analysis Services 认证所需执行的步骤，并且，本节包含下列过程：

- 『设置 Windows 用户权限』
- 第 40 页的『配置 Windows 服务』
- 第 40 页的『确保在 Microsoft Analysis Services 中配置了用户』

设置 Windows 用户权限

本节描述将 Windows 机器配置为访问 Microsoft Analysis Services 数据库数据所需执行的步骤。

注：如果正在集群环境中运行 DB2 Alphablox 并访问 Microsoft Analysis Services 数据源，则必须将集群的每个节点都配置为在同一个 Windows 用户之下运行，并且每台机器都必须设置安全策略，如本节所述。

设置 Windows 用户的权限

为了使 DB2 Alphablox 能够正确地连接到 Microsoft Analysis Services，运行 DB2 Alphablox 时使用的 Windows 用户（即，当 DB2 Alphablox 启动时登录的 Windows 用户，或者，如果 DB2 Alphablox 正在作为服务运行，则为用来运行服务的用户）必须具有“以操作系统方式操作”用户权限。

注：必须在 Microsoft Analysis Services 中设置该 Windows 用户。

在 Windows 系统上，执行下列步骤来对用来运行 DB2 Alphablox 的用户启用此用户权限。

1. 从 Windows 的开始菜单中，选择**所有程序 > 管理工具 > 用户管理器**。
“Windows 用户管理器”窗口显示。
2. 从**策略**菜单中，选择**用户权限**。
3. 选择标注为**显示高级用户权限**的复选框。
4. 在标注为**权限**的下拉列表中，选择**以操作系统方式操作**。
5. 单击**添加**按钮。
6. 在**添加用户和组**对话框中，确保标注为**查找范围**的下拉列表显示了要添加的帐户的正确域名或机器名。
7. 单击**显示用户**按钮。
8. 在**名称**列表框中，选择要添加的用户帐户。
9. 单击**添加**按钮。该用户名现在列在**添加名称**列表框中。
10. 重新引导机器。

注：DB2 Alphablox 数据源中访问 Microsoft Analysis Services 的用户必须与被授予“以操作系统方式操作”用户权限的 Windows 用户相匹配。

设置 Windows 用户的权限

在 Windows 系统上，执行下列步骤来对用来运行 DB2 Alphablox 的用户启用此用户权限。

1. 从 Windows 的开始菜单中，通过选择**设置 > 控制面板**打开控制面板文件夹。
2. 打开**管理工具**。

3. 从**管理工具**控制面板文件夹中打开**本地安全策略**文档。
4. 在**本地安全设置**窗口中展开**本地策略**文件夹。
5. 选择**用户权利指派**文件夹。
6. 在**本地安全设置**窗口的右窗格中，双击**以操作系统方式操作用户权限**。
7. 在**以操作系统方式操作用户权限**的**本地安全策略设置**窗口中，单击**添加**按钮并选择用来运行 DB2 Alphablox 的用户。如果 DB2 Alphablox 正在某个网络用户下运行，则确保选择正确的网络用户（例如，对于 CORPNET Windows 网络上名为 *alphablox* 的网络用户，选择 CORPNET\alphablox）。
8. 选取**本地策略设置**框并单击**确定**按钮。
9. 关闭**本地安全策略**窗口。
10. 重新引导机器。

注：DB2 Alphablox 数据源中访问 Microsoft Analysis Services 的用户必须就是被授予“以操作系统方式操作”用户权限的 Windows 用户。

配置 Windows 服务

要将 DB2 Alphablox 配置为使用与已登录用户不同的用户来运行，必须将 DB2 Alphablox 作为服务来运行。如果 DB2 Alphablox 连接至 Microsoft Analysis Services，则必须在 Microsoft Analysis Services 中设置该 Windows 用户，并且该用户必须具有“以操作系统方式操作”用户权限。

要将该服务配置为在另一个 Windows 用户之下启动，请执行下列步骤：

1. 从 Windows 的“开始”菜单中，选择**设置**。
2. 从下拉菜单中选择**控制面板**。
3. 双击**服务**图标。
4. 在**服务**窗口中，双击 **DB2 Alphablox (AlphabloxAnalytics)** 服务。
5. 在**服务**对话框中，选择**此帐户**单选按钮。
6. 在右边的文本框中输入用户名。
7. 在**密码**文本框中输入密码。
8. 通过在**确认密码**文本框中再次输入该密码来进行验证。
9. 单击**确定**按钮。

确保在 Microsoft Analysis Services 中配置了用户

在 DB2 Alphablox 应用程序中指定的用于连接至 Microsoft Analysis Services 的任何用户，无论是通过 DB2 Alphablox 数据源配置的还是通过应用程序中的用户名和密码参数（或相关联的方法）指定的，都必须是 Windows 用户，并且必须在 Microsoft Analysis Services 中进行了正确配置。注意，Windows 域组 Everyone 仅包括域用户，而不包括本地机器用户。

注：如果您尝试作为机器的本地用户登录到 Microsoft Analysis Services，则该用户必须隶属于 Administrators 组以及 OLAP Administrator 组才能可靠地进行连接。如果该用户并非同时隶属于这两个管理员组，则该用户第一次将能够登录，但后续的尝试将失败，并且将发生类似于以下的异常：

com.alphablox.util.NotAuthorizedException: 找不到提供程序。可能是未正确地安装

使用 JDBC 数据源

本节包括下列与 JDBC 数据源相关的小节:

- 『为 Sybase JConnect 关系驱动程序设置环境』
- 『设置 JDBC 跟踪』
- 第 42 页的『将受支持的 JDBC 驱动程序更新到另一个版本』
- 第 42 页的『添加其他 JDBC 驱动程序』
- 第 42 页的『修改类路径设置』

为 Sybase JConnect 关系驱动程序设置环境

如果正在使用 Sybase JConnect 驱动程序来连接至 Sybase 关系数据库, 则必须在 Sybase 数据库服务器上运行一个 SQL 脚本文件。该 SQL 脚本文件位于以下目录中:

```
<db2alphablox_dir>/tools/sql
```

其中 <db2alphablox_dir> 是 DB2 Alphablox 的安装目录。下表显示了该 SQL 脚本的名称以及运行该脚本时必须使用的 Sybase 数据库服务器。

SQL 脚本	运行脚本所用的 Sybase 数据库服务器
sql_asa.sql	SQL Anywhere 5.5.02 或更高版本
sql_server.sql	低于版本 12 的 SQL Server 版本
sql_server12.sql	SQL Server V12 和更高版本

确定正在运行的 Sybase 数据库服务器并在该数据库服务器上运行适当的脚本。每个脚本都设置 Sybase JConnect JDBC 驱动程序所需的一些存储过程和系统对象。要了解有关在 Sybase 数据库服务器上运行 SQL 脚本文件的详细信息, 请参阅 Sybase 文档。

必须在 Sybase 数据库服务器上执行适当的 SQL 脚本才能够从 DB2 Alphablox 连接至该数据库服务器。

设置 JDBC 跟踪

对于 DB2 Alphablox 中的任何关系数据源驱动程序, 可以启用 JDBC 跟踪。启用 JDBC 跟踪后, JDBC 驱动程序将把参考消息记录至 DB2 Alphablox 控制台和日志文件。所有 JDBC 跟踪消息都在 INFO 消息级别记录。

要对关系数据源启用 JDBC 跟踪, 请将**启用 JDBC 跟踪**下拉列表设置为**是**。此选项仅可用于使用某一受支持关系数据库驱动程序的数据源。在启用 JDBC 跟踪之后, 将消息级别设置为 INFO 或更高。例如, 在“控制台”窗口中, 输入以下命令:

```
report info
```

使用 JDBC 跟踪来对连接至关系数据库时发生的任何问题进行调试。在启用 JDBC 跟踪后, 将记录下列信息:

- JDBC 驱动程序日志记录信息
- DB2 Alphablox 在连接至关系数据库时返回的连接信息
- 与数据库返回给 DB2 Alphablox 的每个结果集中的列元数据相关的信息

如果您遇到连接问题, 或者需要更多关于了解 JDBC 跟踪的信息, 请与 DB2 Alphablox 客户支持联系。

将受支持的 JDBC 驱动程序更新到另一个版本

如果需要更新 DB2 Alphablox 用来访问关系数据库的其中一个受支持的 JDBC 驱动程序，则执行下列步骤：

要点： DB2 Alphablox 仅测试了《安装指南》的『系统需求』一节中列出的受支持的 JDBC 驱动程序；其他驱动程序尚未经过测试，并且可能会导致意外的结果。要了解更多信息，请与 DB2 Alphablox 客户支持联系。

1. 获取新的 JDBC 驱动程序的 Java 类。通常，这些文件具有 zip 或 jar 文件扩展名（例如，ojdbc14.jar）。
2. 查找 JDBC 驱动程序安装期间指定的目录。
3. 将您正在更新的 JDBC 驱动程序文件重命名（例如，将其更改为 <driverName>.zip.old）以保留一个副本以备需要还原到受支持的版本时使用。
4. 将新的驱动程序文件重命名为与现有的受支持驱动程序文件同名，然后将其复制到 <db2alphablox_dir>/lib/ 目录中。
5. 更新 DB2 Alphablox 启动环境以访问新的 JDBC 驱动程序类，如下一节所述。
6. 在更新类路径信息之后，应重新启动服务器以使更改生效。

添加其他 JDBC 驱动程序

缺省情况下，在进行安装或升级后，可供 DB2 Alphablox 应用程序使用的关系数据源在 jdbcdrivers.xml 文件中定义。您不应该修改此文件 - 在升级 DB2 Alphablox 期间，将自动覆盖此文件。如果需要创建其他 DB2 Alphablox 数据源，则可使用可选的 jdbcdrivers_additional.xml 文件。

执行下列步骤来为 DB2 Alphablox 创建其他数据源：

1. 查找位于以下目录中的 jdbcdrivers_additional.xml 文件：
<db2alphablox_dir>/repository/servers/
2. 您可以编辑此文件，但应该在同一个目录中创建此文件的备份副本并将其命名为 jdbcdrivers_additional.xml。在同一个目录中有缺省的 jdbcdrivers_additional.xml 文件的备份副本，该副本名为 jdbcdrivers_additional.xml.example。
3. 通过使用此文件中包含的 JDBC 驱动程序示例，创建您所需的任何其他数据源，并确保：
 - a. 在要提供的每个新 JDBC 驱动程序定义中，将 disabled 属性由 true 更改为 false。缺省情况下，示例 JDBC 驱动程序处于禁用状态。
4. 必须重新启动应用程序服务器才能使这些更改生效。
5. 编辑所有 DB2 Alphablox 类路径以包括新的驱动程序，如下面的第 42 页的『修改类路径设置』所述。

修改类路径设置

在将其他驱动程序添加到 jdbcdrivers_additional.xml 文件中之后，还必须更新类路径信息以便可以找到新的 JDBC 驱动程序。根据您的配置的具体情况来执行下列步骤。

WebSphere

要在 WebSphere 上添加或删除 JDBC 驱动程序，不需要修改任何类路径设置。只要将其他 JDBC 驱动程序放到 `<websphere_dir>/AppServer/lib/ext` 目录中就可以添加那些 JDBC 驱动程序，也可以从该目录中删除没用的或旧的 JDBC 驱动程序。应重新启动应用程序服务器以使类路径更改生效。

WebLogic

对于 BEA WebLogic 应用程序服务器安装，可以修改以下批处理命令文件中的类路径信息：`<db2alphablox_dir>/bin/jdbcsetup.bat` (Windows) 或 `<db2alphablox_dir>/bin/jdbcsetup.sh` (UNIX)。应重新启动应用程序服务器以使更改生效。

Tomcat

执行下列步骤来修改仅在 Apache Tomcat 下运行的 DB2 Alphablox 的类路径信息：

Windows: 如果 Tomcat 在 Windows 控制台中运行，则可以在文本编辑器中打开 `<db2alphablox_dir>/bin/jdbcsetup.bat` 文件，然后根据需要修改类路径设置。

如果应用程序服务器作为 Windows 服务运行，则可以在文本编辑器中打开 `<tomcat_dir>/conf/wrapper.properties` 文件（此文件位于 Tomcat 目录中，而不是位于 DB2 Alphablox 目录中），然后根据需要修改类路径设置。应重新启动应用程序服务器以使更改生效。

Linux 和 UNIX: 如果在 Linux 或 UNIX 系统上将 Apache Tomcat 与 DB2 Alphablox 配合使用，则可以在文本编辑器中打开 `<tomcat_dir>/bin/analytics.sh` 文件，然后根据需要修改类路径设置。应重新启动应用程序服务器以使更改生效。

第 8 章 用户定义

DB2 Alphablox 用户管理页面允许您创建、更改和删除用户。

也可以通过按每个页面右上角的[帮助](#)链接来获得有关[应用程序](#)和[数据源](#)页面的信息。

注：要更改现有的用户，在转至另一页面之前务必单击**保存**按钮。否则，将不保存对页面所作的更改。

创建新用户

管理选项卡下方的[用户](#)链接将打开用来创建和编辑新用户的页面。在安装期间，将创建两个用户（Admin 和 Guest）。单击[用户](#)页面中的**创建**按钮来定义其他用户。选择现有用户名并单击**编辑**或**删除**按钮来修改或删除它。如果用户损坏，则可以通过重新启动 DB2 Alphablox 来修复它们。本节提供用来从 DB2 Alphablox 主页中定义新用户的步骤。

执行下列操作以定义新的用户：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 单击[用户](#)链接。**用户**页面显示。
4. 单击**创建**按钮。**创建用户**页面显示。

注：根据您的安全性配置的不同，此屏幕可能未包含密码信息。

5. 在[用户名](#)文本框中，输入用户的标识（必需）。用户名是用来登录到 DB2 Alphablox 的名称。当在非英语语言环境中运行时，允许的字符是 A-Z、a-z、0-9、下划线或特殊字符（例如，重音字符）。用户名的显示是区分大小写的，但认证的实际用户名不区分大小写。名称 *Public*、*Private* 和 *Properties* 已被保留且其不能用作对象名。
6. （可选）输入用户的**全名**、**描述**和**电子邮件地址**。
7. 在[密码](#)文本框中，输入用户的初始密码。密码区分大小写。
8. 在[确认密码](#)文本框中，重新输入密码以确认它。
9. 在[允许用户编辑概要文件](#)下拉列表中，指示该用户是否具有编辑概要文件的特权。将此属性设置为**是**将允许该用户使用 DB2 Alphablox 主页上的[我的概要文件](#)链接来访问和修改他的概要文件。
10. 提供任何定制属性的值。（如果定义了任何定制用户属性，则它们将显示在[允许用户编辑概要文件](#)下拉列表下方。）
11. 单击**保存**按钮以定义新用户并返回到“用户”页面。

注：创建新用户时，将自动地把该用户指定为 *Public* 组的成员。

更改或删除现有的组或用户

本节提供用来从 DB2 Alphablox 主页中更改现有用户或组的属性或删除它们的步骤。

更改现有用户的属性

执行下列操作以更改现有用户的属性:

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 单击**用户**链接。**用户**页面显示。
4. 单击**编辑**按钮。**编辑用户**页面显示。
5. 更改任何需要更改的值。

注: 在 WebSphere 和 WebLogic 安装中, “密码”将显示为“数据源密码”, 这是因为此密码将不会用于应用程序的认证。在这些情况下, 用户认证是由 WebSphere 或 WebLogic 处理的。这还表示当对应用程序使用 DB2 Alphablox 用户名和密码时, 该密码将仅用于数据源认证。

6. 单击**保存**按钮以保存您对**一般属性**面板所作的更改并返回到**用户**页面。
7. 要更改用户的应用程序属性, 请单击**应用程序属性**选项卡。进行任何必需的更改, 然后单击**保存**按钮。
8. 要更改用户的组成员资格或角色成员资格, 请单击**成员资格**选项卡。进行任何必需的更改。注意, 对成员资格的更改将立即生效。要了解有关组成员资格和角色成员资格的信息, 请参阅第 46 页的『更改用户所属的组』和第 54 页的『更改用户或组所属的角色』。

删除现有的用户

执行下列操作以删除现有的用户:

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 单击**用户**链接。**用户**页面显示。
4. 从用户列表中选择用户。
5. 单击**删除**按钮以删除该用户。

要点: 删除用户的操作将永久地从 DB2 Alphablox 中除去该用户。

更改用户所属的组

本节显示添加或修改用户和组的组成员资格的步骤。

执行下列操作以修改用户所属的组:

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 单击**用户**链接。**用户**页面显示。
4. 单击**编辑**按钮。**编辑用户**页面显示。
5. 单击**成员资格**选项卡。
6. 要添加用户所属的新组, 在**可用的组**列表中选择一個组并单击向左箭头按钮以将其移至**组成员资格**列表。
7. 要除去用户所属的组, 在**组成员资格**列表中选择一個组并单击向右箭头按钮以将其移至**可用的组**列表。

8. 要添加用户所属的新角色，在**可用的角色**列表中选择角色，然后单击向左箭头按钮以将其移至**角色成员资格**列表。

注： 对用户的组或角色成员资格所作的更改将立即生效。

第 9 章 组定义

组管理页面允许您创建、更改和删除组。

也可以通过按每个页面右上角的[帮助](#)链接来获得有关组页面的信息。

注：要更改现有的组，在转至另一页面之前务必单击**保存**按钮。否则，将不保存对页面所作的更改。

创建新组

管理选项卡下方的[组](#)链接将打开用来创建和编辑新组的页面。在安装期间，将创建两个组（Administrators 和 Public）。单击组页面中的**创建**按钮来定义其他组。选择现有组名并单击**编辑**或**删除**按钮来修改或删除它。如果某个组损坏，则可以通过重新启动 DB2 Alphablox 来修复它。本节提供用来从 DB2 Alphablox 主页中定义新组的步骤。

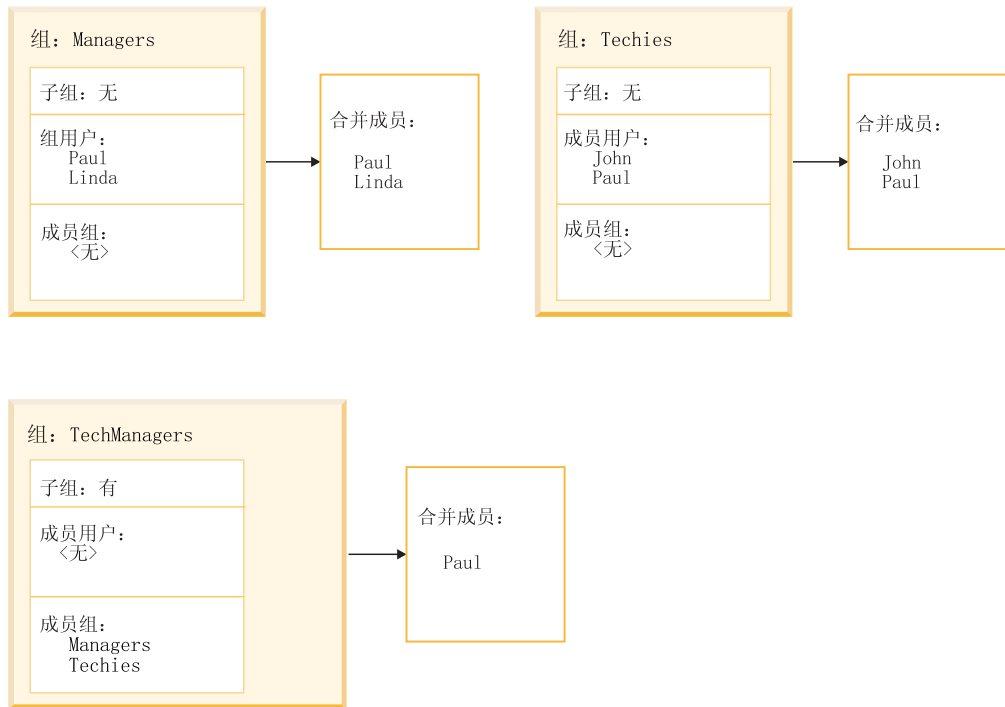
执行下列操作来定义新组：

1. 作为 *admin* 用户或作为隶属于 administrators 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。常规页面显示。
3. 单击**组**链接。组页面显示。
4. 单击**创建**按钮。创建组页面显示。
5. 在**组名**文本框中，输入组的名称（必需）。当在非英语语言环境中运行时，允许的字符是 A-Z、a-z、0-9、下划线或特殊字符（例如，重音字符）。组名的显示是区分大小写的，但认证的实际组名不区分大小写。名称 *Public*、*Private* 和 *Properties* 已被保留，不能用作对象名。
6. （可选）输入该组的**描述**。
7. 如果此组将作为子组，则在**子组**下拉列表中选择**是**，否则选择**否**。要了解有关子组的信息，请参阅第 49 页的『了解子组』。
8. 在**成员用户**和**成员组**选择框中将用户和 / 或组添加到此组中。要完成此操作，请滚动用户和组的列表。按下 **Ctrl** 键以选择多个要添加到此组中的用户或组。
9. 提供任何定制属性的值。（如果定义了任何定制组属性，则它们将显示在**成员组**选择框下方。）
10. 单击**保存**按钮以定义新组并返回到组页面中。

了解子组

子组是一种机制，在这种机制下，属于多个组的用户可以自动成为另一个组的成员。当组的**子组**下拉列表设置为**是**时，作为结果，它的用户成员由每个组中被指定为新组成员的用户交集组成。

下图显示了使用子组的示例。在此示例中，组 *TechManagers* 使用一个子组来自动包括同时属于 *Managers* 和 *Techies* 组的成员。



通过使用 *TechManagers* 组的子组，管理员将创建一个组，该组自动包括 *Managers* 与 *Techies* 组的所有公共成员的集合。在此示例中，同时属于那些组的用户只有 *Paul*。如果将其他用户同时添加到 *Managers* 和 *Techies* 组中，则他们将自动成为 *TechManagers* 组的成员。

注： 在使用子组时，仍然可以通过在 **成员用户** 选择框中选择各个用户来将他们手工添加到组中。

更改或删除现有的组

本节提供用来从 DB2 Alphablox 主页中更改现有用户或组的属性或删除它们的步骤。

更改现有的组

执行下列操作以修改现有的组：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。常规页面显示。
3. 单击**组**链接。组页面显示。
4. 从组列表中选择组。
5. 单击**编辑**按钮。编辑组页面显示。
6. 更改任何需要更改的值。
7. 单击**保存**按钮以保存更改并返回到用户页面。
8. 要更改用户的应用程序属性，请单击**应用程序属性**选项卡。
9. 进行任何必需的更改。
10. 单击**保存**按钮。

11. 要更改用户的组成员资格或角色成员资格，请单击**成员资格**选项卡。
12. 进行任何必需的更改。注意，对成员资格的更改将立即生效。要了解有关组成员资格和角色成员资格的信息，请参阅第 46 页的『更改用户所属的组』和第 54 页的『更改用户或组所属的角色』。
13. 单击**保存**按钮。

删除现有的组

执行下列操作以删除现有的组：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 单击**组**链接。**组**页面显示。
4. 从组列表中选择组。
5. 单击**删除**按钮以删除该组。

要点：删除组的操作将永久地从 DB2 Alphablox 中除去该组。

第 10 章 角色定义

角色管理页面允许您创建、更改和删除角色。仅当正在 Apache Tomcat 配置中运行 DB2 Alphablox 时，“角色”页面才会显示；如果您正在使用 WebSphere 或 WebLogic，则应用程序服务器提供此功能。用户角色通常是在 Web 应用程序服务器的应用程序部署工具中指定的。角色一旦在部署工具中进行了定义（并添加到部署描述符文件），就可以在 DB2 Alphablox 中正确地使用。

也可以通过按每个页面右上角的[帮助](#)链接来获得有关角色页面的信息。

注：要更改现有的角色，在转至另一页面之前务必单击**保存**按钮。否则，将不保存对页面所作的更改。

定义新角色

DB2 Alphablox 可以通过角色来控制对应用程序的访问。角色包含用户和 / 或组的列表以及与列表中的每个条目相关联的访问许可权。

执行下列操作以创建新角色：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 单击**角色**链接。**角色**页面显示。
4. 单击**创建**按钮。**创建角色**页面显示。
5. 在**角色名**文本框中，输入角色的名称（必需）。当在非英语语言环境中运行时，允许的字符是 A-Z、a-z、0-9、下划线或特殊字符（例如，重音字符）。名称的显示是区分大小写的，但认证的实际名称不区分大小写。名称 *Public*、*Private* 和 *Properties* 已被保留，不能用作对象名。
6. （可选）输入该角色的**描述**。
7. 单击**保存**按钮以定义新角色并返回到**角色**页面中。

更改和删除现有的角色

执行下列操作以更改现有的角色：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 单击**角色**链接。**角色**页面显示。
4. 从角色列表中选择角色。
5. 单击**编辑**按钮。**编辑角色**页面显示。
6. 如果有必要的话，更改角色的描述。
7. 在继续之前，单击**保存**按钮，否则将不保存更改。

8. 要更改指定给该角色的组，请单击**成员组**选项卡。这将显示一个页面，该页面列出了当前隶属于该角色的组以及当前未隶属于该角色的可用组的列表。使用左箭头按钮和右箭头按钮来将选择的项从**可用的组**列表移到**成员组**列表以及从**成员组**列表移到**可用的组**列表。
9. 单击**保存**按钮以保存角色的组成员资格属性。
10. 要更改隶属于角色的用户，请单击**成员用户**选项卡。这将显示一个页面，该页面列出了当前隶属于该角色的用户以及当前未隶属于该角色的可用用户的列表。使用左箭头按钮和右箭头按钮来将选择的项从**可用的用户**列表移到**成员用户**列表以及从**成员用户**列表移到**可用的用户**列表。
11. 单击**保存**按钮以保存角色的用户成员资格属性。

更改用户或组所属的角色

执行下列操作以更改用户或组被指定到的角色：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 单击**用户**链接或**组**链接。**角色**页面显示。
4. 从**用户**或**组**页面上的用户（或组）列表中选择用户（或组）。
5. 单击**编辑**按钮。**编辑用户**（或**编辑组**）页面打开。
6. 单击**成员资格**选项卡。
7. **成员资格**面板下半部分有一个**角色成员资格**框和一个**可用的角色**框。使用左箭头按钮和右箭头按钮来将选择的项从**可用的角色**列表移到**角色成员资格**列表以及从**角色成员资格**列表移到**可用的角色**列表。

删除现有的角色

执行下列操作以删除现有的角色：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 单击**角色**链接。**角色**页面显示。
4. 从角色列表中选择角色。
5. 单击**删除**按钮以删除该角色。

要点：删除角色的操作将永久地从 DB2 Alphablox 中除去该角色。

第 11 章 安全性和认证

这些主题描述使用在 Apache Tomcat 配置中运行的 DB2 Alphablox、使用外部 Web 服务器或使用另一个 Web 服务器（Sun iPlanet、Microsoft IIS 或 Apache HTTP Server）时的安全性和认证。

要了解在与商用应用程序服务器（如 WebSphere 或 WebLogic）一起运行时的安全性和认证信息，请参阅那些产品的文档。

注：DB2 Alphablox 8.4 和 8.4.1 支持不同的外部 Web 服务器。DB2 Alphablox 8.4.1 不支持 Sun iPlanet。DB2 Alphablox 8.4.1 支持 Apache HTTP Server 2.0，而 DB2 Alphablox 8.4 支持 Apache HTTP Server 1.3。要了解详细信息，请参阅《安装指南》中的『系统需求』一节。

DB2 Alphablox 认证方式和安全方式

用户必须先通过认证，DB2 Alphablox 才能将页面传递给用户。这种情况对于任何底层 Web 服务器都成立。然而，当使用外部 Web 服务器时，对 DB2 Alphablox 进行的认证通常是透明的。

缺省情况下，不会将 HTML 页面传递到在 Web 服务器后端配置了 DB2 Alphablox 的应用程序服务器，并且，在该应用程序服务器接收到 JSP 页面或具有 DB2 Alphablox 内容的页面之前，不会请求那些 HTML 页面。可以将 Web 服务器配置为对 HTML 页面进行认证，但缺省情况下，DB2 Alphablox 配置不支持这种认证。如果要在 Web 服务器后端配置了 DB2 Alphablox 时进行认证，请将 Web 服务器配置为支持对所有 HTML 页面进行认证，或者确保欢迎页面是 JSP 文件。

DB2 Alphablox 支持两种类型的用户认证：

认证类型	描述
应用程序服务器认证	当 DB2 Alphablox 在未带有外部 Web 服务器的应用程序服务器中运行时，认证和授权由此应用程序服务器管理。
Web 服务器认证	当将 Sun iPlanet、Apache HTTP Server 或 Microsoft IIS 用作 Web 服务器时，Web 服务器执行认证操作。当 DB2 Alphablox 尝试认证用户时，Web 服务器将用户标识传递给 DB2 Alphablox。 注： Web 服务器不会同时将用户标识和密码传递给 DB2 Alphablox，因为这样就违反了安全性要求。

Alphablox 8.4.1 中的安全模型

DB2 Alphablox 8.4.1 提供了基于标准 Servlet 2.4 安全性 API 的新安全模型。因为 Alphablox 8.4.1 和 8.4 支持不同版本的 Apache Tomcat，所以对安全性的支持也有所不同，这是因为这两个 Tomcat 版本支持不同的安全模型和 servlet 规范。在 DB2 Alphablox 8.4.1 中，现在直接在 Apache Tomcat 的 `conf/server.xml` 文件中配置安全性

信息。要在 Tomcat 5.5 中配置安全性，请参阅有关配置域的 Apache Tomcat 文档 (<http://tomcat.apache.org/tomcat-5.5-doc/realms-howto.html>)。

在下列三个包中提供了支持此安全模型的 Alphablox API:

- com.alphablox.security.jaas
- com.alphablox.security.jndi
- com.alphablox.security.ntlm

基于 Java 认证和授权服务 (JAAS) 的 Alphablox 登录模块

com.alphablox.security.jaas 包提供基于 JAAS 的 Alphablox 登录模块。com.alphablox.security.jaas.AlphabloxLoginModule 类通过从 DB2 Alphablox 存储库中访问认证信息来认证用户。如果在 Tomcat 5.5 安装目录的 *conf/alphablox-jaas.config* 登录配置文件中将 debug 选项设置为 true (缺省值)，则调试消息会输出到 Alphablox 服务器日志。

Tomcat 5.5 中的 JNDI 域配置

Apache Tomcat JNDI 域不支持 LDAP 动态组，但 DB2 Alphablox 提供可用来从 LDAP 动态组中访问用户和组信息的 Alphablox JNDI 域。另外通过 LDAP 动态组搜索角色成员关系，com.alphablox.security.jndi.AlphabloxJNDIRealm 类继承了 Tomcat 5.5 中对 JNDIRealm 的基本支持。此类定义四个 JNDIRealm 中提供的属性的附加属性。

属性	描述
userName	可选。设置要作为用户的唯一名称读取的属性名。此值应该与 JNDI 域的 userPattern 或 userSearch 字符串中使用的属性匹配。缺省值是 uid。
memberURL	可选。设置动态组 URL 属性名。缺省值是 memberURL。
cacheEnabled	可选。对关联的用户名启用或禁用动态组的高速缓存。缺省值是 true。
dynamicGroupSearch	必需。设置用于搜索 LDAP 动态组的 roleBase 的 JNDI 过滤器。缺省值是 null。启用高速缓存 (cacheEnabled="true") 有助于改善性能，因为不需要重复查询动态组。

要了解详细的 API 信息，请参阅 DB2 Alphablox 信息中心中的 Javadoc 文档。

在 Apache Tomcat 5.5 中添加 Alphablox JNDI 域

要在 Tomcat 中添加 Alphablox JNDI 域，请配置 Apache Tomcat *conf/server.xml* 文件并将 className 设置为 com.alphablox.security.jndi.AlphabloxJNDIRealm:

```
<Realm
  className="com.alphablox.security.jndi.AlphabloxJNDIRealm"
  connectionURL="ldap://your_ldap.your_server.com:port"
  userBase="..."
  userSearch="..."
  userRoleName="..."
  roleBase="..."
  roleName="..."
  roleSearch="..."
  dynamicGroupSearch="(& (cn={0}) (objectclass=groupofurls))"
/>
```

注:

- 在 XML 中，“&”字符需要编码为“&”。

- 在更改 *server.xml* 文件后，需要重新启动 Tomcat。

Admin 权限与用户权限

在安装 DB2 Alphablox 时，它将创建一个具有管理特权的单用户概要文件。此概要文件的名称是 *admin*，并且不能被更改。*admin* 的缺省密码是 *password*。当使用 Web 服务器安全性时，在 Web 服务器的认证方案下必须存在名为 *admin* 的用户。（当使用 IIS 和 Windows® NTLM 时，*admin* 可以是本地用户而不是域用户。）

admin 用户可以通过 DB2 Alphablox 的管理页面对 DB2 Alphablox 对象（包括用户属性）进行读/写访问。DB2 Alphablox 用户只能对他们自己的用户属性进行读/写访问。*admin* 用户或另一个隶属于 administrators 组的 DB2 Alphablox 用户可以将一个或多个应用程序的访问权授予特定的用户、用户组或所有用户。

除去应用程序的 Guest 登录权限

缺省情况下，安装 DB2 Alphablox 时设置了两个缺省用户，即 Guest 和 Admin。要使 Guest 用户无法访问应用程序，可以修改应用程序的 *web.xml* 文件，即将 *auth-constraint* 和 *security-role* 元素的 *role-name* 元素中的 *AlphabloxUser* 替换为 *AlphabloxAuthenticatedUser*。

应用程序服务器安全域和应用程序

通常，您在开发服务器上创建应用程序，以后，在部署应用程序时，您将应用程序移到生产服务器上。在两台未共享同一实例名的不同服务器之间移动应用程序时，开发者和管理员需要了解：所迁移的应用程序的 *web.xml* 文件中指定的安全域可能与新位置中的安全域不匹配。在 IBM WebSphere 和 BEA WebLogic 应用程序服务器上，当应用程序尝试访问同一台服务器上具有不同安全域的其他应用程序时，将提示用户进行认证。为了防止发生这种情况，需要修改应用程序的 *web.xml* 文件以使安全域匹配。

另一种可能会提示用户进行认证的情形是，将作为 WAR 或 EAR 文件接收到的应用程序导入 WebSphere 和 WebLogic Application Server。未检查实例名，并且实例名可以不同。

Web 服务器认证与 DB2 Alphablox 认证

DB2 Alphablox 仅允许认证用户访问应用程序。如果未经正确的认证，用户就无法连接到数据源、检索数据或在应用程序页面上查看数据。由于 DB2 Alphablox 不在目录级别锁定访问权，所以认证成功的使用者可以使用浏览器来尝试打开受限制的应用程序。虽然用户可以打开应用程序页面，但是不会在其中显示数据。

要了解有关防止用户浏览位于应用程序所在目录下的目录和文件的信息，请参阅第 63 页的『禁用目录浏览』。

在缺省安装中，DB2 Alphablox 执行它自己的认证。然而，如果使用另一个 Web 服务器，可以将 DB2 Alphablox 设置为让 Web 服务器执行认证。由于 Web 服务器认证可以根据局域网中可能已存在的用户帐户进行认证，所以它可以减轻 DB2 Alphablox 管理员的管理负担。本节描述了如何设置 DB2 Alphablox 以让 Web 服务器执行认证。

注：当您选择 WebServerRealm 适配器时，Web 服务器安全性将优先于任何其他安全性。在这些情况下，DB2 Alphablox 中的密码是什么并不重要。

在带有 Apache Tomcat 3.2.4 的 Alphablox 8.4 上使用 Sun iPlanet Web 服务器安全选项

对于基本文本或明文方式下的大多数用户和浏览器组合，Sun iPlanet 安全模型能够很好地发挥作用。要通过 iPlanet 提供的 URL 来管理 DB2 Alphablox，请将名为 *admin* 的用户添加到 Web 服务器的用户数据库中。否则，请直接访问 DB2 Alphablox URL（例如，使用 `http://<servername>:<port>/AlphabloxAdmin/home`）。

要对 iPlanet 使用 DB2 Alphablox 安全性，请在 iPlanet Web 服务器上禁用安全性。这样，所有用户都由 DB2 Alphablox 认证。

另外，务必进行正确的 DB2 Alphablox 设置以确保环境能够发挥作用并且安全可靠。要了解更多信息，请参阅第 62 页的『配置 DB2 Alphablox 以使用基于 Web 服务器的安全性』。

为 IIS NTLM 设置 Microsoft 安全选项

当将 DB2 Alphablox 与 Microsoft IIS Web 服务器配合使用时，可以设置安全认证，以便 IIS 在用户登录到 DB2 Alphablox 时执行认证（而不是由 DB2 Alphablox 执行认证）。如果将 IIS 配置为在可信域中接受 Windows 登录，则那些可信用户可以在不进行任何其他认证的情况下访问 DB2 Alphablox；即，当他们在登录到可信域的情况下访问 DB2 Alphablox 时，就不需要输入用户名或密码。可信用户是已在 Windows 可信域中通过认证的 Windows 用户，该可信域通常是大型企业的 Windows 服务器上的可信域。

在此配置中，IIS 处理所有请求认证。在处理请求之前，DB2 Alphablox 要求已认证该用户名的请求。因此，如果 Microsoft Internet Explorer 用户将浏览器的缺省安全设置（在“高级设置”中）更改为不提示输入用户名和密码，则 DB2 Alphablox 就不能处理来自这些浏览器的请求。

要设置 DB2 Alphablox 以便使用 NTLM 安全地操作 Microsoft Internet Information Services (IIS)，请执行下列任务：

- 第 59 页的『安装 Microsoft IIS』
- 第 59 页的『安装 DB2 Alphablox 并选择 Microsoft IIS 作为 Web 服务器』
- 第 59 页的『在 Microsoft IIS 中配置安全设置』
 - 第 59 页的『在 Microsoft IIS 中启用安全设置』
 - 第 60 页的『在 Microsoft IIS 中限制匿名用户权限』
- 第 61 页的『为 NTLM 创建名为 admin 的 Windows 本地用户』
- 第 61 页的『在 Tomcat 5.5 中为 Alphablox 8.4.1 配置 NTLM 安全性』
- 第 61 页的『登录到 DB2 Alphablox』

另外，务必进行正确的 DB2 Alphablox 设置以确保环境能够发挥作用并且安全可靠。要了解详细信息，请参阅第 62 页的『配置 DB2 Alphablox 以使用基于 Web 服务器的安全性』。

安装 Microsoft IIS

在安装 DB2 Alphablox 之前，必须先安装 Microsoft IIS。有关安装 IIS 的信息，请参阅 Microsoft 文档。

安装 DB2 Alphablox 并选择 Microsoft IIS 作为 Web 服务器

要运行 DB2 Alphablox 并使用 Microsoft IIS 作为 Web 服务器，在安装期间，必须在“选择 Web 服务器配置”页上选择创建新的 **Microsoft IIS** 配置选项来创建新的 Microsoft IIS 配置。安装过程将在 Web 服务器根目录下安装必需的文件。

要将 DB2 Alphablox 与 IIS 配合使用，必须在安装 DB2 Alphablox 之前安装 IIS。

在 Microsoft IIS 中配置安全设置

使用 Microsoft IIS Microsoft 管理控制台配置下列内容。

在 Microsoft IIS 中启用安全设置： 要启用适当的安全设置：

1. 打开 Microsoft IIS 管理控制台。
2. 在左窗格中，右键单击适当的 Web 服务器（通常是 **Default Web Site**）并选择 **Properties**。将打开 Default Web Site Properties 窗口。
3. 单击 **Directory Security** 选项卡。
4. 按 **Edit** 按钮。要通过 NTLM 使用 Windows 可信认证，请选中 **Windows Integrated authentication** 复选框（确保未选中其他复选框）。注意，只有 Microsoft Internet Explorer 用户才能以这种方式登录。

仅当指定了客户机的主机名时，这种方法才有效。

注：

- 在 DB2 Alphablox 8.4 中，要指定授权主机名，请单击 DB2 Alphablox 主页中的管理选项卡，单击一般属性部分中的系统链接，并在系统页面的已授权的客户机列表文本框中输入任何授权客户机。要限制只能在运行 DB2 Alphablox 的机器上访问客户机，请在已授权的客户机列表文本框中输入 localhost。
- 在 DB2 Alphablox 8.4.1 中，可直接在 Tomcat 5.5 中指定授权客户机。Tomcat 5.5 允许您指定“远程主机过滤器”或“远程地址过滤器”，以标识准许使用或拒绝使用的 IP 地址或主机名。例如，要阻止所有来自除 localhost 之外的主机名的请求：

```
<Valve  
  className="org.apache.catalina.valves.RemoteAddrValve" allow="127.0.0.1" />
```

要了解有关“远程主机过滤器”和“远程地址过滤器”的更多信息，请访问 <http://tomcat.apache.org/tomcat-5.5-doc/config/valve.html>。

DB2 Alphablox 将提示用户输入用户名和密码，然后由 DB2 Alphablox 进行认证。

安全类型	描述
Allow Anonymous	<p>此选项允许用户使用匿名帐户登录。如果对 IIS 使用的是 NTLM 安全性，则不要选取此选项。用户通过认证之后，它们可以浏览应用程序所在的 webapps 目录下的目录和文件。要防止浏览目录，请参阅第 63 页的『禁用目录浏览』。</p> <p>选中此复选框将授权用户不受限制地访问资源。要了解详细信息，请参阅第 60 页的『在 Microsoft IIS 中限制匿名用户权限』。</p>
Basic Authentication (clear text)	<p>对于 Mozilla Firefox 和 Microsoft Internet Explorer 浏览器，此选项会导致以明文方式传送用户标识和密码。</p> <p>注：在 Mozilla Firefox 浏览器访问 Microsoft IIS 上的 DB2 Alphablox 应用程序的环境中，务必选取此选项。</p> <p>在 Mozilla Firefox 和 Internet Explorer 浏览器都访问 Microsoft IIS 上的 DB2 Alphablox 应用程序的环境中，请选择此选项和 Windows Integrated authentication。</p>
Windows Integrated authentication	<p>这是使用 IIS 安全性的建议方法。此选项仅用于同时使用 IIS Web 服务器和 Microsoft Internet Explorer 浏览器的环境，它导致以 Microsoft 专用格式（而不是明文格式）传送用户标识和密码。</p>

- 如果选取了 **Allow Anonymous Access** 框（因而不使用可信 NTLM 安全性），请单击 **Allow Anonymous Access** 复选框右边的 **Edit** 按钮。将显示匿名用户名（通常格式为 IUSR_<HOSTNAME>）。请记录匿名用户名并转至第 60 页的『在 Microsoft IIS 中限制匿名用户权限』中的指示信息。
- 单击 **OK** 按钮以关闭 Authentication Methods 窗口。
- 单击 **OK** 按钮以关闭 Default Web Site Properties 窗口。

在 Microsoft IIS 中限制匿名用户权限： 仅当不使用可信 NTLM 安全性时，才需要执行此过程。只有 Microsoft IIS 和 Windows 高级管理员才应执行这些步骤。仅限制匿名用户对 webapps 目录的访问权不足以限制对 Web 服务器上的对象的访问权。因此，建议使用此方法。

如果在 IIS 下设置了 Allow Anonymous，则请执行下列步骤以限制匿名用户的权限：

- 打开 Windows 文件资源管理器。
- 找到 *Inetpub* 目录并单击鼠标右键以打开它的菜单。IIS 使用 *Inetpub* 目录来存储 Web 服务器允许访问的文档，其缺省目录是 *C:\inetpub*。
- 从下拉列表中选择属性以打开目录属性窗口。
- 单击安全选项卡。
- 按窗口底部的添加按钮以打开添加用户和组窗口。
- 单击显示用户按钮以显示用户列表。
- 选择匿名用户的名称（例如，IUSR_<hostname>）。
- 按添加按钮以添加匿名用户。
- 从访问类型列表中，滚动到不访问并选择它。
- 按确定按钮以返回到“目录权限”窗口并选择替换对子目录的权限复选框。

11. 按**确定**按钮。Windows 将更新 *InetPub* 目录下的整个目录树，这将花费几分钟的时间。

为 NTLM 创建名为 **admin** 的 Windows 本地用户

如果在安装 DB2 Alphablox 的过程中添加了 Microsoft IIS 管理帐户，则可以跳过此步骤，并使用该管理用户而不是在本地机器上创建 *admin* 帐户。

可信 NTLM 安全性要求为 DB2 Alphablox *admin* 用户创建一个 Windows 帐户。缺省情况下，当对 Microsoft IIS 上的 NTLM 进行认证时，用户名与 Web 服务器的域（通常是网络域）中的用户名对应。

在大多数情况下，DB2 Alphablox *admin* 用户不是网络域中的用户。而是，将 *admin* 用户添加到 IIS 主机的本地域中。在 IIS 下作为 *admin* 进行认证时，用户名的格式为：

machine-name\admin

在 Tomcat 5.5 中为 Alphablox 8.4.1 配置 NTLM 安全性

1. 打开 Tomcat 5.5 目录中的 *conf/server.xml* 文件
2. 找到 AJP/13 连接器的定义，它看起来如下所示：

```
<Connector port="8009"
  enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
```

3. 将 *tomcatAuthentication* 属性添加至 Connector 标记并将它设置为 *false*：

```
<Connector port="8009" tomcatAuthentication="false"
  enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
```

4. 通过注释掉现有 *Realm* 标记并添加新的 *Realm* 标记来替换现有标记，以使用 DB2 Alphablox 的 *AlphabloxNTLMRealm*：

```
<!--
  <Realm className="org.apache.catalina.realm.JAASRealm" debug="10"
    appName="Alphablox"
    userClassNames="com.alphablox.security.jaas.UserPrincipal"
    roleClassNames="com.alphablox.security.jaas.RolePrincipal"
    useContextClassLoader="false" />
-->
  <Realm className="com.alphablox.security.AlphabloxNTLMRealm" />
```

登录到 DB2 Alphablox

当作为 *admin* 用户通过 Microsoft IIS 连接到 DB2 Alphablox 时，必须：

- 使用具有正确格式的登录名。

如果 IIS 机器登录到与认证 *admin* 用户的客户机不同的域中，则必须传递客户机域名和用户名。例如，如果 *admin* 用户正在从 WEBDEV1 域中的客户机登录，则登录名的格式将变为：

WEBDEV1\admin

- 确保正确地设置了浏览器认证。

缺省情况下，Microsoft Internet Explorer Web 浏览器使用用户的 Windows 用户名和密码来对 NTLM 认证这些用户。要作为另一个用户（例如，*admin*）登录到 Microsoft IIS，需要更改此缺省行为。（因为 *admin* 用户名和密码不会与此用户名和密码相匹配，所以登录尝试将失败。）

完成下列步骤以迫使 Microsoft Internet Explorer 浏览器在认证期间提示输入用户名和密码:

1. 在 Internet Explorer 中, 从工具菜单中, 选择 **Internet 选项**。
2. 选择**安全选项卡**。
3. 如果是通过内部网 (LAN) 使用应用程序的, 则请选择**本地区域内部网**。
4. 按**自定义级别按钮**以设置安全级别。
5. 在**用户认证部分**的**登录部分**中, 选择**提示输入用户名和密码按钮**。
6. 单击**确定按钮**以关闭“安全设置”窗口。
7. 单击**确定按钮**以关闭“Internet 选项”窗口。

Windows Integrated authentication 意味着浏览器从服务器接收到一个协商头, 此协商头指定是使用 Kerberos 认证还是使用 Windows Integrated authentication (以前称为 NTLM 或 NT Challenge/Response)。要跳过协商过程, 请执行下列步骤:

1. 停止 DB2 Alphablox 并打开命令提示符, 然后浏览到 Inetpub\AdminScripts 目录。
2. 输入以下命令:

```
cscript adsutil.vbs set w3svc/NTAuthenticationProviders "NTLM"
```

3. 要检查认证, 请输入:

```
cscript adsutil.vbs get w3svc/NTAuthenticationProviders
```

缺省值是 Negotiate, NTLM。在设置为“仅 NTLM”之后, 它应该显示“NTLM”。

4. 输入以下命令: `iisreset computerName /RESTART`。这将重新启动 IIS。
5. 重新启动 DB2 Alphablox, 您应该能够通过 IIS 登录到 DB2 Alphablox 中。

配置 DB2 Alphablox 以使用基于 Web 服务器的安全性

无论是否使用 Web 服务器, 为了方便和安全起见, 请您务必查看下列 DB2 Alphablox 设置:

- 『配置自动生成用户帐户』
- 第 63 页的『过滤 IP 地址』
- 第 63 页的『设置目录权限』
- 第 63 页的『禁用目录浏览』

配置自动生成用户帐户

在 Web 服务器响应之前, 即使在 Microsoft IIS 上成功认证了用户, 该用户也必须具有 DB2 Alphablox 用户标识才能发出请求。DB2 Alphablox 可以自动生成用户帐户, 而不是必须为 IIS 或 iPlanet 已知的每个用户创建用户帐户。

要配置 DB2 Alphablox 以自动创建用户标识:

1. 作为 *admin* 用户或作为隶属于 administrators 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理选项卡**。“常规”页面打开。
3. 在“一般属性”部分中, 单击**系统链接**。
4. 选中**自动生成用户概要文件复选框**。
5. 按**保存按钮**。

以后，当新用户尝试登录时，DB2 Alphasbx 将提示输入用户标识和密码，并自动创建该用户的帐户。如果您正在将 Microsoft IIS 用作 Web 服务器，并且要对 DB2 Alphasbx 使用 Windows 认证，则必须执行第 58 页的『为 IIS NTLM 设置 Microsoft 安全选项』中描述的附加配置工作。

限制: 在使用 IIS NTLM 安全性时，如果需要使用 Mozilla Firefox 浏览器来管理 DB2 Alphasbx，则第一次登录时一定要使用 Mozilla Firefox 浏览器来登录。使用 Mozilla Firefox 浏览器登录将为该用户创建一个明文密码；否则，使用 Mozilla Firefox 浏览器进行的认证将失败。

要点: 对 IP 地址进行过滤（如下一节所述）可以避免可能有用户恶意地绕过 Web 服务器安全性。

过滤 IP 地址

在 DB2 Alphasbx 8.4（而不是 8.4.1）中，可以将 DB2 Alphasbx 配置为仅接受来自指定客户机的请求。在基于 Web 服务器的安全性下，DB2 Alphasbx 应该仅接受来自 Web 服务器主机（即，Web 服务器本身或 Web 服务器主机的管理员）的请求。在 DB2 Alphasbx 主页上，选择**管理**选项卡，打开**一般属性**页面并单击**系统**链接，可使用**已授权的客户机列表**属性来创建可接受的 DNS 名称列表或 IP 地址列表。如果您只想接受来自本地机器的连接，则将**已授权的客户机列表**设置为 *localhost*。

在 DB2 Alphasbx 8.4.1 中，可直接在 Tomcat 5.5 中指定授权客户机。Apache Tomcat 5.5 允许您指定“远程主机以标识准许使用或拒绝使用的 IP 地址或主机名。例如，要阻止所有来自除 local host 之外的主机名的请求：

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="127.0.0.1" />
```

要了解有关“远程主机过滤器”和“远程地址过滤器”的更多信息，请访问 <http://tomcat.apache.org/tomcat-5.5-doc/config/valve.html>。

设置目录权限

当将 DB2 Alphasbx 与 Microsoft IIS 配合使用时，应该防止用户访问 Web 服务器 docroot 或 DB2 Alphasbx 的 webapps 目录下的受限文件。要了解有关在 IIS 中设置目录权限的详细信息，请参阅 IIS 文档。

禁用目录浏览

因特网浏览器允许用户任意访问 Web 服务器上的目录。为了防止用户在不了解准确路径和文件名的情况下以此方式找到并访问文件，必须在 Web 服务器上禁用目录浏览。请参阅 Web 服务器的文档。

第 12 章 扩展 DB2 Alphablox

本章描述如何通过定制计算、DHTML 客户机和可扩展用户管理器中的定制来扩展 DB2 Alphablox 的内置功能。

概述

虽然 DB2 Alphablox 提供了许多功能来支持大部分分析应用程序的需求，但是开发者也可以扩展平台以使用定制计算、用户管理器和 DHTML 客户机。当包括的标准计算 API 不足够时，定制计算可能是必需的。对用户管理器进行的扩展允许您对认证、授权和个性化进行定制。也可以定制 DHTML 客户机以处理您的独特需求。下列各节提供了有关计算扩展、用户管理器扩展和 DHTML 客户机扩展的简要描述。并且，还提供了有关配置 DB2 Alphablox 以处理您所作的定制的信息。

计算扩展

DB2 Alphablox 提供了内置的计算 API 来支持大部分分析应用程序的标准计算需求。但是，如果您发现附带的功能无法满足您的需求，则可以通过执行下列步骤来创建自己的定制计算扩展：

1. 编写计算扩展以扩展 Function 类。

- a. 实现以下方法：

```
public double getResult(double[] variables)
```

getResult 方法返回对数字列表变量执行用户定义的计算所得到的结果。

- b. 确保此文件属于以下包：

```
com.alphablox.util.calculator
```

- c. 文件名的首字母必须是大写的，其余字母是小写的。

2. 编译此文件。

3. 将此文件添加到 JAR 文件中。

4. 将该 JAR 文件放在以下目录中：

```
<alphabloxDirectory>\lib
```

5. 修改启动文件以在类路径中包括这个 JAR。要了解有关如何正确地设置此类路径的详细信息，请参阅第 66 页的『设置类路径』。

可以在《DHTML 客户机开发者指南》的『DataBlox 参考』一节中找到关于定制计算的更多信息。

用户管理器扩展

可以扩展 DB2 Alphablox 用户管理器以支持无法使用现成 DB2 Alphablox 功能满足的定制用户认证、授权和个性化需求。要了解有关可用于扩展 DB2 Alphablox 用户管理器的 Java 接口和服务器命令的信息，请参阅第 86 页的『可扩展用户管理器』。要了解有关定义类路径以访问定制 Java 类的信息，请参阅第 66 页的『设置类路径』。

DHTML 客户机扩展

DB2 Alphablox DHTML 客户机为用户提供了丰富的现成功能。对于许多应用程序页面，此功能可能已足够。但是，DHTML 客户机背后的 Blox UI 模型使开发者能够在他们的页面上添加定制、使用定制 JavaBeans 组件以及创建定制 JSP 标记库以进一步增强分析应用程序。

要了解有关使用 Blox UI 模型来扩展 DB2 Alphablox 能力的详细信息，请参阅《DHTML 客户机开发者指南》和 *Developer's Reference for the DHTML Client*。

配置 DB2 Alphablox 以支持定制 Java 类

如前所述，可以使用定制 Java 类来创建对 DHTML 客户机、计算能力和用户管理器所作的定制。当创建和使用定制类时，您应该了解下列注意事项：

- 可以将定制类（这些类通常封装在 JAR 文件中）存储在服务器上您选择的任何位置中。

要点： 如果将 Java 类存储在 DB2 Alphablox 安装目录中，则升级期间可能会删除掉您的文件。因此，Alphablox 建议将这些类存储在 DB2 Alphablox 目录外部的目录中或者将它们包括在应用程序的目录中：

```
<alphabloxDirectory>/webapps/<applicationDirectory>
```

- 在 DB2 Alphablox 升级期间，对 DB2 Alphablox 启动批处理文件（下一节（第 66 页的『设置类路径』）对此文件作了描述）所作的修改将丢失。

设置类路径

要让 DB2 Alphablox 使用您的定制类，需要将类目录添加到启动批处理文件中指定的 DB2 Alphablox 类路径中。根据您所使用的应用程序服务器来执行下列步骤。

对于 Apache Tomcat 实现：

1. 在文本编辑器中，打开 <db2alphablox_dir>/appserver/bin/aas.bat 文件（Windows 平台）或 <db2alphablox_dir>/appserver/bin/aas.sh 文件（Linux 和 UNIX 平台），其中 <db2alphablox_dir> 是 DB2 Alphablox 的安装目录。
2. 查找设置类路径的行并追加使用同一语法的一行以指向类的位置。例如，如果类位于 c:/myclasses/classes 目录中：

```
set CLASSPATH=%CLASSPATH%;%AAS_LIB%\xerces.jar
// Add your class path below
set CLASSPATH=%CLASSPATH%; c:/myclasses/classes
```

如果正在将 DB2 Alphablox 作为 Windows 服务运行，则还需要在服务参数文件 <db2alphablox_dir>\appserver\conf\wrapper.properties 中设置类路径：

```
wrapper.class_path=$WRAPPER_AAS_HOME$\lib\aasserver.jar
// Add your class path below
wrapper.class_path=c:/myclasses/classes
```

3. 保存更改。
4. 应重新启动 DB2 Alphablox 以使更改生效。

对于 WebLogic 应用程序服务器，请执行下列步骤：

1. 在文本编辑器中，打开 <db2alphablox_dir>/bin/aasetup.bat (Windows 平台) 或 <db2alphablox_dir>/bin/aasetup.sh (Linux 和 UNIX 平台)
2. 查找设置类路径的行并追加使用同一语法的一行以指向类的位置。例如，如果类位于 c:/myclasses/classes 目录中：

```
set CLASSPATH=%CLASSPATH%;%AAS_CP%  
// Add your class path below  
set CLASSPATH=%CLASSPATH%; c:/myclasses/classes
```

如果正在将 DB2 Alphablox 作为 Windows 服务运行，则还需要在服务参数文件 <db2alphablox_dir>\bin\aasetup_nt_service.bat 中设置类路径

3. 重新启动 WebLogic。如果 DB2 Alphablox 正在作为 Windows 服务运行，则再次运行 <BEA>/weblogic700/server/bin 下的 installSvc.cmd。

如果使用的是 WebSphere，则必须创建一个包含您的类（不能有松散类）的 JAR 文件，然后将该 JAR 文件放在 <websphere_dir>/AppServer/lib/ext 中。重新启动 WebSphere 以使更改生效。

第 13 章 配置 DB2 Alphablox 属性

DB2 Alphablox 主页的**管理**选项卡下的**常规**链接提供了一些链接，这些链接指向一组用于执行公共管理任务的 Web 页面。注意，也可以使用控制台命令来执行所有这些任务以及其他几项任务，如第 123 页的第 19 章，『DB2 Alphablox 控制台命令』所述。要访问 DB2 Alphablox，请在浏览器中输入以下 URL：

```
http://<servername>/AlphabloxAdmin/home
```

其中 <servername> 表示服务器的名称以及运行 DB2 Alphablox 时使用的端口号。本章阐述定义 DB2 Alphablox 运行属性的过程并描述了 DB2 Alphablox 日志。

DB2 Alphablox 管理任务

管理选项卡的**常规**页面上的一般属性部分提供了一些链接，这些链接指向用于执行下列与 DB2 Alphablox 相关的任务的界面：

- 『配置启动属性』
- 第 70 页的『配置系统属性』
- 第 71 页的『指定 Telnet 端口』
- 第 72 页的『配置 DB2 Alphablox 立方体管理器』

配置启动属性

最初的安装和配置过程确定了**启动**页面上的属性值。这一组属性表示用于启动 DB2 Alphablox 的绝对最低要求。

注：在部署应用程序之前，请复审并确定其他 DB2 Alphablox 属性的正确值，如第 70 页的『配置系统属性』所述。

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 在该页面的一般属性部分中，单击**启动**链接以查看和修改 DB2 Alphablox 启动属性：

在**启动**页面上，可以提供下列启动属性值：

属性名	缺省值	描述
实例名	AlphabloxAnalytics	这个 DB2 Alphablox 实例的名称。此名称是在安装期间确定的，除非重新安装 DB2 Alphablox，否则不能更改此名称。 注： 这不是 DB2 Alphablox 机器的域名（DNS）。
日志文件名	Server.log	这个 DB2 Alphablox 会话的日志文件名。 日志文件位于 <repository>/servers/<instanceName>/logs 目录中。
命令文件名	abc.console	可选命令文件的名称。DB2 Alphablox 在启动期间将查找具有此名称的文件并从中读取一系列 DB2 Alphablox 命令。此文件中的命令与通过 DB2 Alphablox 控制台输入的那些命令使用相同的语法。

属性名	缺省值	描述
缺省消息级别	INFO	要显示（以及写入日志文件）的消息的最低（最不严重）级别。有效值（按严重性递增顺序排列）包括：DEBUG、VERBOSE、INFO、SYSTEM、WARNING、ERROR 和 FATAL。要获取消息级别的描述，请参阅第 129 页的『消息级别』。
启用 DB2 是 Alphasblox 日志		是否允许 DB2 Alphasblox 将消息写入日志。
DB2 Alphasblox 空闲持续时间	15 分钟	指定 DB2 Alphasblox 在暂挂之前的不活动分钟数。暂挂是一种状态，在此状态下，DB2 Alphasblox 使用较少的资源。任何请求都将自动地使服务器活动继续。
DB2 Alphasblox 控制台缺省消息级别	FATAL	要发送到 System.out 的最低（最不严重的）消息级别（对于 DB2 Alphasblox Apache Tomcat 实现，输出将在 Tomcat 控制台上显示）。有效值（按严重性递增顺序排列）包括：DEBUG、VERBOSE、INFO、SYSTEM、WARNING、ERROR 和 FATAL。
装入所有用户对象	启用	将所有用户对象装入内存。缺省情况下，用户管理器将用户对象存储在 DB2 Alphasblox 存储库中。当用户数目较多时，如果将此属性与“系统”属性中的“保持用户处于已装入状态（以分钟计）”设置配合使用，就可以使内存管理效率更高并且用户管理器启动速度更快。

4. 输入这些属性的必需值。
5. 单击**保存**按钮以应用更改。

注：必须停止并重新启动 DB2 Alphasblox 才能使这些更改生效。

配置系统属性

在初始安装之后并在部署应用程序之前，请查看并编辑 DB2 Alphasblox 的系统属性。

1. 作为 *admin* 用户或作为隶属于 administrators 组的用户登录到 DB2 Alphasblox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 在该页面的一般属性部分中，单击**系统**链接以查看和 / 或修改 DB2 Alphasblox 系统属性：

在系统属性页面中，可以提供下列属性值：

属性名	缺省值	描述
Web 服务器前缀	URL <空>	正确的值取决于所使用的 Web 服务器。 安装后，在没有预先与 DB2 Alphasblox 客户支持联系的情况下，请不要更改此值。
新建日志启动消息级别	INFO	确定导致写入新日志文件的最不严重消息级别。

属性名	缺省值	描述
新建日志结束消息级别	FATAL	确定导致写入新日志文件的最严重消息级别。对此属性和新建日志启动消息级别设置相同的值（例如，ERROR）将创建仅包含指定级别的消息的日志。
启用认证	是	设置用户在访问 DB2 Alphablox 资源和 DB2 Alphablox 应用程序之前是否必须经过认证。当值为“否”时，所有用户都作为 Guest 登录，并且不需要密码。
自动生成用户概要文件	否	确定当用户登录到 DB2 Alphablox 时是否自动创建用户概要文件。当设置为是时，将导致任何用户请求都被接受（而不考虑密码），并且该用户将被授予 DB2 Alphablox 访问权。
已授权的客户机列表	<空>	列出有权访问 DB2 Alphablox 的用户。值是 IP 地址和 / 或主机名的逗号分隔列表；不允许通配符。空列表指示允许所有用户访问 DB2 Alphablox。 注： 当使用 Web 服务器安全性时，设置授权客户机以便只有 Web 服务器才能访问 DB2 Alphablox 十分重要。
消息历史记录大小	100	设置保存在消息历史记录区域中以供您在控制台中查看的消息数目。当该区域变满时，DB2 Alphablox 将回绕到开头并覆盖最旧的消息（那些消息仍保存在日志文件中）。
退出时保存	是	设置会话结束时是否保存 DB2 Alphablox 的当前状态（及其所有属性）。如果管理员或组装人员在会话期间进行了更改但却未能保存更改，则保留此值为“是”可以确保不会丢失那些更改。
缺省 HTML 客户机主题	coleman	此属性设置以 HTML 格式显示应用程序页面时使用的缺省主题。（要了解更多信息，请参阅《开发者指南》。）
SMTP 服务器	安装期间指定的	设置 SMTP 服务器的名称，以便使用 JavaMail 来发送邮件的应用程序可以成功地工作。请将此值设置为有效 SMTP 邮件服务器的名称。
保持用户处于已装入状态（以分钟计）	-1	设置在最后一个用户会话实例完成后在内存中保留用户对象的分钟数。缺省值 -1 无限期地保持用户处于已装入状态（当 DB2 Alphablox 正在运行时）。当用户数目较多时，将此属性与“启动”属性中的“装入所有用户对象”设置配合使用，可以使内存管理效率更高并且用户管理器启动速度更快。

1. 输入这些属性的必需值。
2. 单击**保存**按钮以应用更改。

注：必须停止并重新启动 DB2 Alphablox 才能使属性更改生效。

指定 Telnet 端口

执行下列操作以指定 DB2 Alphablox 的侦听端口：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。

- 单击**管理**选项卡。**常规**页面显示。
- 在该页面的一般属性部分中，单击 **Telnet 控制台** 链接以查看和修改 DB2 Alphablox 端口。

在 **Telnet 控制台** 页面中，可以修改下列属性值：

属性名	缺省值	描述
Telnet 控制台端口	23	DB2 Alphablox 控制台的 Telnet 版本在工作时使用的端口。 如果缺省 Telnet 设备是 DB2 Alphablox，则访问 Telnet 控制台时不必输入端口号。将此值设置为 0（零）以使 Telnet 无法访问 DB2 Alphablox 控制台。
Telnet 用户名	admin	可以运行 Telnet 会话的用户的名称。
Telnet 密码	password	Telnet 用户名的密码。
确认 Telnet 密码		当您在 Telnet 密码 文本框中输入了密码时，必须在此文本框中确认该密码。
Telnet 超时	15 分钟	指定对 DB2 Alphablox 进行的 Telnet 会话在超时之前需经过的分钟数。当 Telnet 会话超时时，该会话将终止。
重新启动控制台管理器	未选取	当选取了此属性时，您按下 保存 按钮时将重新启动控制台管理器。

- 输入必需的端口号。
- 单击**保存**按钮以应用更改。

注：必须停止并重新启动相应的服务管理器才能使这些更改生效。要重新启动服务管理器，请选取已更改的端口下面的框并单击**保存**按钮。

配置 DB2 Alphablox 立方体管理器

单击 **DB2 Alphablox 立方体管理器** 链接将打开一个页面，该页面允许管理员对可以装入到服务器中的 DB2 Alphablox 立方体数目进行限制。每个 DB2 Alphablox 立方体都有可能使用大量的系统资源，因此管理员可能想根据运行 DB2 Alphablox 的机器上的可用内存量来限制立方体数。

要限制系统上的 DB2 Alphablox 立方体数：

- 作为 *admin* 用户或作为隶属于 administrators 组的用户登录到 DB2 Alphablox 主页。
- 单击**管理**选项卡。**常规**页面显示。
- 在该页面的一般属性部分中，单击 **DB2 Alphablox 立方体管理器** 链接以查看和/或修改 DB2 Alphablox 系统属性：
- 选取**最大立方体数**框。
- 在**最大立方体数**文本框中输入 DB2 Alphablox 立方体的最大数目。
- 单击**保存**按钮以应用更改。

要了解有关使用 DB2 Alphablox Cube Server 来创建和管理 DB2 Alphablox 立方体的详细信息，请参阅《*DB2 Alphablox Cube Server 管理员指南*》。

定制属性定义

可以使用定制属性来使用户登录详细信息与 DB2 Alphablox 用户概要文件相关联。要了解有关用户概要文件的详细信息，请参阅第 45 页的第 8 章，『用户定义』。DB2 Alphablox 提供了一个示例定制属性文件来使您可以更容易地完成此工作。

属性的有效值以及它的缺省值都是从属性定义中获取的。开发者可以使用程序来处理用户属性，例如，使用 RepositoryBlox `getUserProperty()` 方法来处理用户属性。

注：当定义属性值需要空格的定制属性时，请使用 ` `；而不是输入空格。无法识别输入的空格，并将从属性值中除去空格。

定义新的用户属性

执行下列操作以定义新的用户属性：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。常规页面显示。
3. 在该页面的**定制属性**部分中，单击**用户定义**链接。用户定义页面显示。
4. 单击**创建**按钮。创建用户定制属性页面显示。
5. 在**属性名**文本框中提供名称（必需），并且在**显示标注**文本框中提供描述。在通过**管理**选项卡的**应用程序**链接显示的页面上的应用程序定义中，将显示此描述。

注：如果您正在使用基于 LDAP 的用户管理器或其他具有自己的编辑器的外部用户管理器（如 NTLM），则需要在**属性名**下的“外部存储库”和“本地存储库”两者之间进行选择。单击“外部存储库”单选按钮并从下拉列表中进行选择以便让 DB2 Alphablox 装入定制属性。

6. 如果要让该属性接受多个值，则单击**多项选择**按钮。
7. 从下拉列表中选择**用户访问权**级别。（要定义、更改或删除应用程序属性，管理特权总是必需的。）
 - **隐藏：**该属性不会显示在“新建用户和组”、“更改用户和组”以及“管理用户概要文件”页面上。只能在应用程序中（例如，通过 JavaScript）或通过本页面更改它的值。
 - **可视：**该属性及其缺省值将显示在“新建用户和组”、“更改用户和组”以及“管理用户概要文件”页面上，但是用户无法更改该值。
 - **编辑：**该属性及其缺省值将显示在“新建用户和组”、“更改用户和组”以及“管理用户概要文件”页面上，并且可以被更改。如果该属性具有多个有效值，则用户可以从列表中选择值。如果该属性不具有有效值列表，则用户可以输入值。
8. 输入该属性的**缺省值**（如果有的话）。
9. 在**值列表**文本框中，输入可选的用逗号分隔的有效属性值字符串。此处输入的值将用来填充用户可以从中进行选择的列表。如果未输入任何值，并且如果将**用户访问权**设置为**编辑**，则相应的页面上将显示一个文本框，用户可以在该文本框中输入值。
10. 单击**保存**按钮以保存新属性并返回到**常规**页面中。

更改用户属性

执行下列操作以更改用户属性：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 在该页面的**定制属性**部分中，单击**用户定义**链接。**用户定义**页面显示。
4. 选择要更改的定义并单击**编辑**按钮。**编辑用户定制属性**页面显示，并且它的当前值显示在输入字段中。
5. 进行必需的更改。
6. 单击**保存**按钮以进行更改并返回到**常规**页面中。

删除用户属性

执行下列操作以删除用户属性：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 在该页面的**定制属性**部分中，单击**用户定义**链接。**用户定义**页面显示。
4. 从现有属性的列表中选择要删除的属性定义。
5. 单击**删除**按钮以删除该属性并返回到**常规**页面中。

注：删除属性定义的操作是永久性的；无法恢复已删除的属性定义。

定义新的定制应用程序属性

执行下列操作以定义定制应用程序定义：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 在该页面的**定制属性**部分中，单击**应用程序定义**链接。**应用程序定义**页面显示。
4. 单击**创建**按钮。将显示**创建应用程序定制属性**页面。
5. 在**属性名**文本框中提供名称（必需），并且在**显示标注**文本框中提供描述。在通过**管理**选项卡的**应用程序**链接显示的页面上的应用程序定义中，将显示此描述。
6. 如果要想该属性接受多个值，则单击**多项选择**按钮。
7. 从下拉列表中选择**用户访问权级别**。（要定义、更改或删除应用程序属性，管理特权总是必需的。）
 - **隐藏：**该属性不会显示在它的相应页面上。只能在应用程序中（例如，通过 JavaScript）或通过本页面更改它的值。
 - **可视：**该属性及其缺省值将显示在它的相应页面上，但是用户无法更改该值。
 - **编辑：**该属性及其缺省值将显示在它的相应页面上，并且可以被更改。如果该属性具有多个有效值，则用户可以从列表中选择所需的值。如果该属性不具有有效值列表，则用户可以输入值。
8. 输入该属性的**缺省值**（如果有的话）。

9. 在**值列表**文本框中，输入可选的用逗号分隔的有效属性值字符串。此处输入的值将用来填充用户可以从中进行选择的列表。如果未输入任何值，并且如果将**用户访问权**设置为**编辑**，则相应的页面上将显示一个文本框，用户可以在该文本框中输入值。
10. 单击**保存按钮**以保存新属性并返回到**常规**页面中。

更改应用程序属性

执行下列操作以更改应用程序属性定义：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 在该页面的**定制属性**部分中，单击**应用程序定义**链接。**应用程序定义**页面显示。
4. 选择要更改的定义并单击**编辑**按钮。**编辑应用程序定制属性**页面显示，并且它的当前值显示在输入字段中。
5. 进行必需的更改。
6. 单击**保存按钮**以进行更改并返回到**常规**页面中。

删除应用程序属性

执行下列操作以删除应用程序属性定义：

1. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
2. 单击**管理**选项卡。**常规**页面显示。
3. 在该页面的**定制属性**部分中，单击**应用程序定义**链接。**应用程序定义**页面显示。
4. 从现有属性的列表中选择要删除的属性定义。
5. 单击**删除**按钮以删除该属性并返回到**常规**页面中。

注：删除属性定义的操作是永久性的；无法恢复已删除的属性定义。

注：在此图中，*UserPropDesc.properties* 文件的内容导致**允许用户编辑概要文件**下拉列表下的所有内容都显示。

创建和管理注释集合

可以使用 DB2 Alphablox 来管理要在多维网格中的数据单元格中提供的用户注释。下列各节描述了如何使用 DB2 Alphablox 的“管理”选项卡下的“注释管理”页面来配置“注释集合”以便与特定的数据源配合使用。在受支持的 Web 浏览器上，可在 DHTML 客户机中查看注释。

注：注释集合要求指向附加配置步骤，以使用 Microsoft JDBC Driver for SQL Server，它是 WebSphere 中包括的选项。请与 DB2 Alphablox 客户支持联系以获取帮助。

访问“注释管理”对话框

要访问“注释管理”页面（该页面将在一个新窗口中打开），请执行下列步骤：

1. 打开 DB2 Alphablox 主页。
2. 单击“管理”选项卡。
3. 在左边的导航菜单中的“运行时管理”下面，单击“注释管理”链接。

4. “注释管理”对话框应该显示在一个新的浏览器窗口中。

注：要使用“注释管理”对话框，您需要具有创建和废弃关系表的权限。要在开发定制注释应用程序时使用 CommentsBlox API，可能需要具有选择、插入、更新、删除、创建和废弃表的权限。

定义和访问数据源

要定义用于存储注释集合的数据源，请执行下列步骤：

1. 在“数据源”标题下，单击“名称”选择列表并选择已定义的用来存储注释集合的 DB2 Alphablox 数据源。[注意：要了解更多有关在 DB2 Alphablox 中定义数据源的信息，请参阅『数据源定义』。]
2. 在用户名和密码输入字段中，输入有权在数据源中创建表的用户的登录名和密码。[注意：如果这些字段为空白，则在连接后将自动提交缺省数据源用户名和密码（如果在 DB2 Alphablox 数据源定义中定义了缺省数据源用户名和密码的话）。]
3. 按“连接”按钮。如果已定义的数据源和用户信息有效，则将建立与该数据源的连接，并且“注释管理”对话框上的“集合”部分将变为处于启用状态。

定义注释集合

在用户可以开始将注释添加至网格中的数据单元格之前，您必须首先创建注释集合以存储相关的注释。

注：只要注释集合未包含任何注释，就可以对该集合添加维或除去维。

1. 要创建新的注释集合，请执行下列步骤：
2. 单击“集合”列表框下面的“创建”按钮。在浏览器窗口的右边，将显示三个新的部分。
3. 在“名称”部分中，输入注释集合的名称。[注意：名称一定不能包含空格，并且必须符合用于存储注释的数据库表的命名要求。]
4. 在“字段”部分中，可以将字段添加到注释集合中或者从注释集合中删除字段。缺省情况下，包括了三个必需的字段：Author、Timestamp 和 CommentText 字段。可以添加其他字段，即定义它们的名称，如果您愿意的话，还可以提供所定义的每个字段的描述。还可以添加可选的 CellValue 字段。CellValue 字段将导致自动地把所选单元格的当前单元格值添加到与该单元格相关联的注释中。
5. 在“维”部分中，必须从已定义的 DB2 Alphablox 数据源列表中选择多维数据源。所选立方体中的维列表应该会显示在所列出的数据源下方的文本框中。
6. 选择要让用户能够对哪些维添加注释。要选择多个维，必须按下 Ctrl 键并单击其他每个维。

注：选择的维将影响用户添加的注释的作用域，如以下两个示例所示：

- 示例 1: 如果一个立方体有 3 个维 (Year、Product 和 Region)，并且在“维”列表中只选择了其中两个维 (Product 和 Year)，则添加到 Region 维 (在“维”列表中未选择这个维) 的任何成员的注释都将显示在所有其他 Region 成员中。
 - 示例 2: 如果一个立方体有 3 个维 (Year、Product 和 Region)，并且在“维”列表中选择了全部三个维，则注释将仅显示在注释被添加到的单元格中。
7. 按“保存”按钮。这时，新定义的集合的名称将显示在“集合”列表中。

使用 Microsoft SQL Server 或 Sybase 数据库时的注释集合

如果您计划使用 Microsoft SQL Server 或 Sybase 数据库来存储可能包含非 ASCII 字符（这些字符在非英语语言中很常见）的注释，则需要执行下列步骤：

1. 对于定义在每个注释集合，需要将自动生成的表中的列的数据类型由 `varchar` 更改为 `nvarchar`。

下列五个表由所有已定义的使用数据源的注释集合所共享：

- Collections: `CollectionName`
- Comments: `CommentText`
- Context: `ContextName`
- Dimensions: `DimName`
- FieldDefinitions: `descr` 和 `FieldName`
- FieldValues: `fieldvalue`

并且，对于每个注释集合，将创建一个唯一的表：

- `ABXMBRT_<collectionName>`

其中 `<collectionName>` 是您对注释集合指定的名称。您需要将下列各列的数据类型更改为 `nvarchar`：

- `ABXMBRT_<collectionName>`: `MemberName`
2. 在应用程序中的所有 JSP 页面上，应该在每个页面的 JSP page 伪指令中指定 Unicode 字符集，如下所示：

```
<%@ page contentType="text/html; charset=UTF-8" %>
```

显示注释集合定义

要显示现有集合的定义，请执行下列步骤：

1. 在连接到已定义的数据源之后，预定义的注释集合列表将显示在“集合”文本框中，并且“创建”、“显示”和“删除”按钮处于启用状态。如果尚未连接到已定义的数据源，则首先执行上面的『定义和访问数据源』中的步骤来连接至数据源。
2. 单击“集合”文本框中列出的其中一个注释集合，然后按“显示”按钮。集合定义应该会显示在该窗口的右半部分中。

删除注释集合

要除去已定义的注释集合，请执行下列步骤：

1. 连接至已定义的数据源（请参阅上面的『定义和访问数据源』）。
2. 在“集合”文本框中，单击要除去的注释集合的名称。
3. 按“删除”按钮。应该会从列表中除去该集合名。

添加和显示注释

在定义了注释集合之后，用户就可以将注释添加到已启用注释的网格的数据单元格中。当用户右键单击某个已启用注释的数据单元格时，将显示一个菜单。在“注释”子菜单中，用户可以选择“添加注释”或“显示注释”。

要了解有关 CommentsBlox 的更多信息，请参阅 *Developer's Reference* 中的 CommentsBlox 一节。

创建远程 PDF 处理器

为了提高性能、进行内存管理或者让多台 DB2 Alphablox 主机共享 PDF 处理，您可能会决定在远程专用服务器上运行 PDF 引擎。

配置远程 PDF 处理器

要使用远程 PDF 处理器而不是内部服务器：

1. 在远程服务器上创建名为 *AlphabloxPDFRemote* 的根目录。
2. 在此根目录中创建两个新目录：*lib* 和 *repository*。
3. 在 *repository* 目录中创建两个子目录：*font* 和 *theme*。
4. 将下列 JAR 文件从 DB2 Alphablox 安装目录的 *db2alphablox_dir/lib/* 目录复制到远程服务器的 *lib* 目录：
 - *bforeport.jar*
 - *pdfserver.jar*
 - *xalan.jar*
 - *xercesImpl.jar*
 - *xml-apis.jar*
 - *icu4j_3_4_1.jar*
5. 将所有文件从 *alphablox db2alphablox_dir/repository/theme* 目录复制到远程服务器上的 *theme (repository/theme)* 目录中。
6. 将所有文件从 *alphablox db2alphablox_dir/repository/font* 目录复制到远程服务器上的 *font (repository/font)* 目录中。
7. 将 *StartPDFRemoteServer* 脚本文件（用于 Windows 系统的 *StartPDFReportServer.bat* 或用于 Linux 和 UNIX 系统的 *StartPDFReportServer.sh*）从 DB2 Alphablox 安装目录的 *db2alphablox_dir/bin/* 目录复制到远程服务器上的 *AlphabloxPDFRemote* 目录。
8. 对 *StartPDFRemoteServer* 脚本文件做如下更改：
 - 更改 *LIB* 设置，使其指向远程服务器 *lib* 目录。
 - 确保使 *JAVA* 设置指向有效的 *JRE* 或 *JDK*。
 - 根据需要，更改 *LISTEN_PORT* 设置。
 - 确保在当前目录中启动脚本（缺省情况下，脚本文件会建立一个目录）。
9. 运行该批处理文件。现在，远程 PDF 处理器应该可用。
10. 要将远程 PDF 处理器与 DB2 Alphablox 应用程序配合使用，请执行下一节中的步骤来将 DB2 Alphablox 配置为使用远程服务器。

配置远程 PDF 报告管理

PDF 报告处理是通过使用 DB2 Alphablox 管理页面中的“PDF 报告运行时管理”设置来管理的。要配置远程服务器，请执行下列步骤：

1. 打开 DB2 Alphablox 管理页面，然后单击“管理”选项卡。

2. 在该页面左边的“运行时管理”下面，单击“PDF 报告 (DHTML)”链接以打开“PDF 报告”对话框。
3. 要配置远程 PDF 服务器，请进行下列更改：
 - a. 单击“编辑”按钮。
 - b. 将“服务器”设置由“使用内部服务器”更改为“使用 PDF Report Server”。
 - c. 定义远程服务器的“主机名”和“主机端口”。
 - d. 如果您愿意的话，可以选择“压缩”选项，也可以让它设置为缺省值“不压缩”。这些选项基于 JDK 中定义的压缩设置。实际上，低设置与高设置的压缩率差别并不大。低设置能够有效地压缩文件，而最高设置所能够提供的压缩率仅比最低设置略高。
 - e. 保存更改。如果您已成功配置了远程 PDF 服务器，则“状态”应该会指示“可用”。

DB2 Alphablox 日志文件

DB2 Alphablox 将各种 DB2 Alphablox 事件和错误记录到日志文件中。被写至 DB2 Alphablox 控制台的每条消息也都被写入日志文件。日志文件将保留到它们被删除或移动为止，这有助于保留有关 DB2 Alphablox 的活动历史记录。此历史记录可以用于管理审计目的，也可以由 DB2 Alphablox 客户支持使用。

日志文件滚动时间间隔设置

缺省情况下，DB2 Alphablox 每天创建一个新的日志文件，并且在应用程序服务器每次启动时创建一个新的日志文件。作为管理员，您可以选择在 DB2 Alphablox 管理页面中更改日志文件滚动设置，也可以通过编辑 `server.properties` 文件来更改日志文件滚动设置。

要在 DB2 Alphablox 管理页面中设置时间间隔，请打开浏览器并转到 DB2 Alphablox 管理页面中的“管理”选项卡，然后单击“一般属性”下的“系统”。缺省情况下，“新日志滚动时间间隔”设置为每天滚动。您可以编辑此设置以将其更改为永不滚动或者更改为指定的分钟数、小时数、天数、星期数或千字节数。

要在 `server.properties` 文件中设置日志文件滚动时间间隔，请打开 `server.properties` 文件，该文件位于以下目录中：

```
<repository_dir>/servers/<instanceName>/
```

找到 `LogRollOverInterval` 设置并使用下列值来将其更改为期望的设置：

将此值设置为下列其中一个值（缺省值是 1D）：

#M - 在 (#) 分钟后滚动

#H - 在 (#) 小时后滚动

#D - 在 (#) 天后滚动

#W - 在 (#) 个星期后滚动

#K - 在写入 (#) K 字节后滚动

无 - 永远不滚动

日志文件名

日志文件位于以下目录中:

```
<repository_root>/servers/<instance_name>/logs
```

活动日志文件的缺省名称是 `Server.log`。此名称是在通过单击**管理**选项卡 > **一般属性**部分 > **启动**链接访问的页面上配置的。

DB2 Alphablox 每次启动时都会创建一个新的日志文件。当 DB2 Alphablox 启动时, 将把先前日志文件重命名为:

```
Server<timestamp>.log
```

其中 `<timestamp>` 是日志中最后一个条目的时间。此时间戳记具有以下格式:

```
YYMMDD_HHmmSS
```

例如, 如果 DB2 Alphablox 上次是在 2000 年 6 月 3 日 6:22:35 PM 关闭的, 则当它下次启动时, 将把旧日志文件重命名为 `Server000603_182235.log` 并创建名为 `Server.log` 的新日志文件以存储当前日志条目。注意, 此时间戳记基于 24 小时时钟 (在此示例中, 18 表示 6 PM)。如果在**管理**页面上更改了日志文件的名称, 则将把时间戳记追加到新名称后面。

注: 活动日志文件不会将时间戳记追加到它的名称后面; 它的名称就是**管理**页面上指定的名称。

管理日志文件

可以将旧的日志文件永远保留下来而不会引起任何问题, 但是, 在一定时间之后, 您可能想要建立一个过程来将它们归档。保留旧的日志文件的缺点是它们占用了磁盘空间。它们所占用的磁盘空间大小取决于 DB2 Alphablox 的活动以及在管理页面上设置的缺省消息级别。如果将消息级别设置为 `DEBUG` 或 `VERBOSE`, 则日志文件将会相当快地增大。

由于旧的日志文件的重命名使用时间戳记作为其的文件名的组成部分, 所以, 您可以创建实用程序来帮助管理这些日志文件。例如, 您可以创建一个脚本来在日志文件被保留三个月之后删除或移动它们。

第 14 章 用户管理器和个性化 (Alphablox 8.4.1)

个性化管理器管理用于定制应用程序内容的个性化功能。在 DB2 Alphablox 8.4.1 中，用户管理器中提供的安全性功能和个性化功能分为两个模型。

DB2 Alphablox 8.4.1 提供基于 Servlet 2.4 和 Java 认证和授权服务标准的安全模型。第 55 页的『Alphablox 8.4.1 中的安全模型』中讨论了此安全模型。

对于个性化，新的个性化管理器提供了一个接口，用于在创建、装入或删除用户或组时通知 DB2 Alphablox。个性化管理器还允许您指定和修改用户或组的属性、修改组中包含的用户和组、创建自己的用户或组并将它添加到 Alphablox 存储库。

个性化管理器

DB2 Alphablox 8.4.1 中的新个性化管理器允许您从外部用户资源库中读取用户和组属性。在创建、装入或删除用户或组对象时，会通知 `com.alphablox.personalization` 包中的个性化管理器接口。此 `PersonalizationManager` 对象还允许使用用户资源库中定义的属性增加 DB2 Alphablox 用户和组。

如果正在使用个性化管理器，请在 `alphablox_dir/repository/servers/instance` 目录中添加 `config.xml` 文件。当 DB2 Alphablox 启动时，它执行下列操作：

1. 在存储库中查找 `config.xml` 文件。
2. 创建 `PersonalizationManager` 对象的实例。
3. 设置提供的属性。
4. 将 `PersonalizationManager` 对象指定给用户管理器服务。

以下是 `<Server>` 标记的示例：

```
<Server>
  <Service name="User Manager">
    <PersonalizationManager
      className="myPackage.CustomPersonalizationManager"
    />
  </Service>
</Server>
```

个性化管理器具有下列特征和行为：

- 只有在通过 DB2 Alphablox “管理页面” 定义外部用户资源库中的定制用户或组属性后，才能通过 DB2 Alphablox “管理页面” 使用这些属性。
- 在将这些属性定义为定制属性后，DB2 Alphablox 仅执行只读操作。通过 DB2 Alphablox “管理页面” 进行的更改仅对内存中的 DB2 Alphablox 是本地的，并且不会回写到外部用户资源库中。下次装入用户对象时会覆盖这些值。
- 如果需要在 `config.xml` 文件中更改连接信息或其他属性，则需要重新启动 Apache Tomcat，这将重新启动 DB2 Alphablox。

访问 JNDI 用户和组属性

本节中提供的信息适用于 DB2 Alphablox 8.4.1。

可以使基于 JNDI 的存储库中的用户和组属性可用于 DB2 Alphablox，并可通过 RepositoryBlox API 访问。要使这些属性可用，需要在 `alphablox_dir/repository/servers/instance/config.xml` 文件中指定属性：

1. 在 `alphablox_dir/repository/servers/instance` 目录中添加 `config.xml` 文件：

```
<Server>
  <Service name="User Manager">
    <PersonalizationManager
      className="com.alphablox.personalization.jndi.JNDIPersonalizationManager"
      connectionURL="ldap://machineName:port"
      userBase="ou=People,dc=company,dc=com"
      userSearch="(& (uid={0}) (objectclass=person))"
      userProperties="Mail, Phone, Fax"
      groupBase="ou=Groups,dc=company,dc=com"
      groupSearch="(& (cn={0}) (objectclass=groupofuniquenames))"
      groupProperties="Description"
    />
  </Service>
</Server>
```

下表包括可用于访问 JNDI 用户和组特性的属性。

属性	描述
connectionURL	必需。JNDI 连接 URL。示例： <code>ldap://machine1.ibm.com:10456</code>
contextFactory	可选。创建初始目录上下文时要使用的上下文工厂。缺省值是 <code>com.sun.jndi.ldap.LdapCtxFactory</code> 。
groupBase	必需。搜索组时要使用的 JNDI 基本。
groupProperties	必需。要从 JNDI 存储库中指定给 DB2 Alphablox 用户对象的用户属性。此属性将覆盖在 DB2 Alphablox “管理页面” 中定义的所有定制用户属性。
groupSearch	必需。搜索组时要使用的 JNDI 搜索字符串。组搜索可能包含 MessageFormat 替换语法 {0}，它将替换为要搜索的组的唯一名称。
groupSubtree	可选。确定在搜索组时是否使用 groupBase 的子树。
password	可选。连接至连接 URL 时要使用的密码。
userBase	必需。搜索用户时要使用的 JNDI 基本。
userName	可选。连接至连接 URL 时要使用的用户名。
userProperties	必需。要从 JNDI 存储库中指定给 Alphablox 用户对象的用户属性。此属性将覆盖在 DB2 Alphablox “管理页面” 中定义的所有定制用户属性。
userSearch	必需。设置搜索用户时要使用的 JNDI 搜索字符串。用户搜索可能包含 MessageFormat 替换语法 {0}，它将替换为要搜索的用户的唯一名称。
userSubtree	可选。搜索用户时要使用的 userBase 的子树。

要使用 RepositoryBlox API 访问 JSP 文件中的用户属性的值：

1. 通过使用 DB2 Alphablox “管理页面” 来添加相同的定制用户属性。转至**管理**选项卡 > **一般属性**页 > **客户属性**部分。第 73 页的『定制属性定义』中讨论了详细步骤。
2. 在 JSP 文件中，通过 RepositoryBlox API 访问此定制属性及其值：

```
<%@ taglib uri="bloxtld" prefix="blox" %>
...
<blox:repository id="myRepository" />
...
<html>
<head>
    <blox:header />
</head>
<body>
Your region is: <%= myRepository.getUserProperty("region") %>
...

```

3. 要使 *config.xml* 文件更改生效，重新启动 Apache Tomcat，这将重新启动 DB2 Alphablox。

第 15 章 用户管理器 (Alphasbx 8.4)

本章描述如何为“LDAP 集成”配置和使用 DB2 Alphasbx 用户管理器功能以及如何扩展“可扩展用户管理器”个性化引擎以实现定制安全性。

DB2 Alphasbx 用户管理器概述

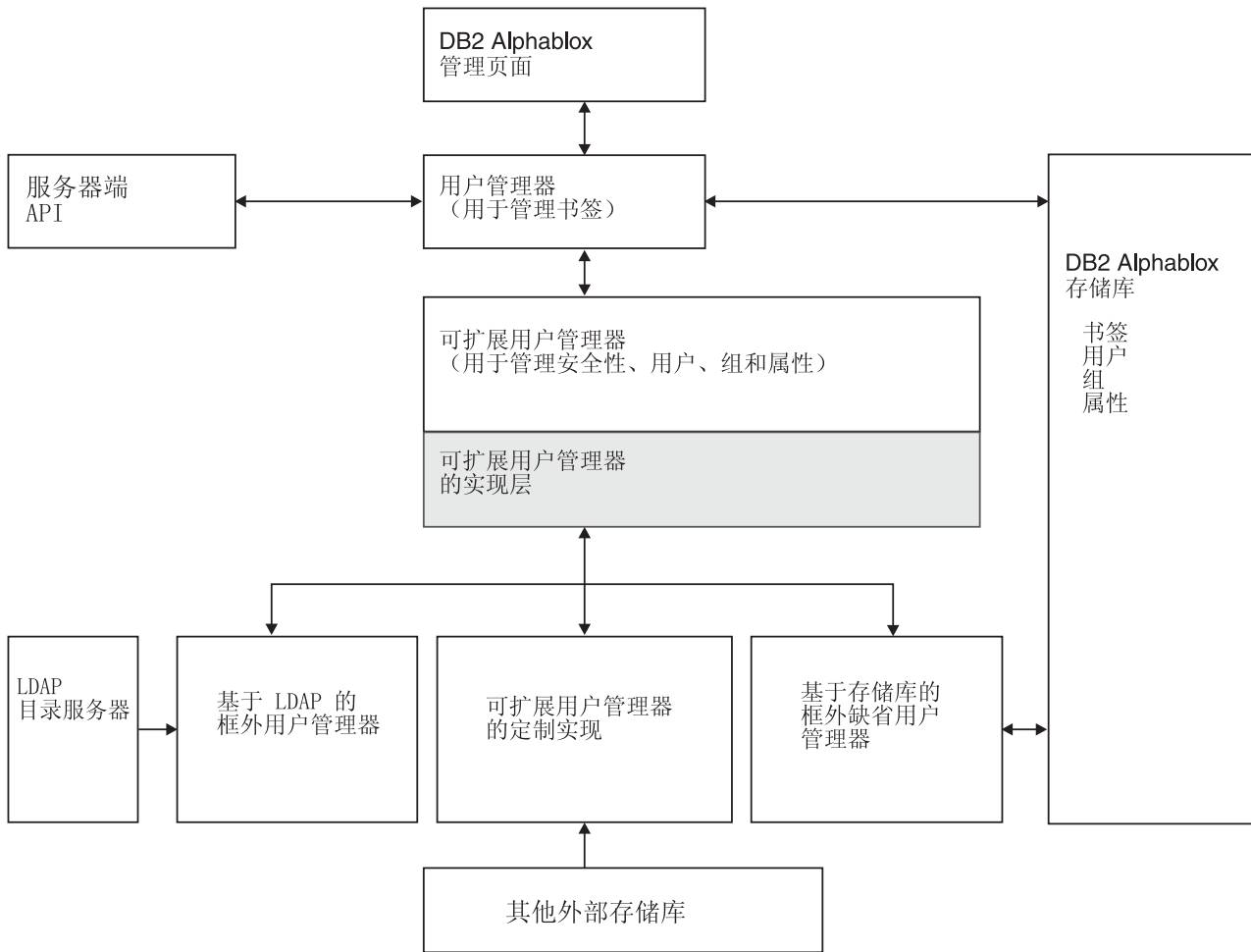
注：本节仅适用于 DB2 Alphasbx 8.4 (在)。

DB2 Alphasbx 用户管理器管理用户认证和授权并提供了用于定制应用程序内容的个性化能力。缺省情况下，DB2 Alphasbx 使用 DB2 Alphasbx 存储库和 J2EE 安全性 API 来管理用户和组信息。两个 J2EE 安全方法 (isUserInRole() 和 getUserPrinciple()) 用来标识用户身份以及该用户是否隶属于特定角色。

DB2 Alphasbx 还提供了现成的轻量级目录访问协议 (LDAP) 集成解决方案。此解决方案允许 DB2 Alphasbx 使用 LDAP 目录服务器来识别 DB2 Alphasbx 用户、组和定制属性，以便对用户进行认证和授权。

用户管理器是在名为可扩展用户管理器的个性化引擎基础上构建的。对于需要使用定制安全性的环境，可扩展用户管理器个性化引擎提供了一个接口，该接口允许扩展两个现成的安全性解决方案 (DB2 Alphasbx 存储库和 LDAP) 中的任何一个。您也可以插入别的外部用户管理器 (如 NTLM) 或一些现有的 Enterprise JavaBeans (EJB)。

下图显示了用户管理器及其可扩展用户管理器个性化引擎的体系结构：



注意，在图中：

- 通过 RepositoryBlox，可以从服务器端 API 使用程序访问用户、组及相关属性信息。
- DB2 Alphablox 提供了两个现成的安全性和个性化解决方案：基于 DB2 Alphablox 存储库的用户管理器和基于 LDAP 的用户管理器。
- 基于缺省存储库的用户管理器读写 DB2 Alphablox 存储库。
- 现成的基于 LDAP 的用户管理器仅对 LDAP 服务器执行读操作。
- 通过可扩展用户管理器 API，可以使用其他外部存储库（如 NTLM）或一些现有的 Enterprise JavaBeans (EJB)。
- 不管使用什么存储库（DB2 Alphablox、LDAP 或其他外部源），您总是可以通过 RepositoryBlox API 来访问用户属性。

可扩展用户管理器

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

可扩展用户管理器个性化引擎是 DB2 Alphablox 用户管理器的核心。它使您能够：

- 定制现成的用户认证和 / 或授权，
- 插入外部用户管理器以便访问 DB2 Alphablox 存储库外部的其他数据源中的用户 / 组信息，或者

- 使用除 J2EE 安全性 API 以外的安全设置

两个现成的 DB2 Alphablox 安全性和个性化实现 - 基于 DB2 Alphablox 存储库的用户管理器和基于 LDAP 的用户管理器 - 都是根据此引擎构建的。要使用基于 LDAP 的用户管理器, 需要执行一些配置步骤。第 87 页的『基于 LDAP 的用户管理器』描述了这些步骤。要实现定制安全性, 或者要定制两个现成的安全性实现中的任何一个, 可以扩展可扩展用户管理器。第 90 页的『可扩展用户管理器接口』描述了有关扩展可扩展用户管理器接口的详细信息。

基于 LDAP 的用户管理器

注: 本节仅适用于 DB2 Alphablox 8.4 (在 8.4.1 中不受支持)。

DB2 Alphablox 提供了现成的可扩展用户管理器实现以使用 LDAP 目录服务器来识别 DB2 Alphablox 用户、组和定制属性:

- DB2 Alphablox 仅对 LDAP 服务器执行读操作。
- 管理员只能创建已在 LDAP 服务器中定义的 DB2 Alphablox 用户或组。
- 管理员不能从 DB2 Alphablox 主页上的**管理**选项卡中重命名或修改成员资格。
- 通过 DB2 Alphablox 主页上的**管理**选项卡删除用户或组并不会从底层存储库中删除项, 也不会影响 LDAP 服务器。
- 缺省情况下, 除非 “admin” 用户是 LDAP 用户, 否则它不再可用。
- 即使用户 “guest” 和组 “public” 不存在于 LDAP 中, 它们也仍然可用。

配置 DB2 Alphablox 以使用 LDAP 用户管理器

注: 本节仅适用于 DB2 Alphablox 8.4 (在 8.4.1 中不受支持)。

基于 LDAP 的用户管理器可以与 WebSphere、WebLogic 和 Tomcat 应用程序服务器配合使用。执行下列步骤以将 DB2 Alphablox 配置为利用与 LDAP 目录服务器的现成集成:

1. 使用 ExtUserManager Telnet 控制台命令来将服务器设置为使用 LDAP, 如第 88 页的『设置基于 LDAP 的用户管理器属性』所述。
2. 确保使用以下 Telnet 控制台命令来将服务器属性 autoCreateUsers 设置为 true:

```
set server autoCreateUsers true
```

注: autoCreateUsers 的缺省值是 false。要使 DB2 Alphablox 能够自动地创建用户 (如果对 LDAP 服务器成功地认证了这些用户的话), 必须将此属性设置为 true。

注: “外部用户管理器” 必须已定义了组 “public” 和用户 “guest” (它们可以仅仅是虚拟实例)。

Apache Tomcat 配置的其他需求

1. 在 LDAP 目录服务器中创建 AlphabloxAdministrator 组。
2. 将至少一个用户添加到 AlphabloxAdministrator 组中, 或者, 将包含至少一个用户的现有组作为子组添加到 AlphabloxAdministrator 组中。

设置基于 LDAP 的用户管理器属性

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

1. 对 DB2 Alphablox 打开 Telnet 控制台连接。

注：如果通过 DB2 Alphablox 主页上的**管理**选项卡访问 Telnet 控制台，则必须重新启动 DB2 Alphablox 才能使更改生效。通过使用标准的 Telnet 控制台，不需要重新启动 DB2 Alphablox。

2. 输入以下命令：

```
ExtUserManager setToDefaults ldap "<ldapProps>"
```

其中，<ldapProps> 是用分号分隔的一系列属性值对。例如：

```
"host:localhost;port:389;admin:cn=DirectoryManager;  
password:password;debug:false;base:dc=alphablox,dc=com"
```

注意：以上命令应该在一行上输入，并且该行不能带有换行符。

在执行 ExtUserManager 命令之后，DB2 Alphablox 将从缺省的基于 DB2 Alphablox 存储库的用户管理器切换到基于 LDAP 的用户管理器。

注意：此切换将导致所有当前用户断开连接。

下表列出了所有可以在 <ldapProps> 字符串中定义的必需属性和可选属性：

属性	描述	示例
host	LDAP 目录服务器主机	host:localhost
port	LDAP 目录服务器端口	port:389
admin	LDAP 目录服务器管理员用户名	admin:cn=Directory Manager
password	LDAP 目录服务器管理员密码	password:myPassword
base	LDAP 目录服务器搜索条件	base:o=myCompany.com
debug	（可选）调试方式设置；缺省值是 false	debug:true

技巧：LDAP 中的搜索条件用来指定搜索起始点。它指向目录层次结构中的一个条目的专有名称以使搜索更为有效。

访问定制用户属性

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

您可以定义定制用户属性以使应用程序进一步个性化。一旦定义了定制用户属性，就可以针对每个用户对属性指定不同的值，然后使用 RepositoryBlox 的 API 来通过程序访问这些属性值。对于基于 LDAP 的用户管理器或者其他外部用户管理器，DB2 Alphablox 将只装入被标记为外部属性的定制用户属性。

可以使用 DB2 Alphablox 主页上的**管理**选项卡来定义定制用户属性。在定制属性定义页面上，确保选取了**外部属性**框。要了解详细的步骤，请参阅第 73 页的『定制属性定义』。一旦将定制用户属性标记为外部属性，当有关外部定制用户属性值的信息被请

求时（例如，通过 `RepositoryBlox.getUserProperty("myExternalUserProp")` 调用请求该信息），DB2 Alphablox 用户管理器就将自动转至外部源以访存该值。

运行时行为

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

基于 LDAP 的用户管理器具有下列运行时行为：

- 在运行时，DB2 Alphablox 不会自动地创建组。管理员必须通过 DB2 Alphablox 主页上的管理选项卡显式地注册 LDAP 组。
- 如果对 LDAP 服务器成功地认证了用户，并且如果通过 Telnet 命令将 `autoCreateUsers` 服务器属性设置为 `true`（如第 87 页的『配置 DB2 Alphablox 以使用 LDAP 用户管理器』所述），则 DB2 Alphablox 将自动创建用户。
- DB2 Alphablox 将只装入被定义为外部定制属性的 LDAP 用户和组属性。

可扩展用户管理器 Telnet 控制台命令

`ExtUserManager Telnet` 控制台命令允许您指示 DB2 Alphablox 使用另一个类来管理用户或组，或者指示它使用

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。外部存储库。

`ExtUserManager` 的一般语法如下所示：

```
ExtUserManager <Property> <Value>
```

其中，Property 可以是：

可能的属性	描述	值
<code>umclassname</code>	用户管理器类名	此值必须是类路径中实现了 <code>IUserManager</code> 接口的有效类名，否则将发生错误。要了解更多信息，请参阅第 91 页的『定制安全性实现』。
<code>userclassname</code>	用户类名	此值必须是类路径中实现了 <code>IUser</code> 接口的有效类名，否则将发生错误。要了解更多信息，请参阅第 91 页的『定制安全性实现』。
<code>groupclassname</code>	组类名	此值必须是类路径中实现了 <code>IGroup</code> 接口的有效类名，否则将发生错误。要了解更多信息，请参阅第 91 页的『定制安全性实现』。
<code>customstartupproperty</code>	一个定制启动属性，开发者可以在他们的代码中使用此属性来在用户管理器启动时读取值。	此属性的值。
<code>setToDefaults</code>	要使用的缺省存储库	请参阅下一节第 90 页的『设置缺省存储库』。

注：如果设置了新的定制启动属性，则必须停止并重新启动 DB2 Alphablox。

设置缺省存储库

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

要设置所有属性以使用缺省的基于 LDAP 的用户管理器，请使用以下 Telnet 控制台命令：

```
ExtUserManager setToDefaults ldap <ldapProps>
```

其中 ldaprops 是包含连接至 LDAP 所需的所有信息的属性。不需要重新启动 DB2 Alphablox，但是所有当前正在运行的用户都将断开连接。要了解使用基于 LDAP 的用户管理器时所需执行的配置步骤的详细信息以及有关 ldaprops 属性语法的详细信息，请参阅第 87 页的『基于 LDAP 的用户管理器』。

要复位所有属性以使用缺省的基于 DB2 Alphablox 存储库的用户管理器，请使用以下 Telnet 控制台命令。

```
ExtUserManager setToDefaults repository
```

此命令将指示用户管理器停止并再次启动以便从 DB2 Alphablox 存储库中获取最新的用户信息。不需要重新启动 DB2 Alphablox，但是所有当前正在运行的用户都将断开连接。

除去不再存在于外部用户存储库中的用户和组

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

以下命令指示可扩展用户管理器从存储库中除去那些不再存在于外部用户存储库（如 LDAP 或 NTLM）中的用户和组：

```
ExtUserManager clean
```

可扩展用户管理器接口

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

可扩展用户管理器框架由三个主要接口组成：IUserManager、IUser 和 IGroup。下表列出了每个接口的用途和其中的方法。还列出了实现特定接口的类。

接口	描述	实现类
<i>IUserManager</i>	负责在运行时创建和查找用户和组。	AbstractUserManager
<i>IUser</i>	为用户提供底层存储库中的认证、授权和用户属性的只读视图。	AbstractUser
<i>IGroup</i>	提供底层存储库中的成员资格信息和组属性的只读视图	AbstractGroup

接口和实现类位于 com.alphablox.personalization 包中。这个包的 Javadoc 位于以下目录中：

```
<db2alphablox_dir>/system/documentation/javadoc/blox/index.html
```

其中 <db2alphablox_dir> 是 DB2 Alphablox 的安装目录。

定制安全性实现

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

可扩展用户管理器具有三个 DB2 Alphablox 属性来标识用于认证和授权的用户、组 and 用户管理器类：

- `umclassname`

对于基于 DB2 Alphablox 存储库的用户管理器，缺省值是 `com.alphablox.personalization.repository.RepUserManager`。对于基于 LDAP 的用户管理器，缺省值是 `com.alphablox.personalization.ldap.LDAPUserManager`。

- `userclassname`

对于基于 DB2 Alphablox 存储库的用户管理器，此属性的缺省值是 `com.alphablox.personalization.repository.RepUser`。对于基于 LDAP 的用户管理器，缺省值是 `com.alphablox.personalization.ldap.LDAPUser`。

- `groupclassname`

对于基于 DB2 Alphablox 存储库的用户管理器，此属性的缺省值是 `com.alphablox.personalization.repository.RepGroup`。对于基于 LDAP 的用户管理器，缺省值是 `com.alphablox.personalization.ldap.LDAPGroup`。

您可以编写自己的用户、组或用户管理器类，然后使用 Telnet 控制台命令 `ExtUserManager` 来对这些 DB2 Alphablox 属性设置新值以指向您的用户/组/用户管理器类。

下表显示了当您需要定制部分或全部现成的安全方案时需要扩展的类。

目标	解决方案
缺省的基于 DB2 Alphablox 存储库的用户管理器中的定制认证或授权	扩展 <code>AlphabloxUser</code> 类
缺省的基于 LDAP 的用户管理器中的定制认证或授权	扩展 <code>ldapUser</code> 类
单点登录	实现 <code>AlphabloxUser</code> 类中的 <code>getPassword()</code> 方法
从除 DB2 Alphablox 或 LDAP 以外的存储库中读取用户/组属性	扩展 <code>AlphabloxUser</code> 或 <code>AlphabloxGroup</code> 类
从除 DB2 Alphablox 或 LDAP 以外的存储库中读取动态组成员资格信息	扩展 <code>AlphabloxUser</code> 或 <code>AlphabloxGroup</code> 类
NTLM 支持	全面实现可扩展用户管理器接口

注：要让 DB2 Alphablox 使用您的类，需要将类目录添加到启动批处理文件中指定的 DB2 Alphablox 类路径中。要了解有关如何设置类路径的详细信息，请参阅第 66 页的『设置类路径』。

单点登录

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

要在 DB2 Alphablox 数据源被配置为使用 DB2 Alphablox 用户名和密码时实现单点登录，可以使用 AlphabloxUser 类中的 getPassword() 方法来获取基本 64 位经过编码的密码。下表显示了不同情况下的单点登录解决方案：

用户管理器	数据源带有 LDAP 吗?	认证方法
基于 LDAP 的用户管理器	是	基本 DB2 Alphablox 认证（浏览器要求的认证）；不需要执行操作
基于 LDAP 的用户管理器	否	使用 getPassword()
其他存储库	是或否	使用 getPassword()

定制安全性示例

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

以下是为了定制安全性而扩展可扩展用户管理器的三个示例。每个示例都显示了：

- 如何扩展由缺省的基于 DB2 Alphablox 存储库的用户管理器使用的缺省用户、组或用户管理器类，以及
- 使用 Telnet ExtUserManager 命令来设置新的类名。

要获取全面实现外部用户管理器的完整示例，请参阅以下目录中的“简单用户管理器”示例：

```
<db2alphablox_dir>/system/documentation/admin/Examples/
```

提供了 Java 源文件。

示例 1：将 DB2 Alphablox 设置为使用外部用户管理器

注：本节仅适用于 DB2 Alphablox 8.4（在 8.4.1 中不受支持）。

此示例演示如何定制缺省的基于 DB2 Alphablox 存储库的用户管理器。在本例中，您可能想在用户管理器启动时从文件中读取定制属性列表。涉及的步骤包括：

1. 在包中编写一个定制用户管理器类（在此示例中，该类是名为 com.myCompany.user 的包中名为 MyUserManager 的类）
 - 该类扩展在基于 DB2 Alphablox 存储库的用户管理器中使用的 AlphabloxUserManager 类
2. 使用以下 Telnet 控制台命令来设置新的 umclassname 服务器属性值：

```
ExtUserManager umclassname com.myCompany.user.MyUserManager
```

定制 MyUserManager 类可以是以下这个简单示例：

```
package com.myCompany.user;
import com.alphablox.personalization.alphablox.*;
import com.alphablox.personalization.*;
import java.util.*;

public class MyUserManager extends AlphabloxUserManager {
    void start(Properties props) throws PEngineException {
        super.start(props);
    }
}
```



```

        String myXmlFile = (String) prop.get("customstartupprop");
        // read the xml file and do other things
    }
}

```

示例 2: 将 DB2 Alphablox 设置为使用另一个用户类

注: 本节仅适用于 DB2 Alphablox 8.4 (在 8.4.1 中不受支持)。

此示例演示如何定制缺省的基于 DB2 Alphablox 存储库的用户管理器中的用户授权。在本例中, 您想要允许每个用户都可以访问 DB2 Alphablox 或应用程序并覆盖 *IUser* 接口中的 *isUserInRole()* 方法。涉及的步骤包括:

1. 在包中编写一个定制用户类 (在此示例中, 该类是名为 `com.myCompany.user` 的包中名为 `MyUser` 的类)
 - 该类扩展在基于 DB2 Alphablox 存储库的用户管理器中使用的缺省 `AlphabloxUser` 类
2. 使用以下 Telnet 控制台命令来设置新的 `userclassname` 服务器属性值:

```
ExtUserManager userclassname com.myCompany.user.MyUser
```

其中 `MyUser` 可以是以下这个简单示例:

```

package com.myCompany.user;
import com.alphablox.personalization.repository.*;
import com.alphablox.personalization.*;
import java.util.*;

public class MyRepositoryUser extends AlphabloxUser {
    public boolean isUserInRole(HttpServletRequest req,
        String [] roles) throws PEngineException {
        // Everybody can get in
        return true;
    }
}

```

示例 3: 将 DB2 Alphablox 设置为使用另一个组类

注: 本节仅适用于 DB2 Alphablox 8.4 (在 8.4.1 中不受支持)。

此示例演示如何定制缺省的基于 DB2 Alphablox 存储库的用户管理器以便每次完成组成员资格检查时打印系统消息。涉及的步骤包括:

1. 在包中编写一个定制组类 (在此示例中, 该类是名为 `com.myCompany.user` 的包中名为 `MyGroup` 的类)
 - 该类扩展在基于 DB2 Alphablox 存储库的用户管理器中使用的缺省 `AlphabloxGroup` 类
2. 使用以下 Telnet 控制台命令来设置新的 `groupclassname` 服务器属性值:

```
ExtUserManager groupclassname com.myCompany.user.MyGroup
```

其中 `MyGroup` 可以是下面这个简单示例:

```

package com.myCompany.user;
import com.alphablox.personalization.alphablox.*;
import com.alphablox.personalization.*;
import java.util.*;

public class MyGroup extends AlphabloxGroup {
    public boolean containsUser(IUser user,
        boolean checkSubGroups) throws PEngineException {

```

```

        boolean exists = super.containsUser(user,checkSubGroups);
        System.out.println("User membership checked.");
        return exists;
    }
}

```

接口方法交叉引用

注: 本节仅适用于 DB2 Alphablox 8.4 (在 8.4.1 中不受支持)。

本节提供可扩展用户管理器中的三个接口的参考材料。对于接口中的每个方法, 提供了对该方法的描述、该方法的语法以及有关其用法的说明。

要获取定制的用户管理器的完整示例, 请参阅以下目录中的“简单用户管理器”示例:

<db2alphablox_dir>/system/documentation/admin/Examples/

接口	方法
<i>IUserManager</i>	findGroup() findUser() getExternalProperties() getPrincipleUserName() hasExternalEditor() resume() setCaseSensitiveGroups() setCaseSensitiveUsers() start() stop() suspend()
<i>IUser</i>	authenticate() authorize() getName() getPassword() getPropertiesSubset() isUserInRole() refresh()
<i>IGroup</i>	containsGroup() containsUser() getName() getPropertiesSubset() refresh()

IUserManager 接口

注: 本节仅适用于 DB2 Alphablox 8.4 (在 8.4.1 中不受支持)。

AbstractUserManager 类实现 *IUserManager* 接口以便在运行时查找用户和组信息并指出用户标识。要扩展这个类, 请在代码中添加以下 import 语句:

```
import com.alphablox.personalization.*;
```

要获取扩展 `AbstractUserManager` 类的示例，请参阅第 92 页的『示例 1: 将 DB2 Alphablox 设置为使用外部用户管理器』。

findGroup()

查询组并返回适当的 `IGroup` 实例。

语法

```
IGroup findGroup(String id, boolean fromCache);  
// throws PEngineException
```

其中:

自变量	描述
<code>id</code>	组标识。
<code>fromCache</code>	是否在高速缓存中查询组

用法

在下列情况下，此方法将返回 `null`:

- 如果在内存中不存在该组并且 `fromCache` 设置为 `true`
- 如果在内存中不存在该组、该组在底层用户管理器存储库中不是有效的组并且 `fromCache` 设置为 `true`

findUser()

查找用户并返回 `IUser` 的适当实例。

语法

```
IUser findUser(String id, boolean fromCache);  
// throws PEngineException
```

其中:

自变量	描述
<code>id</code>	用户标识
<code>fromCache</code>	是否从高速缓存中查找用户

用法

在下列情况下，此方法将返回 `null`:

- 如果在内存中不存在该用户并且 `fromCache` 设置为 `true`
- 如果在内存中不存在该用户、该用户在底层用户管理器存储库中不是有效的用户并且 `fromCache` 设置为 `true`

getExternalProperties()

返回通过外部编辑器定义的属性的字符串数组。

语法

```
String[] getExternalProperties(); //throws PEngineException
```

getPrincipleUserName()

返回与此请求相关联的用户名。

语法

```
String getPrincipleUserName(HttpServletRequest request);  
// throws PEngineException
```

其中:

自变量	描述
request	当前 HTTP 请求

用法

在创建新的 DB2 Alphablox 会话时, DB2 Alphablox 使用此方法来确定用户名。可以使用此方法来覆盖 DB2 Alphablox 的缺省行为。

hasExternalEditor()

返回是否存在外部用户管理器编辑器。

语法

```
boolean hasExternalEditor();
```

用法

如果外部用户管理器具有自己的编辑器, 则返回 `true`。例如, 当使用基于 LDAP 的用户管理器时, 此方法返回 `true`。对于缺省的基于 DB2 Alphablox 存储库的用户管理器, 此方法返回 `false`。除非是具有外部编辑器的完整可扩展用户管理器实现, 否则不应该将此方法实现为返回 `true`。

resume()

使用户管理器继续工作。

语法

```
void resume(); // throws PEngineException
```

用法

当 DB2 Alphablox 用户管理器继续提供服务时, 调用此方法。在 DB2 Alphablox 用户管理器的服务暂挂之后, 调用此方法以使其继续工作。

示例

下列代码记录一条指示用户管理器已重新启动的消息:

```
import com.alphablox.personalization.*;  
public class MyUserManager extends AbstractUserManager {  
    ...  
    public void resume() throws PEngineException {  
        System.out.println("Resumed");  
    }  
    ...  
}
```

另请参阅

第 98 页的『suspend()』

setCaseSensitiveGroups()

指定组名是否应该区分大小写。

语法

```
void setCaseSensitiveGroups(boolean caseSensitive);  
    // throws PEngineException
```

其中:

自变量	描述
caseSensitive	true - 组名区分大小写

用法

当将此方法设置为 true 时，组名将区分大小写，同时用户管理器也将遵守该规则。

setCaseSensitiveUsers()

指定用户名是否应该区分大小写。

语法

```
void setCaseSensitiveUsers(boolean caseSensitive);  
    // throws PEngineException
```

其中:

自变量	描述
caseSensitive	true - 用户名区分大小写

用法

当将此方法设置为 true 时，用户名将区分大小写，同时用户管理器也将遵守该规则。

start()

启动用户管理器。

语法

```
void start(java.util.Properties props);  
    // throws PEngineException
```

其中:

自变量	描述
props	与个性化相关的 DB2 Alphablox 属性。有效属性包括: umclassname、groupclassname、customproperty 和 ldapprops。要了解详细信息，请参阅第 89 页的『可扩展用户管理器 Telnet 控制台命令』。

用法

每次 DB2 Alphablox 用户管理器启动时都会调用此方法。此方法用来与适当的存储库建立连接并根据需要将用户和组对象实例化。对于基于 DB2 Alphablox 存储库的用户管理器，将把存储库中的所有用户和组实例化。在基于 LDAP 的用户管理器中，只需要创建 DB2 Alphablox 中注册的用户和组。因此，好的做法是根据需要创建用户和组，例如在调用 `findUser()` 和 `findGroup()` 方法时创建用户和组。

stop()

停止用户管理器并释放所有资源。

语法

```
void stop(); // throws PEngineException
```

用法

当 DB2 Alphablox 用户管理器停止它的服务时，调用此方法。

另请参阅

第 97 页的『`start()`』，第 98 页的『`suspend()`』。

suspend()

暂挂外部用户管理器并释放未使用的资源，如数据库连接。

语法

```
void suspend(); // throws PEngineException
```

用法

当 DB2 Alphablox 用户管理器暂挂它的服务时，调用此方法。

另请参阅

第 96 页的『`resume()`』

IUser 接口

要扩展这个类，请在代码中添加以下 `import` 语句：

```
import com.alphablox.personalization.*;
```

要获取扩展 `AbstractUser` 类的示例，请参阅第 93 页的『示例 2：将 DB2 Alphablox 设置为使用另一个用户类』。

authenticate()

检查通过 HTTP 请求提供的用户和密码是否与存储库中存储的用户和密码相同。

语法

```
boolean authenticate(HttpServletRequest request,  
                    String authorizationHeader);  
// throws PEngineException
```

其中：

自变量	描述
request	HTTP 请求。
authorizationHeader	是否在高速缓存中查询组

用法

如果用户保存的密码与参数密码相同，则返回 `true`。此方法仅适用于 DB2 Alphablox Apache Tomcat 配置（适用于 Apache Tomcat；不适用于 Microsoft IIS）。在该配置中，DB2 Alphablox 使用基本认证（即，浏览器要求的认证）。当用户第一次访问会话时，DB2 Alphablox Tomcat 拦截器将调用此方法。此方法有两个参数：请求对象和编码的授权头。这个授权头包含用户名和密码。可以使用以下方法来对授权头进行译码：

```
AbstractUserManager.getDecoder().decode(authorizationHeader);
```

在获取密码之后，如果您打算将它用作 `getPassword()` 方法的返回值以允许对该数据源进行单点登录，则可以选择将密码保存在对象内存中。

authorize()

检查用户在提供的角色列表中是否存在。

语法

```
boolean authorize(HttpServletRequest request,
                  String[] roles);
// throws PEngineException
```

其中：

自变量	描述
request	HTTP 请求。
roles	角色列表，将对这个列表进行检查以确定用户是否包含在此列表中。

用法

如果用户是其中一个提供的角色，则返回 `true`。

getEmail()

获取用户的电子邮件。

语法

```
String getEmail(); // throws PEngineException
```

用法

如果 DB2 Alphablox 维护电子邮件名称而不是从外部存储库中读取该名称，则返回 `null`。对于 LDAP，将从 LDAP 中获取用户的电子邮件和全名，并且不能在 DB2 Alphablox 管理页面中编辑它们的值。在该情况下，可以使用此方法来获取用户的电子邮件。

getFullName()

获取用户的全名。

语法

```
String getFullName(); // throws PEngineException
```

用法

如果 DB2 Alphablox 维护全名而不是从外部存储库中读取该名称，则返回 null。对于 LDAP，将从 LDAP 中获取用户的电子邮件和全名，并且不能在 DB2 Alphablox 管理页面中编辑它们的值。在该情况下，可以使用此方法来获取用户的全名。

getName()

获取用户名。

语法

```
String getName(); // throws PEngineException
```

用法

用户名必须是有效的 DB2 Alphablox 用户名。

getPassword()

获取用户的密码。

语法

```
String getPassword(); // throws PEngineException
```

用法

获取用户的基本 64 位经过编码的密码。当数据源被设置为使用 DB2 Alphablox 用户名和密码时，DB2 Alphablox 将使用保存在存储库中经过编码的密码来进行认证以及访问数据源。使用此方法来实现定制安全性，如对数据源的单点登录。将在内存中临时使用通过此方法保存的任何密码。注意，应该先使用下列代码来对此密码进行编码：

```
AbstractUserManager.getEncoder().encode(password);
```

然后再将其返回给调用程序。

另请参阅

第 91 页的『单点登录』。

getPropertiesSubset()

获取您感兴趣的属性子集。

语法

```
java.util.Properties getPropertiesSubset(String[] propList);  
// throws PEngineException
```

其中：

自变量

描述

propList

属性子集的字符串数组

用法

返回一个 `Properties` 对象，该对象包含所传递的用户属性子集。除非开发者进行了更改，否则假定将从内存中获取值。

isUserInRole()

标识用户是否属于其中一个指定的角色。

语法

```
boolean isUserInRole(HttpServletRequest request,  
                      String[] roles);  
// throws PEngineException
```

其中:

自变量	描述
<code>request</code>	HTTP 请求
<code>roles</code>	用户所属的角色。

用法

如果用户属于列表中的其中一个角色，则返回 `true`。角色实现留待开发者完成。此方法的实现基于标准的 J2EE API `Request.isUserInRole(String role)` 方法。DB2 Alphablox 使用此方法来确定用户对管理功能的访问权以及书签写访问权。

对于基于 LDAP 的用户管理器，此方法调用 `IGroup.containsUser()` 方法，这是因为角色等同于 LDAP 中的组。

另请参阅

第 101 页的『`containsGroup()`』

refresh()

刷新在内存中高速缓存的所有信息，并从底层存储库中获取最新的信息。

语法

```
void refresh(); // throws PEngineException
```

用法

强制刷新内存中保存的任何用户信息并从外部存储库中获取最新的信息。

IGroup 接口

要扩展这个类，请在代码中添加以下 `import` 语句:

```
import com.alphablox.personalization.*;
```

要获取扩展 `AbstractGroup` 类的示例，请参阅第 93 页的『示例 3: 将 DB2 Alphablox 设置为使用另一个组类』。

containsGroup()

检查此组是否包含子组。

语法

```
boolean containsGroup(IGroup group, boolean checkSubGroups);  
    // throws PEngineException
```

其中:

自变量	描述
group	IGroup 的实例。
checkSubGroups	true - 检查子组; false - 不检查子组。

用法

如果此组包含子组, 则返回 true。

containsUser()

检查此组是否包含指定的用户。

语法

```
boolean containsUser(IUser user, boolean checkSubGroups);  
    // throws PEngineException
```

其中:

自变量	描述
user	IUser 的实例。
checkSubGroups	true - 检查子组; false - 不检查子组

用法

如果该组包含指定的用户, 则返回 true。

getName()

获取组名。

语法

```
String getName();    // throws PEngineException
```

用法

组名必须是有效的 DB2 Alphablox 组名。

getPropertiesSubset()

获取您感兴趣的组属性子集。

语法

```
java.util.Properties getPropertiesSubset(String[] propList);  
    // throws PEngineException
```

其中:

自变量	描述
propList	包含属性的字符串数组。

用法

返回一个 `Properties` 对象，该对象至少包含所传递的组属性列表。除非开发者进行了更改，否则假定将从内存中获取值。

refresh()

刷新在内存中高速缓存的所有信息，并从底层存储库中获取最新的信息。

语法

```
void refresh();           // throws PEngineException
```

用法

强制刷新内存中保存的任何组信息并从外部存储库中获取最新的信息。

第 16 章 使用数据库存储库

本章描述如何配置 DB2 Alphablox 以将关系数据库用作 DB2 Alphablox 存储库。

DB2 Alphablox 存储库概述

DB2 Alphablox 存储库是一个对象库，DB2 Alphablox 使用它来跟踪应用程序、用户、组、书签和其他信息。存储库既可以位于操作系统文件系统中，也可以位于关系数据库中。如果将 DB2 Alphablox 配置设置为在集群环境中运行，则存储库必须位于关系数据库中。关系数据库允许多个服务器节点读写存储库，同时确保存储库中的数据具有一致的状态。

DB2 Alphablox 环境中的存储库

在 DB2 Alphablox 中，“存储库管理器”控制着对存储库的访问。无论存储库是位于文件系统中还是位于关系数据库中，“存储库管理器”都使用 Java 命名和目录接口（JNDI）来与存储库通信。JNDI 层对不同类型的存储库（DB2 Alphablox 文件系统存储库或数据库存储库）使用不同的服务提供程序。

存储库支持的数据库与数据源访问所支持的那些数据库相同，如《安装指南》所述。要了解有关 DB2 Alphablox 的体系结构的更多详细信息，请参阅第 7 页的『DB2 Alphablox 体系结构』。

存储库转换实用程序是一个 Java 程序，它用来在文件系统与文件系统之间、在文件系统与数据库之间、在数据库与文件系统之间或者在不同数据库之间转换存储库。在安装过程中，转换实用程序将自动运行，也可以手工运行它。要了解有关使用转换实用程序的详细信息，请参阅第 106 页的『使用存储库转换实用程序』。

关系存储库的优点

将关系数据库用作 DB2 Alphablox 存储库具有两项主要优点：

- 关系存储库可以由多台服务器访问，因此，对 DB2 Alphablox 启用了集群环境，并提供了一种方法来实现实际上不受限制的可伸缩性。
- 关系存储库提供了数据库环境中的所有工具，这些工具具有工业强度，并且能够完成诸如确保事务完整性、执行备份和复原操作、回滚至一致状态以及复制数据库等工作。

存储库中存储了对于 DB2 Alphablox 的操作来说至关重要的对象，因此，通过提供所有数据库工具以确保数据完整性，确保了系统更健壮并且更可靠。

配置 DB2 Alphablox 存储库

DB2 Alphablox 存储库的初始状态是在安装过程中确定的。DB2 Alphablox 存储库有两个服务提供程序，根据存储库类型的不同，它使用其中一个服务提供程序来访问存储库：**DB2 Alphablox 文件系统服务提供程序**和**DB2 Alphablox 数据库服务提供程序**。缺省状态是使用**DB2 Alphablox 文件系统服务提供程序**。DB2 Alphablox 支持的所有

关系数据库都是 **DB2 Alphablox 数据库服务提供程序**所支持的数据库。要了解有关在安装过程中指定存储库类型以及有关受支持的关系数据库的详细信息，请参阅《[安装指南](#)》。

检查存储库类型

要检查服务器正在使用哪个存储库服务提供程序，请执行下列步骤：

1. 确保 DB2 Alphablox 正在运行。
2. 作为 *admin* 用户或作为隶属于 *administrators* 组的用户登录到 DB2 Alphablox 主页。
3. 单击**管理**选项卡。“管理”选项卡的**常规**页面显示。
4. 在页面的一般属性部分中，单击**存储库管理器**链接以查看 DB2 Alphablox 存储库属性。

存储库服务提供程序条目列出了使用中的 JNDI 服务提供程序，**数据库适配器**条目列出了所使用的 JDBC 驱动程序，其他条目列出了数据源的配置信息。如果正在使用文件系统服务提供程序，则**存储库位置**条目将列出存储库文件的存储目录。

注：不能在“存储库管理器”属性页中更改存储库类型。要更改存储库类型，请使用存储库转换实用程序。

使用存储库转换实用程序

如果要更改存储库类型（例如，由文件系统更改为 Oracle 数据库），则必须运行存储库转换实用程序。存储库转换实用程序是一个 Java 程序，它在命令行窗口（在 Windows 系统上是 MS-DOS 窗口，在 Linux 和 UNIX 系统上是 *xterm* 或其他命令窗口）中运行。它创建将 DB2 Alphablox 存储库从一个位置移至另一位置时所必需的表和 / 或文件。可以使用存储库转换实用程序来将存储库从文件系统移至文件系统、从文件系统移至数据库、从数据库移至文件系统或者从数据库移至数据库。

启动存储库转换实用程序

在安装了 DB2 Alphablox 的 Windows 系统上，可以从**开始**菜单中启动存储库转换实用程序，也可以通过在 Windows 系统上运行以下文件来启动它。

要点：在运行存储库转换实用程序之前，总是首先关闭 DB2 Alphablox。

```
<db2alphablox_dir>/Tools/convert/ConvertRepository.exe
```

或者，在 Linux 和 UNIX 系统上，运行以下文件：

```
<db2alphablox_dir>/Tools/convert/ConvertRepository
```

其中 *<db2alphablox_dir>* 是 DB2 Alphablox 的安装目录。

存储库转换实用程序交互式命令行选项

下表描述了存储库转换实用程序主菜单的每个交互式命令行选项。

选项	描述
1	设置 DB2 Alphablox 文件管理器根目录

设置用作文件系统存储库的存储库文件的目录位置。此目录必须是有效的 DB2 Alphablox 目录，否则操作将失败并返回错误。

选项	描述
2 设置 DB2 Alphablox 实例名	选择存储库转换实用程序将要访问的 DB2 Alphablox 实例。指定的实例必须是有效的 DB2 Alphablox 实例，否则操作将失败并返回错误。
3 将一个存储库转换为另一个存储库	<p>将现有存储库转换为另一个存储库，从而将所有的必需数据从一个存储库移至另一个存储库。提供了用来将存储库从文件转换为数据库、从数据库转换为文件、从文件转换为文件以及从数据库转换为数据库的选项。当您转换存储库时，您将被提示使用下列其中一个选项：</p> <ul style="list-style-type: none"> • COPY: 将一个存储库的内容复制到另一个存储库中，并保留原始存储库不变。 • MOVE: 从一个存储库中移出存储库内容并将内容移入另一个存储库。原始存储库被删除。 <p>还将提示您对目标存储库使用下列其中一个选项：</p> <ul style="list-style-type: none"> • NEW: 在目标位置创建一个新存储库。 • UPDATE: 根据源存储库更新目标存储库中的数据，而不是将目标存储库替换为新存储库。 • OVERWRITE: 在目标存储库中重新创建表并将所有数据都替换为源存储库中的信息。OVERWRITE 选项不保留目标存储库中的任何数据。
4 创建一个空的数据库存储库	在数据库中创建 DB2 Alphablox 存储库所需的表。此选项仅创建具有初始内容的表；它不会用任何存储库对象来填充那些表。
5 验证并修复存储库	检查存储库是否有问题并报告所找到的任何问题。提供了用于验证文件存储库和数据库存储库的选项。对于数据库存储库，此选项还可以更正某些问题。
6 将 DB2 Alphablox 更改为使用另一个存储库	将 DB2 Alphablox 实例重定向为指向另一个存储库。更改到的存储库必须存在，并且必须能够从 DB2 Alphablox 机器访问它。
7 转换实用程序选项	允许在详细实用程序消息与正常实用程序消息之间进行切换，并允许擦除用于存储历史记录缓冲区（转换实用程序使用该历史记录来记住您输入的信息）。并且，提供了一个选项，允许您指定不同的 DDL 模式文件来创建存储库表。除非已经进行了全面测试，否则不要指定不同的 DDL 文件。
8 退出	退出存储库转换实用程序。

存储库转换实用程序将保留一个名为 `repositoryconvert.log` 的日志文件，此日志文件保存了转换实用程序会话中的所有活动。`repositoryconvert.log` 文件与转换实用程序位于同一个目录中（`<db2alphablox_dir>/Tools/convert`）。

从文件系统转换为数据库

在将存储库从文件系统转换为数据库之前，请收集下列信息：

- 文件系统存储库的完整路径（例如，`d:\alphablox\repository`）

- 数据库的连接信息

以下过程描述了在将 DB2 Alphablox 存储库从文件系统转换为数据库时所需执行的步骤。

1. 关闭 DB2 Alphablox。
2. 启动存储库转换实用程序（要了解详细信息，请参阅第 106 页的『启动存储库转换实用程序』）。
3. 通过输入数字 **3** 来选择选项 3（将一个存储库转换为另一个存储库）并按 Enter 键。
4. 通过输入数字 **1** 来选择选项 1（将文件转换为数据库）并按 Enter 键。
5. 通过按 Enter 键确认存储库根目录。

注： 如果显示为缺省目录的目录不是存储库目录，则确保您正在访问正确的 DB2 Alphablox 实例。存储库转换实用程序所访问的缺省实例是已安装的实例的名称，缺省情况下，该名称为 AlphabloxAnalytics。如果您使用另一个实例名，则退出当前序列并从主菜单上的选项 2 设置正确的实例名。

6. 如果所有内容都正确，则通过输入数字 **1** 选择**继续**并按 Enter 键。
7. 选择与数据库服务器相对应的数据库。例如，如果您正在使用 Oracle 8.1.7，则输入 **2**。
8. 根据提示，输入数据库的配置信息。
9. 如果所有内容都正确，则通过输入数字 **1** 选择**继续**并按 Enter 键。
10. 输入 COPY 或 MOVE。COPY 保留旧存储库不变并在目标存储库中创建旧存储库的副本，MOVE 将删除旧存储库并在目标存储库中创建一个新存储库。
11. 输入 NEW、UPDATE 或 OVERWRITE。使用 NEW 来创建带有新表的新存储库（如果还不存在该存储库的话）。使用 UPDATE 来保留存储库的现有表结构，并将其更新为包括源存储库中的数据。使用 OVERWRITE 来删除所有旧数据和数据库表并在原位置重新创建新表和数据。
12. 如果要让 DB2 Alphablox 实例使用新存储库，则对将 **DB2 Alphablox 更新为使用新存储库**提示输入 **Y**。
13. 要更新 DB2 Alphablox 实例的所有必需属性，请对**更新 DB2 Alphablox 属性**提示选择 ALL。ALL 指定要转换所有服务器属性，SPECIFIC 指定仅转换特定于本地机器的服务器属性（而不转换集群属性），GLOBAL 指定仅转换由集群共享的属性（而不转换本地机器条目）。NONE 指定不对 DB2 Alphablox 实例更改任何属性。
14. 如果所有内容都正确，则通过输入数字 **1** 选择**继续**并按 Enter 键。
15. 转换完成后，您将返回到主菜单。输入 **8** 以退出存储库转换实用程序。

从数据库转换为文件系统

在将存储库从数据库转换为文件系统之前，请收集下列信息：

- 数据库存储库的连接信息
- 文件系统存储库将要使用的驻留位置的完整路径（例如，d:\alphablox\Repository）

以下过程描述了在将 DB2 Alphablox 存储库从数据库转换为文件系统时所需执行的步骤。

1. 关闭 DB2 Alphablox。
2. 启动存储库转换实用程序（要了解详细信息，请参阅第 106 页的『启动存储库转换实用程序』）。
3. 通过输入数字 **3** 来选择选项 3（**将一个存储库转换为另一个存储库**）并按 Enter 键。
4. 通过输入数字 **2** 来选择选项 2（**将数据库转换为文件**）并按 Enter 键。
5. 选择与数据库服务器相对应的数据库。例如，如果您正在使用 Oracle 8.1.7，则输入 **2**。
6. 根据提示，输入数据库的配置信息。
7. 如果所有内容都正确，则通过输入数字 **1** 选择**继续**并按 Enter 键。

存储库转换实用程序将尝试连接至该数据库。

8. 输入目标存储库根目录，然后按 Enter 键。
9. 如果所有内容都正确，则通过输入数字 **1** 选择**继续**并按 Enter 键。
10. 输入 COPY 或 MOVE。COPY 保留旧存储库不变并在目标存储库中创建旧存储库的副本，MOVE 将删除旧存储库并在目标存储库中创建一个新存储库。
11. 输入 NEW、UPDATE 或 OVERWRITE。使用 NEW 来创建新存储库（如果还不存在该存储库的话）。使用 UPDATE 来保留存储库的现有结构，并使用源存储库中的数据来更新该存储库。使用 OVERWRITE 来删除所有旧数据和结构并在原位置重新创建新结构和数据。
12. 如果要让 DB2 Alphablox 实例使用新存储库，则对将 **DB2 Alphablox 更新为使用新存储库**提示输入 **Y**。
13. 要更新 DB2 Alphablox 实例的所有必需属性，请对**更新 DB2 Alphablox 属性**提示选择 ALL。ALL 指定要转换所有服务器属性，SPECIFIC 指定仅转换特定于本地机器的服务器属性（而不转换集群属性），GLOBAL 指定仅转换由集群共享的属性（而不转换本地机器条目）。NONE 指定不对 DB2 Alphablox 实例更改任何属性。
14. 如果所有内容都正确，则通过输入数字 **1** 选择**继续**并按 Enter 键。
15. 转换完成后，您将返回到主菜单。输入 **8** 以退出存储库转换实用程序。

配置实例以使用现有的存储库

在将 DB2 Alphablox 实例配置为使用现有存储库之前，请收集所要连接的数据库存储库的连接信息。

以下过程描述了在将 DB2 Alphablox 实例配置为使用现有 DB2 Alphablox 存储库时所需执行的步骤。

1. 关闭 DB2 Alphablox。
2. 启动存储库转换实用程序（要了解详细信息，请参阅第 106 页的『启动存储库转换实用程序』）。
3. 通过输入数字 **6** 并按 Enter 键来选择选项 6（**将 DB2 Alphablox 更改为使用另一个存储库**）。
4. 通过输入数字 **2** 来选择选项 2（**数据库存储库是目标**）并按 Enter 键。
5. 选择与数据库服务器相对应的数据库。例如，如果您正在使用 Oracle 8.1.7，则输入 **2**。

6. 根据提示，输入数据库的配置信息。
7. 如果所有内容都正确，则通过输入数字 **1** 选择继续并按 Enter 键。

存储库转换实用程序将尝试连接至该数据库，然后，它将把 DB2 Alphablox 实例更新为使用指定的存储库。

8. 输入 **8** 以退出存储库转换实用程序。

命令行语法

在大多数情况下，您能够以交互方式使用存储库转换实用程序，如先前各节所述。但是，您可以使用命令行选项来指定不同的数据库 DDL 文件或者将转换实用程序作为自动化脚本的一部分来运行。存储库实用程序的基本语法如下所示：

```
java -cp [class_path] com.alphablox.util.convert.Convert operation
destination [source] [arguments]
```

其中：

<i>class_path</i>	是 DB2 Alphablox 实例的 Java 类路径。请在 DB2 Alphablox 启动文件（例如，Windows 平台上的 AnalysisServer.bat 以及 Linux 和 UNIX 平台上的 AnalysisServer.sh）中查找 <i>class_path</i> 的示例。
<i>operation</i>	是要执行的存储库转换实用程序操作。要了解每个可用操作的描述，请参阅第 110 页的『操作描述』。
<i>destination</i>	是用于描述目标存储库的属性文件的相对路径或绝对路径。要获取样本目标文件，请参阅第 111 页的『样本源属性文件和目标属性文件』。
source	是用于描述源存储库的属性文件的相对路径或绝对路径。必须对 COPY 或 MOVE 操作指定 <i>source</i> 自变量。要获取样本目标文件，请参阅第 111 页的『样本源属性文件和目标属性文件』。
arguments	是第 111 页的『自变量』的表中指定的一个或多个自变量。

如果未指定 *operation*、*destination*、*source* 或 *arguments*，则存储库转换实用程序以交互方式运行。

操作描述： 下表描述了存储库转换实用程序的可用操作。

操作	描述
HELP	显示命令行帮助。
CHANGE	将活动存储库更改为目标存储库。
COPY	保留源存储库不变并在目标存储库中创建源存储库的副本。
DELETE	永久地删除目标存储库。
MOVE	删除源存储库并在目标存储库中创建新的存储库。
NEW CREATE	在目标位置创建新存储库并使用缺省存储库值来填充该存储库。

VERIFY	对目标存储库运行验证操作并报告结果。
自变量:	下表描述了存储库转换实用程序的自变量。
自变量	描述
DEBUG	输出附加的调试信息以帮助您诊断问题。
LOG:file	指定用于记录所有转换实用程序活动的转换实用程序日志文件的名称。 <i>file</i> 的缺省名称是 <i>repositoryconvert.log</i> 。
SERVER:InstanceName	指定 DB2 Alphablox 实例的实例名。SERVER 自变量是必需的。
OVERWRITE	OVERWRITE 自变量与 NEW、MOVE、COPY 或 CHANGE 操作配合使用，它删除目标存储库中的所有旧数据和结构并在原位置重新创建新结构和数据。未指定此自变量时，如果存储库转换实用程序在目标位置检测到现有的存储库，它就会停止。
PROPS:option	此选项与 SERVER:InstanceName 自变量配合使用，它指定要对目标存储库更新哪些服务器属性。 <i>option</i> 的值是： <ul style="list-style-type: none"> • ALL: 指定要转换所有服务器属性。 • GLOBAL: 指定仅转换由集群共享的属性（而不转换本地机器条目）。 • SPECIFIC: 指定仅转换特定于本地机器的服务器属性（而不转换集群属性）。
UPDATE	指定 MOVE 或 COPY 操作使用源存储库中的信息来更新目标存储库，而不是替换目标存储库的内容。
DDL:file	覆盖用来创建数据库存储库的表、索引和初始内容的缺省 DDL 模式文件。
	要点: 请您极为审慎地使用此自变量。
USEDEST	当指定了此自变量时，它将服务器属性修改为使用目标存储库。如果未指定此自变量，则不更改服务器的存储库属性。

样本源属性文件和目标属性文件: 本节提供与存储库转换实用程序配合使用的样本源文件和目标文件的内容。这两个属性文件指定了存储库的类型、存储库的连接信息、存储库属性文件在运行 DB2 Alphablox 的计算机上的位置以及 DDL 文件的名称。

以下示例显示了一个存储库的属性文件，该存储库驻留在名为 *oracle817* 的服务器上的 Oracle 数据库中：

```
RepositoryTarget=JDBCTarget
java.naming.factory.initial=com.alphablox.jndisp.AlphabloxContextFactory
java.naming.provider.url.server=oracle817
java.naming.provider.url.port=1521
java.naming.provider.url.sid=orcl817
database_driver=oracle.jdbc.driver.OracleDriver
```

```
fileroot=C:\alphablox\analytics\repository\servers\  
commandfile=oracle.dmlsql  
user=user  
password=password
```

以下示例显示了位于文件系统中的存储库的属性文件:

```
RepositoryTarget=ABXTarget  
java.naming.factory.initial=com.alphablox.jndisp.AlphabloxContextFactory  
java.naming.provider.url=C:\alphablox\analytics\repository\  
fileroot=C:\alphablox\analytics\repository\servers\  

```

第 17 章 使用连接池

本章描述如何配置 DB2 Alphablox 以将连接池与多维数据源和关系数据源配合使用。

连接池 - 概述

每当基于 Web 的应用程序需要与多维数据库或关系数据库进行交互时，它必须首先连接至该数据库。这些连接中的每一个连接都会带来开销，即需要使用资源来建立连接、维护该连接以及在不再需要该连接时释放它。在基于 Web 的应用程序中，用户与数据库的交互通常比较短暂，并且，连接数据库以及与数据库断开连接所花费的时间通常比请求本身所花费的时间更多。

为了更有效地处理这些数据库交互，应用程序服务器和数据库通常提供了连接池。连接池是一组可以与应用程序共享的预定义数据库连接，它允许进行所需的数据库交互，但消耗的时间和资源比单个连接少。

如果连接池可用并且已正确地配置了它，DB2 Alphablox 就可以利用连接池来改进应用程序的性能。DB2 Alphablox 连接池可以与用于关系数据库连接池的应用程序服务器（IBM WebSphere 和 BEA WebLogic）配合使用，也可以与 IBM DB2 OLAP Server Deployment Services 或 Hyperion Essbase Deployment Services 中的连接池支持配合使用。

MDB 连接池

在使用标准 JDBC 连接时，不支持多维数据库连接池，多维数据库连接池只能与关系数据源配合工作。对于 DB2 OLAP Server 或 Hyperion Essbase，DB2 Alphablox 可以使用 IBM DB2 OLAP Server Deployment Services 或 Hyperion Essbase Deployment Services 中提供的连接池实现。

DB2 OLAP Server 和 Hyperion Essbase 连接池

可以配置 IBM DB2 OLAP Server Deployment Services 或 Hyperion Essbase Deployment Services，以支持连接池。每个已定义的连接池都指定了要使用的 DB2 OLAP Server 或 Essbase 用户名和密码。可以为所有用户定义连接池访问权（“允许每个人”），也可以为一组指定的用户和组定义连接池访问权。如果 DB2 Alphablox 数据源指定了 IBM DB2 OLAP Server Deployment Services 或 Hyperion Essbase Deployment Services 适配器，并且在 IBM DB2 OLAP Server Deployment Services 或 Hyperion Essbase Deployment Services 中正确地配置了连接池，则 DB2 Alphablox 可以使用这些连接池。要了解有关配置 ESSBASE 连接池的详细信息，请参阅 IBM DB2 OLAP Server Deployment Services 或 Hyperion Essbase Deployment Services 的文档。

Microsoft Analysis Services 和连接池

在 Microsoft Analysis 实现过程中使用连接池有助于提高分析应用程序的性能。缺省情况下，禁用连接池。

当启用了连接池并调用了 `DataBlox disconnect()` 方法时，`DataBlox` 将断开连接，但 ADO 连接将不会关闭并将被置于连接池内以便复用。当后续 `DataBlox` 连接请求与池中的连接相匹配的连接时，将从池中除去此连接并将其指定给此 `DataBlox`。否则，将创建新的连接。合用的连接对数据源用户名、密码、提供程序字符串、目录（`Microsoft Analysis Services` 数据库名称）和模式（由于 `Microsoft Analysis Services` 不使用它，所以通常为空）是很关键的。所有五个属性都要匹配，`DataBlox` 组件才能使用池中的某个连接。在不同数据源上更改提供程序字符串会导致池中出现不同的唯一连接。本质上，这些唯一连接中的每个连接都是一个连接池键。例如，如果有两个 `user1` 至 `database1` 的连接都使用了缺省提供程序字符串“`MSOLAP`”，同时有另一个相同用户和数据库的连接使用了提供程序字符串“`MSOLAP;Default Isolation Mode=1;Execution Location=3;Client Cache Size=0;`”，则会有三个连接使用两个不同的连接池键。

启用连接池

缺省情况下，禁用 `MSOLAP` 连接池。要启用它，需要将 `MSOLAPEnableConnectionPool` 服务器属性设置为 `true`。

要启用连接池：

1. 使用 Telnet 控制台（或使用“`DB2 Alphablox` 管理页面”中提供的“管理控制台”窗口）来打开 `DB2 Alphablox` 服务器。
2. 将 `MSOLAPEnableConnectionPool` 服务器属性设置为“`set MSOLAPEnableConnectionPool true`”并按 `Enter` 键。会出现消息：“服务器属性‘启用 `MSOLAP` 连接池合用’已设置为 `true`。”
3. 输入“`save`”并按 `Enter` 键。会显示消息“服务器属性已保存”。

注：如果您不保存此属性，则只会在当前服务器会话中应用此服务器属性。

要禁用连接池，请重复上述步骤，但要將 `MSOLAPEnableConnectionPool` 属性设置为 `false`。

使用连接池

使用 `DB2 Alphablox Microsoft Analysis Services` 连接池时，要确保将所有 `DataBlox autoDisconnect` 属性设置为 `true`。如果不设置，直至终止用户会话后，连接才会返回到可用的池。另外，是否使用连接池将会使 `clearClientCache()` 方法的行为产生较大差异。如果在启用连接池时调用 `clearClientCache()` 方法，它会从其连接池键与 `DataBlox` 连接匹配的可用池中清除所有连接。另外，您也可使用 `ODBOBridgeJavaAPI.clearPool()` 方法清除整个连接池。此方法会清除可用池中的所有 ADO 连接。

限制连接池

对于具有极高并行性并带有较少唯一合用连接（用户名）的环境，您可通过限制池大小来提高内存使用效率和性能。已增加 `MSOLAPConnectionPoolLimit` 服务器属性，可用来限制 `DB2 Alphablox` 允许每个连接池键所具有的最大活动 `MSOLAP` 数。这包括池中的连接。使用此功能时应谨慎；它只能用于此处所描述的特殊环境。`MSOLAPConnectionPoolLimit` 属性限制每个连接池键的连接数。使用 `MSOLAPConnectionPoolLimit` 服务器属性来设置最大连接数。

例如，将 `MSOLAPConnectionPoolLimit` 设置为 4 将会使每个连接池键的打开的 `MSOLAP` 最大连接数设置为 4。在此示例中，假定与 `MSAS` 的连接都是相同的（`username` 属性除外），并且您只有 3 个连接的用户名（`Exec`、`Manager` 和 `Admin`）。

如果 DataBlox 组件尝试使用用户标识 Admin 来连接并且它是第五个连接，则需要等到该用户名被释放（或关闭）并放回池中，以便打开的连接数符合限制。

调整连接池

通过使用 DB2 Alphablox Microsoft Analysis Services 连接合用功能，您可调整连接池以优化内存使用和性能。需要考虑几个方面的因素。

Microsoft Analysis Services 客户端高速缓存： 在连接中通常认识不到 Microsoft Analysis Services 客户端高速缓存的影响。MSAS 可在 DB2 Alphablox 主机上高速缓存大量内容。例如，将会在 Alphablox 主机的 ADO 层中一直计算和高速缓存已计算的量度。如果要在 Alphablox 会话之间不断复用这些信息，则将包含这些信息的 ADO 连接保持为打开会很有帮助。

连接池注意事项： 当您在连接中使用了有限数量的连接池键（少量唯一用户名）或 DB2 Alphablox 会话有多个使用同一连接池键的 Blox 连接时，可使用连接池。

如果（需要唯一连接池键的）所有用户都是唯一的并且对于每个连接池键，您只有一个 DataBlox，或者，如果您有大量连接池键并且在终止用户会话时无法清除它们，则不要使用连接池。

使用 autoDisconnect 属性： 在使用连接池时，通常，您应该始终使用 DataBlox autoDisconnect 属性。否则，ADO 连接将不返回到池中以供复用。

使用 clearClientCache() 方法： 如果您正在使用大量连接池键（即，每个与 Microsoft Analysis Services 的连接都是一个具体的指定用户），则在用户完成会话时，您将会希望对每个具有唯一连接池键的 DataBlox 组件调用 DataBlox clearClientCache() 方法。一般情况下，通过注销页来完成此操作，当用户明确注销或关闭其浏览器窗口时会调用此页。

使用 clearPool() method： 在多数情况下，定期清除连接池可能会有帮助。对于多数环境来说，仅使用简单计时器 servlet 来每晚清除连接池可能就足够了。

监视和管理连接池： 可在 ODBOBridgeJava 类中使用下列静态方法以监视和管理连接池：

ODBOBridgeJavaAPI.poolSize()

返回可用池中的连接数

ODBOBridgeJavaAPI.clearPool()

断开可用池中的连接

ODBOBridgeJavaAPI.getMaximumNumberOfConnections()

获取可打开的最大 MSAS 连接数

ODBOBridgeJavaAPI.getNumberofOpenConnections()

获取当前打开的 MSAS 连接数

打开的 ADO 连接数等于可用池中的连接数与正使用的连接数之和。

下列 JSP scriptlet 示例使用上述方法来管理和监视连接池：

```
<% ODBOBridgeJavaAPI.setMaximumNumberOfConnections(15);
   int count = ODBOBridgeJavaAPI.poolSize();
   out.write("Number of connections in the pool: "+count+"<br>");
   count = ODBOBridgeJavaAPI.getNumberofOpenConnections();
```

```

out.write("Number of connections active: "+count+"<br>");
count = ODBOBridgeJavaAPI.getMaximumNumberOfConnections();
out.write("Connection limit: "+count+"<br>");
%>

```

RDB 连接池

RDB 连接池是指定的一组相同的数据库的 JDBC 连接，这些连接是在应用程序服务器上注册该连接池时创建的。基于 Web 的应用程序可以从池中“借用”一个连接，在与数据库进行交互期间使用此连接，然后通过关闭此连接来将此连接返回到池中。

下表强调了使用连接池的益处：

益处	描述
缩短了响应时间	每当一个资源尝试访问数据库时，它都必须连接至该数据库。当连接池启动时，它将创建指定数目的物理数据库连接，从而消除了为每个应用程序创建数据库连接的开销。通过消除创建、维护和释放连接的开销，连接池显著缩短了结果集检索操作的响应时间。
底层数据库抽象	连接池提供了底层数据库的抽象层，这使您可以更方便地切换应用程序数据库，而不必处理特定于供应商的 SQL 异常。您只需处理应用程序服务器的连接池异常。
并发连接管理	如果您由于许可证发放限制而只有有限数目个数据库连接，则可能能够使用连接池来管理并发数据库连接的数目。

DB2 Alphablox 对 RDB 连接池的使用

如果正在将 WebSphere 和 WebLogic 应用程序服务器与 DB2 Alphablox 配合使用，则下列各项可以利用连接池：

- DB2 Alphablox 数据源
- 基于数据库的 DB2 Alphablox 存储库
- RDB 立方体
- ReportBlox
- CommentsBlox
- JDBC 连接 bean

上面列出的所有 DB2 Alphablox 功能部件，除了 DB2 Alphablox 数据源以及任何基于关系数据库的 DB2 Alphablox 存储库以外，都会自动地利用 RDB 连接池（如果有 RDB 连接池可用并且正确地配置了 RDB 连接池的话）。

DB2 Alphablox 数据源与 RDB 连接池

要将 RDB 连接池与 DB2 Alphablox 以及 WebSphere 或 WebLogic 应用程序服务器配合使用，必须在“DB2 Alphablox 数据源定义”上选择“应用程序服务器数据源适配器”选项。仅当将 DB2 Alphablox 配置为使用受支持的 WebSphere 和 WebLogic 应用程序服务器时，这个适配器选项才会出现。

注：当使用 WebSphere 或 WebLogic 连接池来访问数据源时，可以将 DB2 Alphablox 与应用程序服务器上的任何受支持关系数据库配合使用。

注：DB2 Alphablox 的 Apache Tomcat 配置不支持 RDB 连接池。

DB2 Alphablox 存储库与 RDB 连接池

要对基于关系的 DB2 Alphablox 存储库启用或禁用 RDB 连接池，请执行下列步骤：

启用

1. 为应用程序服务器上已定义的连接池创建 DB2 Alphablox 数据源。请参阅以上指导以了解详细信息。
2. 运行存储库转换实用程序。

Windows:

```
<serverDirectory>/tools/convert/ConvertRepository.exe
```

Linux 和 UNIX:

```
<serverDirectory>/tools/convert/ConvertRepository.bat
```

3. 如果 DB2 Alphablox 正在使用 WebSphere 或 WebLogic，则选项 8（“配置 Web 应用程序服务器连接池”）将可用。选择此选项，然后按 Enter 键。
4. 选择选项 1（“打开连接池”），然后按 Enter 键。
5. 输入应用程序服务器中定义的数据源的 JNDI 名称。

禁用

1. 运行存储库转换实用程序。
2. 如果 DB2 Alphablox 正在 WebSphere 或 WebLogic 上运行，则选项 8（“配置 Web 应用程序服务器连接池”）将可用。选择此选项，然后按 Enter 键。
3. 选择选项 1（“关闭连接池”）。

对 BEA WebLogic 配置连接池

当对关系数据源使用 BEA WebLogic 连接池时，BEA WebLogic 要求为每个数据源创建一个 WebLogic 用户以便访问已定义的连接池。

要正确地配置 DB2 Alphablox 以使用 WebLogic 连接池，DB2 Alphablox 数据源定义“缺省用户名”和“缺省密码”必须是 WebLogic 用户。如果未使用连接池，则不需要创建 WebLogic 用户就可以访问关系数据源。

第 18 章 使用集群环境

本章描述如何在集群环境中配置和使用 DB2 Alphablox 以便支持可伸缩的分析应用程序。讨论了两个集群环境: WebSphere 集群环境和 WebLogic 单机集群环境。

- 第 119 页的『集群环境概述』
- 第 119 页的『WebSphere 集群环境』
- 第 119 页的『WebLogic 集群环境』
- 第 120 页的『集群控制台命令』

集群环境概述

为了提高可伸缩性和可用性, DB2 Alphablox 可以在两台或更多台服务器的集群中运行。集群中的服务器越多, 系统可以支持的用户数目就越多。集群提供了两项关键功能:

- **可伸缩性:** 集群的容量不限于单一机器。可以动态地将新服务器添加到集群中以提高容量。
- **高可用性:** 集群使用多台服务器的冗余来将客户机与故障隔离。

注: 如果将 DB2 Alphablox 配置设置为在集群环境中运行, 则存储库必须位于关系数据库中。请参阅 第 105 页的『DB2 Alphablox 存储库概述』以了解更多关于 DB2 Alphablox 存储库以及使用关系数据库来对其进行配置的信息。

WebSphere 集群环境

要了解有关在 WebSphere 集群环境中配置和安装 DB2 Alphablox 的信息, 请参阅《安装指南》中的『在 WebSphere 集群环境中使用 DB2 Alphablox』。要了解有关配置和使用 WebSphere 集群的详细信息, 请参阅 WebSphere 服务器文档。

WebLogic 集群环境

WebLogic 集群是一组 WebLogic 服务器, 这些服务器一起工作以提供单服务器环境不能提供的可伸缩性和可靠性。虽然集群对于客户机来说是作为单一服务器出现的, 但是, 集群实际上是一组服务器。这些 WebLogic 集群可以是同一物理机器上的多个实例, 也可以是安装在多台物理机器上的 WebLogic 服务器。

每台 DB2 Alphablox 主机都与集群中的所有其他节点通信以便在存储库中的任何内容发生更改时相互通知。对存储库所作的任何更改(例如, 新用户)都直接存储在存储库中, 每个节点都可以读取和修改存储库。

在 WebLogic 集群环境中配置和安装 DB2 Alphablox

要了解有关在 WebLogic 集群环境中配置和安装 DB2 Alphablox 的信息, 请参阅《安装指南》中的『在 WebLogic 集群环境中使用 DB2 Alphablox』。要了解有关配置和使用 WebLogic 集群的详细信息, 请参阅 WebLogic Server 文档。

在 WebLogic 集群环境中创建新应用程序

要了解有关在 WebLogic 集群环境中定义新的 DB2 Alphablox 应用程序的指示信息，请参阅第 32 页的『当使用 WebLogic 集群时定义应用程序』。

使用 WebLogic 垂直集群

通过使用垂直集群配置，可以改进与 Microsoft Analysis Services 配合工作的 DB2 Alphablox 应用程序的可伸缩性，并可以在单一机器上主管多个服务器实例以提高效率和节省成本。

对于 Microsoft Analysis Services 用户，WebLogic 垂直集群通过在一台机器上运行多个 DB2 Alphablox 实例，使 DB2 Alphablox 应用程序不必受到 Windows 操作系统上 Microsoft Analysis Services 2000 进程 2 GB 限制的约束。这可以改进 Microsoft Analysis Services 的可伸缩性。

当您想要利用 IT 投资并减少与使用多台机器相关联的维护问题时，也可以在功能强大的单一机器上使用垂直集群。

要了解有关在 WebLogic 集群环境中配置和安装 DB2 Alphablox 的信息，请参阅《安装指南》中的『在 WebLogic 集群环境中使用 DB2 Alphablox』。要了解有关配置和使用 WebLogic 集群的详细信息，请参阅 WebLogic Server 文档。

集群控制台命令

除了管理用户界面以外，您还可以使用 DB2 Alphablox 控制台来输入**集群选项**页面上的信息。要显示当前集群属性设置，请在 DB2 Alphablox 控制台窗口中输入以下命令：

```
get service cluster
```

控制台上的屏幕输出类似于：

```
get service cluster
服务集群管理器属性:

IsClustered ..... false
  (启用了集群吗 [true|false])
LeadHost .....
  (集群中的前导 AAS 主机的名称或 IP 地址)
PortNum ..... 7855
  (前导主机侦听集群消息时使用的端口号)
MaxHosts ..... 10
  (集群中的最大 AAS 主机数)
StartupTime ..... 60
  (每个正常节点在连接至集群时要等待的时间(秒))
LiveIsClustered .... false
  (当前已启用集群)
LiveLeadHost .....
  (前导 AAS 主机的当前名称或 IP 地址。)
LivePortNum ..... 7855
  (前导主机正在侦听的当前端口号。)
LiveMaxHosts ..... 10
  (集群中允许的当前最大 AAS 主机数)
LiveStartupTime .... 60
  (每个正常节点在连接至前导主机时要等待的当前时间)
```

下表列出了集群控制台命令以及每个命令的功能描述:

命令语法	描述
cluster shutdown	关闭 DB2 Alphablox 集群中的所有实例。可以从集群中的任何主机的控制台运行 CLUSTER SHUTDOWN 命令。
get service cluster	列出集群属性的当前设置。以“Live”开头的属性（例如 LiveIsClustered）返回集群管理器所使用的实际值；用户不能配置它们。其他属性可以使用 SET SERVICE CLUSTER 命令进行设置，如下所述。
set service cluster <property> <value>	将集群属性的值更改为指定的值。 <i>property</i> 的有效条目是： IsClustered LeadHost PortNum MaxHosts StartupTime 新设置将在集群管理器重新启动后生效（通常是在 DB2 Alphablox 重新启动后生效）。
set service cluster IsClustered <True False>	将集群的状态设置为 <i>enabled</i> 或 <i>disabled</i> 。当设置为 true 时，即启用集群。当设置为 false 时，即禁用集群。
set service cluster LeadHost <lead_host>	设置前导主机名或 IP 地址 (<i>lead_host</i>)。在网络中，此名称或 IP 地址必须解析为安装了前导主机的机器。
set service cluster PortNum <port_number>	设置前导主机侦听集群通信时使用的端口号 (<i>port_number</i>)。在前导主机上，此端口必须可用，否则启动集群时将失败。
set service cluster MaxHosts <maximum_hosts>	设置集群所允许的最大主机数 (<i>maximum_hosts</i>)，包括前导主机。在达到 <i>maximum_hosts</i> 之后，将拒绝任何尝试加入集群的正常主机进入集群。
set service cluster StartupTime <startup_time>	设置正常节点在启动后尝试加入集群的秒数 (<i>startup_time</i>)。在超出 <i>startup_time</i> 之后，正常节点主机将自动关闭。
show hosts	列出当前连接至集群的所有主机。如果禁用了集群，则 SHOW HOSTS 不返回任何内容。

第 19 章 DB2 Alphablox 控制台命令

可以通过 DB2 Alphablox 的控制台或者从 DB2 Alphablox 主页的**管理**选项卡中的 Web 页面来管理它。大部分通过 DB2 Alphablox 主页用户界面配置的管理活动（例如，创建和编辑用户以及创建和编辑组等等）也可以通过控制台命令来配置。本章提供了有关使用控制台的信息并列出了可用的控制台命令。

访问控制台

可以通过两种方法来访问 DB2 Alphablox 控制台：

- 『HTML 控制台』
- 『Telnet 控制台』

HTML 控制台

要从 DB2 Alphablox 主页中访问控制台，请单击**管理**选项卡 > **常规链接** > **启动 DB2 Alphablox 控制台**链接。控制台 HTML 页面打开。

管理员也可以通过从 Windows **开始**菜单中启动 DB2 Alphablox 时启动的“控制台”窗口或者通过 DB2 Alphablox 的启动窗口（在 Linux 和 UNIX 系统上）来在 Telnet 会话中打开控制台。

Telnet 控制台

要访问 Telnet 控制台，请对正在运行 DB2 Alphablox 的机器启动 Telnet 会话，并指定 **Telnet 控制台**管理页面上配置的 Telnet 端口（缺省端口为 23）。例如，如果正在名为 *pastrami* 的机器上运行 DB2 Alphablox，则在命令提示符处输入以下命令：

```
telnet pastrami 23
```

在 Windows 机器上，安装过程将把 Telnet 控制台的快捷方式放在“**开始**”菜单 > **所有程序组** > **DB2 Alphablox** 文件夹 > 实例名文件夹（缺省值是 **AlphabloxAnalytics**）> 控制台中。

输入控制台用户名和密码以对 Telnet 控制台进行认证。

要结束 Telnet 控制台会话，输入以下命令：

```
release
```

命令语法

DB2 Alphablox 命令使用以下语法：

```
COMMAND object [value(s)] COMMAND object [value(s)] object [value(s)]
```

其中：

- object 是下列其中一项：
 - SERVER（缺省值）

- CONSOLE *console ID* (其中, *console ID* 是特定活动控制台的标识, 例如 C1)
- 服务器对象的名称。当在非英语语言环境中运行时, 允许的字符是 A-Z、a-z、0-9、下划线或特殊字符 (例如, 重音字符)。名称的显示是区分大小写的, 但认证的实际名称不区分大小写。名称 *Public*、*Private* 和 *Properties* 已被保留, 不能用作对象名。
- *value* 进一步限定对象

例如, 在下面这一行中, *get* 是命令, *user* 是服务器对象的名称, 而 *admin* 是特定用户对象的名称。

```
get user admin
```

如果输入后面跟随着对象的命令, 则将显示一条消息, 该消息显示了该命令所需的值。例如, 如果输入以下命令:

```
create data source
```

则将显示以下消息:

创建数据源使用多个参数:

```
data_source_name adapter_name SERVER server_name [property value]...
```

属性名是相应用户界面页面上的名称。例如, 如果您正在创建 DB2 OLAP Server 或 Hyperion Essbase 数据源, 则它具有属性名 *application*、*database*、*username*、*password*、*maxrows*、*maxcols* 和 *useaasuserauth*。

命令缩写

为了提高输入命令的速度, 控制台支持命令缩写。例如, H 是 HELP 命令的缩写。如果两个或更多个命令以相同的字母开头, 则缩写的长度必须足以保证该命令唯一。例如, START 和 STATISTICS 命令都是以 STA 开头的; 添加下一个字母将创建唯一的缩写 (STAR 表示 START, STAT 表示 STATISTICS)。

技巧: 注意下列命令使用技巧:

- 控制台对整个命令行语法支持缩写。例如, H C 将检索 CREATE 命令的帮助; G U L 将检索名称以 L 开头的用户的属性。
- 除数据库适配器的名称以外, 不必将值字符串用引号引起来 (例如, “ibm db2 olap server”)。
- 当使用一般属性的缩写时, 使用属性名的前几个字母, 而不是使用它们的描述。例如, 对于属性名 DefaultMessageLevel, 其描述为 Initial Console Message Level, 将使用诸如 DEFAULTM 之类的缩写。

控制台命令列表

大部分命令既可以在“控制台”屏幕 (通过管理选项卡中的常规页面上的启动 **DB2 Alphablox** 控制台链接访问此屏幕) 上输入也可以在 DB2 Alphablox 命令屏幕中输入。但是, 少数命令只能从这两个屏幕中的其中一个屏幕中输入。另外, Telnet 并不支持所有命令。要获取 DB2 Alphablox Cube Server 控制台命令列表, 请参阅《DB2 Alphablox Cube Server 管理员指南》。要获取特定于集群的命令列表, 请参阅第 120 页的『集群控制台命令』。要获取特定于可扩展用户管理器的命令列表, 请参阅第 89 页的『可扩展用户管理器 Telnet 控制台命令』。

注：以下控制台命令仅适用于 DB2 Alphablox。例如，如果从 WebSphere 服务器上安装的 DB2 Alphablox 删除应用程序，则只能从 DB2 Alphablox 除去该应用程序列表。要从 WebSphere 服务器除去该应用程序，必须使用 WebSphere Application Server 管理控制台除去它。

命令和示例	用途和解释
ADD object value object value	将第二个对象添加到第一个对象
add group admin user radams	将名为 radams 的用户添加到名为 admin 的组中
CREATE object value(s)	创建对象（如用户或组）。
create user radams haggis	创建名称值为 radams 并且密码值为 haggis 的新用户
create data source TBC ibm db2 olap server adapter server db2erver1 application demo database basic username user1 password password1 maxrows 1000 maxcols 1000 useasuserauth false	创建名为 TBC 的数据源，它访问名为 db2server1 的 IBM DB2 OLAP Server 上的 TBC (demo/basic) 应用程序。
DELETE object value	从 DB2 Alphablox 和 DB2 Alphablox 存储库中删除对象。从 DB2 Alphablox 中删除应用程序不会从 WebSphere 服务器中除去相应的应用程序。
delete user radams	
delete outlinecache...	从 DB2 Alphablox 中删除名为 radams 的用户。 从 DB2 OLAP Server 或 Essbase 高速缓存中删除条目
EXIT	停止 DB2 Alphablox 并退出“控制台”屏幕。只能从 DB2 Alphablox 命令屏幕中执行此操作；不能从通过 DB2 Alphablox 主页打开的“控制台”屏幕中执行此操作。
ExtUserManager	要获取可扩展用户管理器控制台命令，请参阅第 89 页的『可扩展用户管理器 Telnet 控制台命令』。
GET object value	显示有关对象的信息
get	显示有关 Server 的信息（因为 Server 是缺省对象）
get user radams	显示名为 radams 的用户的属性
HELP object	显示指定的命令的帮助信息
help	显示所有命令的列表
help create	显示 create 命令的描述和语法
KILL object	杀死指定的对象（通常是用户会话）的当前实例
kill 1046976892932726235	杀死标识为 1046976892932726235 的会话
LOAD object value	将对象装入到服务器工作空间中
load user <userName>	将用户 <userName> 装入到服务器工作空间中
load theme	将所有可用的主题装入到服务器工作空间中。（注意，此命令不支持仅装入指定的主题。）
load license	装入（更新）位于 DB2 Alphablox bin 目录中的许可证文件（license.xml）。

命令和示例	用途和解释
LOCK	锁定对服务器的本地访问。要在输入此命令之后访问 DB2 Alphablox，需要输入用户名和密码。只能从 DB2 Alphablox 的“控制台”屏幕和 Telnet 中执行此操作。
MESSAGE object value(s)	将系统消息发送到其他控制台或日志文件
message review 20	将最后 20 条消息发送到当前控制台
message review debug 20	将最后 20 条 DEBUG 消息发送到当前控制台
message C2 Testing	将消息“Testing”发送到控制台 C2
OBJECTS object	显示有关可管理对象的信息
objects	显示可管理对象的层次结构
objects user	显示有关用户的信息
RELEASE	释放用于输入此命令的远程服务器控制台。只能从 DB2 Alphablox 的“控制台”屏幕中执行此操作。
REMOVE object object	从目标对象中除去源对象。
REMOVE group admin user radams	从组 admin 中除去用户 radams。
REPORT object	将当前控制台的 消息级别设置为
report debug	DEBUG、VERBOSE、INFO、SYSTEM、WARNING、ERROR 或 FATAL。 将消息级别设置为 DEBUG。
RESUME object	从暂挂状态启动指定的服务。
resume user	从先前暂挂状态启动用户管理器。
RUN object	运行由目标命名的命令文件。
run abc.console	运行 abc.console 命令文件。
run createUsers.txt	运行 createUsers.txt 文件。（此文件可以是一个批处理文件，它包含一系列用于创建新用户以及确定他们的初始密码的创建用户命令。请参阅『注 2』以获取示例。）
SAVE object value	保存目标的属性
save	立即保存所有 General 属性（因为 General 是缺省对象）。在控制台上输入的更改只对当前会话有效，除非发出此命令或将 DB2 Alphablox 的“退出时保存”属性设置为“是”，否则这些更改将丢失。
save user radams	保存用户 radams 的属性
SET object value object value	对对象设置值。
set user radams password castle	将用户 radams 的密码设置为 castle。
set smtpserver mail	将 SMTP 服务器设置为名为 mail 的邮件服务器。
set resolveAliasesToBaseMembers	

命令和示例	用途和解释
SHOW object	获取目标的信息。
show topics	列出所有可以获得其信息的主题。
show server	显示所有服务器主题，包括用户。
show user radams	显示有关名为 radams 的用户的用户的信息。
show theme	显示主题名和描述。
show outlinecache	显示 DB2 OLAP Server 或 Essbase 高速缓存中的所有条目。
show hosts	显示集群中的每台机器的名称（如果启用了集群的话）。
show	显示对象的列表。SHOW 命令还显示（如果它尚未被装入的话）已装入的 DB2 OLAP Server 或 Hyperion Essbase 客户机 API 的版本。
START object	从停止状态启动服务。
start user	从先前停止状态启动用户管理器。
STATISTICS object value	显示关于目标的可用统计信息。
statistics user	显示所有用户的统计信息。
statistics user radams	显示用户 radams 的统计信息。
STOP object	停止正在运行的或已暂挂的服务。使用 START 来使服务结束停止状态。
stop user	注： 除非所有运行的 DB2 Alphablox 立方体都已停止，否则不要使用 STOP 命令来停止立方体管理器。 停止用户管理器。
SUSPEND object	暂挂服务。例如，暂挂用户管理器将导致无法将新的用户会话实例化。使用 RESUME 来使服务结束暂挂状态。
suspend user	暂挂用户管理器

特定于 Essbase 的控制台命令

DB2 Alphablox 存储在内存中高速缓存的 DB2 OLAP Server 和 Hyperion Essbase 轮廓以提高性能。本节描述两个用来管理轮廓高速缓存的命令的语法以及用来将别名解析为基本成员名的命令的语法。包括下列命令：

- 第 127 页的『RESOLVEALIASSTOBASEMEMBERS 命令』
- 第 128 页的『SHOW OUTLINECACHE 命令』
- 第 128 页的『DELETE OUTLINECACHE 命令』

RESOLVEALIASSTOBASEMEMBERS 命令

在 DB2 OLAP Server 或 Essbase 数据库中，可以将别名设置为成员名。RESOLVEALIASSTOBASEMEMBERS 服务器属性被设计成与书签配合使用，这些书签将查询

与别名保存在一起，这些别名在不同的时候可能解析为不同的成员。例如，如果对引用别名“本季度”的查询建立书签，则该查询今天在 DB2 OLAP Server 或 Essbase 数据库中所引用的实际成员与它 6 个月前引用的成员不同。如果要让该书签不再检索本季度的当前数据，则必须将 RESOLVEALIASESTOBASEMEMBERS 服务器属性设置为 TRUE。

以下是 RESOLVEALIASESTOBASEMEMBERS 控制台命令的语法：

```
set resolvealiasestobasemembers true | false
```

以下命令返回 RESOLVEALIASESTOBASEMEMBERS 服务器属性的当前状态：

```
get resolvealiasestobasemembers
```

```
ResolveAliasesToBaseMembers .... false  
  (ResolveAliasesToBaseMembers)
```

要启用此功能，请输入以下控制台命令：

```
set resolvealiasestobasemembers true
```

技巧：通过输入足够的字母确保名称唯一，可以使用此命令的缩写，如下所示：

```
set resolve true
```

SHOW OUTLINECACHE 命令

以下是用于 DB2 OLAP Server 或 Hyperion Essbase 数据源的 SHOW OUTLINECACHE 命令的语法：

```
show outlinecache [essbasecachemanager [entry [entryname]]]
```

使用此命令来显示有关高速缓存管理器或它管理的条目的信息。例如，使用以下命令来显示 DB2 OLAP Server 或 Essbase 高速缓存管理器所管理的所有条目：

```
show outlinecache essbasecachemanager entry
```

系统将作出以下响应：

```
条目  
  条目 MDB1.Financial.Current  
    ....总访问次数: 7  
    ....当前访问人数: 1
```

该响应显示 DB2 OLAP Server 或 Essbase 高速缓存管理器当前只管理了一个名为 MDB1.Financial.Current 的条目。此高速缓存总共已被访问了 7 次（累积数目），并且当前正被一个连接使用。

DELETE OUTLINECACHE 命令

以下是用于 DB2 OLAP Server 或 Hyperion Essbase 数据源的 DELETE OUTLINECACHE 命令的语法：

```
delete outlinecache [essbasecachemanager [entry [entryname]]]
```

使用此命令来从高速缓存中除去指定的条目。仅当当前连接数（访问人数）为 0 时，DB2 Alphablox 才会除去该条目。除去条目时，将释放与该条目相关联的所有内存资源。例如，使用以下命令来从高速缓存中删除名为 MDB1.Financial.Current 的条目：

```
delete outlinecache essbasecachemanager entry MDB1.Financial.Current
```

系统将作出以下响应，这显示已删除了那个指定的条目：

MDB1.Financial.Current
已删除高速缓存条目。

以下命令及其响应确认该条目已不再位于高速缓存中:

```
show outlinecache essbasecachemanager
```

Essbase 高速缓存管理器
总条目数: 0

有关控制台命令的说明

本节描述一些其他行为并提供一些 DB2 Alphablox 控制台使用技巧。包括下列主题:

- 第 129 页的『查看一般属性』
- 第 129 页的『消息级别』
- 第 130 页的『通过控制台运行文本文件』
- 第 130 页的『DB2 Alphablox 日志消息』

查看一般属性

使用 DB2 Alphablox 控制台来查看一般属性和对象。大多数属性更改会立即生效; 否则, 将显示一条消息以指示在更改生效之前必须重新启动 DB2 Alphablox。要获取当前 DB2 Alphablox 属性的完整列表, 请使用 GET 命令。

控制台底部的链接提供了对常用信息的访问:

- 帮助显示 DB2 Alphablox 命令的列表。
- 状态提供有关 DB2 Alphablox 的状态信息。
- 消息显示 DB2 Alphablox 日志的可滚动视图。
- 会话显示有关这个 DB2 Alphablox 实例上的所有当前会话的状态信息。
- 用户显示用户列表 (无论他们是否已登录)。
- 服务显示 DB2 Alphablox 的组件状态。要获取服务列表, 请参阅第 7 页的『DB2 Alphablox 体系结构』。
- 设置显示每个 DB2 Alphablox 属性 (包括启动属性和扩展属性) 的当前设置。
- 显示显示 DB2 Alphablox 主题的分层列表。
- 历史记录显示通过此控制台发出的命令的历史记录。

要了解有关访问控制台的信息, 请参阅第 123 页的『访问控制台』。

消息级别

可以将 DB2 Alphablox 生成的消息发送至所有活动控制台和活动日志文件。消息信息包括:

- 日期和时间
- 消息级别
- 消息文本
- 消息源 (服务或用户名)

以下是一条典型的消息:

```
01/9/99 12:25:33 PM [VERBOSE] 请求 4223: 正在处理请求  
'/servlet/AnalysisServer/console/consolestyle.css' [会话 41, 用户 radams]
```

按严重性顺序 (最低到最高) 排列的消息级别是:

- DEBUG: 调试信息。
- VERBOSE: 所有系统消息。
- INFO: 不要求管理员执行操作的不重要系统事件。
- SYSTEM: 正常的系统事件，如创建用户。
- WARNING: 可恢复错误，但建议管理员可能要进行调查。
- ERROR: 不可恢复的错误。
- FATAL: 导致服务器终止的错误。

通过控制台运行文本文件

通过使用 `run` 命令，控制台可以执行包含任意数目的 DB2 Alphablox 命令的纯文本文件。例如，假定有一个名为 `d:\CreateUser.txt` 并包含下列各行的文件：

```
create user radams blue
create user sadams green
create user lplanting purple
create user klawrence yellow
create user dmessink orange
```

通过从控制台中执行以下命令，可以通过控制台使用 `run` 命令来运行此文件，以便创建 5 个新用户并确定他们的初始密码：

```
run d:/CreateUser.txt
```

注：文件名必须是标准名称。

DB2 Alphablox 日志消息

在 DB2 Alphablox 日志文件（缺省情况下，是 `Server.log`）中可能会出现如下所示的无害消息：

```
[VERBOSE] 请求 36: 找不到文件 "...resources\basic_en_US.class"
[VERBOSE] 请求 37: 找不到文件 "...resources\basic_en_US.properties"
[VERBOSE] 请求 38: 找不到文件 "...resources\basic_en.class"
[VERBOSE] 请求 39: 找不到文件 "...resources\basic_en.properties" ...
[VERBOSE] 请求 45: 找不到文件 "...resources\meta1_en_US.class"
[VERBOSE] 请求 46: 找不到文件 "...resources\meta1_en_US.properties"
[VERBOSE] 请求 47: 找不到文件 "...resources\meta1_en.class"
[VERBOSE] 请求 48: 找不到文件 "...resources\meta1_en.properties"
...
```

在此示例中，JVM 正在尝试查找对于客户机语言环境的语言和位置来说最合适的资源束。这些资源束对于正确运行 DB2 Alphablox 来说不是必需的；可以安全地忽略这些消息。

第 20 章 管理 Alphablox FastForward 应用程序

本章描述如何管理 Alphablox FastForward 应用程序，包括配置任务和基本管理任务。

概述

Alphablox FastForward 是预安装在 DB2 Alphablox 上的样本应用程序框架，它用来在业务机构内快速地开发、部署和共享定制分析视图。FastForward 框架提供了现成的公共应用程序服务，这些服务包括安全性、协作、定制和个性化。应用程序管理员（通常是 OLAP 管理员）可以创建新版本的 FastForward 应用程序，通过选择报告模板和配置报告参数来发布报告，然后部署新的应用程序，而不必查看任何代码。并且，由于它十分灵活并且可以扩展，所以 JSP 开发者可以修改或扩展应用程序框架，并且可以添加新的定制报告模板以供应用程序管理员配置和部署。

报告应用程序和分析应用程序中的常见功能已被构建到 FastForward 应用程序框架中，这些功能包括：

- 导出到 Microsoft Excel
- 生成可打印的视图
- 方便地保存和共享个人数据视图
- 通过电子邮件将视图发送给别人
- 方便地在不同视图之间进行导航

FastForward 用户的角色

Alphablox FastForward 用户的三种主要角色包括应用程序管理员角色、模板开发者角色和用户角色。这三个小组之间的良好配合将有助于确保基于 FastForward 的应用程序获得成功。下面简要描述关于这三种角色的更多信息。

应用程序管理员

应用程序管理员（通常是 OLAP 管理员）应该能够通过定义几项设置来创建新版本的 FastForward 应用程序、根据可用的报告模板来创建报告并接着快速地部署解决方案以供用户使用。如果使用现有的报告模板无法满足用户的需求，则应用程序管理员将与模板开发者配合工作以创建新的报告模板。应用程序管理员应该能够利用他们的 OLAP 数据库经验以及这里的文档来完成他们的工作（在 FastForward 应用程序的“管理任务”方式下，联机管理帮助也提供了这些文档）。

模板开发者

模板开发者通常是 JSP 开发者，他们主要负责在应用程序管理员无法使用现有模板来配置所请求的报告时创建定制报告模板。通过与应用程序管理员和用户协商，模板开发者应该能够通过修改现有报告模板或根据需要创建新的报告模板来创建新的报告模板。

通过使用 Blox 标记库、服务器端 Java API 和客户端 JavaScript API 以及您的 Web 编程经验，模板开发者应该能够创建模板来满足几乎每一项能够想象到的需求。除了熟悉构建 DB2 Alphablox 应用程序和视图以外，开发者还应该熟悉 FastForward 用户

帮助（在用户方式下，通过“帮助”按钮获得此帮助）和管理员帮助（在“管理任务”方式下，通过“帮助”按钮获得此帮助）。

用户

用户（通常是机构中的业务分析员和其他业务用户）应该能够登录到 FastForward 应用程序中并使用已发布的报告来分析业务问题。根据基于 FastForward 的特定应用程序所提供的交互性，用户可以处理数据、对数据层次结构进行钻取、更改图表类型、添加注释和执行其他操作。在修改视图以回答特定的业务问题之后，用户可以保留他们的当前视图，这可以通过在“专用”选项卡中创建已保存的报告以供将来使用来完成，也可以通过在“组”选项卡中与已定义的应用程序用户组共享那些报告来完成。

对于每个报告，用户通常可以从位于报告上方的应用程序工具栏中选择几个其他选项。除了保存报告以便进行联机分析以外，“导出到 Excel”选项还允许用户将视图导出到 Microsoft Excel 电子表格以便将来进行脱机分析。用户还可以使用“打印预览”选项来打印特定视图的副本。并且，如果他们愿意的话，则可以打开一条包含当前视图链接的电子邮件消息、添加注释并将该消息发送给其他应用程序用户。

如果用户的应用程序未提供所需的报告，他们通常直接向应用程序管理员请求提供新报告。

FastForward 应用程序的系统需求

除了《安装指南》的『系统需求』一节中指定的需求以外，请注意，Alphablox FastForward 只能与 DHTML 客户机配合工作。

创建 Alphablox FastForward 应用程序

在使用管理员权限登录之后，可以通过执行下列步骤来创建新版本的 Alphablox FastForward 应用程序：

1. 打开 Web 浏览器并浏览至“DB2 Alphablox 管理页面”。缺省情况下，将显示“应用程序”页面。
2. 在“应用程序”页面上，启动 Alphablox FastForward 应用程序。
3. 在应用程序工具栏上，单击“管理任务”按钮。
4. 将显示“模板应用程序”对话框窗口，该窗口提供了两个选项，即“创建”或“继续”。请单击“创建”按钮。
5. 在显示的“新建应用程序”窗口中，填写下列字段：

输入字段	描述
上下文名称	输入应该用作目录名的应用程序上下文名称。此名称不应该包含空格。
显示名称	显示在“应用程序”页面中的显示名称。此名称可以包含空格。
描述	可选。对应用程序的简要描述。
管理员角色	必需。在应用程序的 config.xml 文件中设置 AdministratorRole 值（该文件位于应用程序的 WEB-INF 目录中）。缺省设置为 AlphabloxAdministrator。

要点: 不要配置或使用现有的 Alphablox FastForward 应用程序 - 始终创建新版本。服务器以后进行的每一次升级都会覆盖这个样本应用程序, 您所作的修改将丢失。

更改管理员角色

缺省情况下, Alphablox FastForward 应用程序将 AdministratorRole 指定给 AlphabloxAdministrator。要更改可以管理特定应用程序的角色, 请在上面的“新建应用程序”对话框中进行设置, 或者执行下列步骤:

1. 打开 FastForward 应用程序的 config.xml 文件, 该文件位于以下位置:

```
<applicationDirectory>/WEB-INF/config.xml
```

2. 在 config.xml 文件中, 在下面这一节中更改指定的角色, 即将 AlphabloxAdministrator 替换为被指定来负责管理这个特定应用程序的新角色:

```
<param>          <param-name>AdministratorRole</param-name>
<param-value><![CDATA[AlphabloxAdministrator]]></param-value> </param>
```

要了解有关角色定义以及有关创建新角色的信息, 请参阅第 53 页的第 10 章, 『角色定义』。

管理 FastForward 应用程序

FastForward 管理应用程序提供了三种常规功能:

- 管理已发布的报告
- 管理 FastForward 应用程序的基本外观
- 查看和清除应用程序日志

也可以在登录到“管理任务”方式后通过单击应用程序工具栏上的“帮助”按钮来获得有关管理 FastForward 应用程序的信息。

报告访问类别和安全性

FastForward 应用程序为用户提供了三种报告访问类别: “已发布”、“专用”和“组”。

已发布的报告

已发布的报告是从管理屏幕中管理的, 如下所述。如果未采用其他方式对已发布的报告进行管理, 则所有可以访问 FastForward 应用程序的用户都可以查看这些报告。但是, 管理员可以通过对报告文件夹设置安全性来控制对已发布的报告进行的访问。如果一个用户可以访问给定的文件夹, 则他或她就可以访问该文件夹中的任何报告。管理员通过使文件夹与一个或多个 DB2 Alphablox 组相关联来对文件夹指定访问权。

专用报告和组报告

专用访问权报告和组访问权报告是由先前发布的报告的用户创建的。专用访问权报告仅可供创建该报告的用户使用。组访问权报告可供隶属于 DB2 Alphablox 用户组的用户使用, 并且, 任何隶属于给定组的用户都可以创建可供该组访问的报告。

布局和控制

所有基于 Alphablox FastForward 的应用程序都具有类似的布局结构，这包括左边的导航菜单以及右上方的应用程序工具栏。在位于应用程序工具栏下方的报告窗口中查看可用的报告。

导航菜单

在导航菜单上，提供了下列选项：

按钮	描述
创建文件夹	创建一个名为“新文件夹”的新文件夹。将装入“编辑文件夹”屏幕以帮助您更改该文件夹。
创建报告	装入“编辑报告”屏幕，以便可以编辑新报告。必须先成功地保存报告，然后它才会显示在左边的报告树中。
删除报告	弹出一个对话框，请求您确认删除报告。
复制	复制报告，并且在报告名前面加上“复件”。接着，可以将该文件重命名。

管理报告

当处于“管理任务”方式时，Alphablox FastForward 使用应用程序的用户界面提供了基本的报告管理。

创建报告

要创建报告，可以单击工具栏中的“创建报告”按钮。将装入“编辑报告”屏幕，并且您可以配置下列设置：

字段	描述
名称	树中显示的报告名。
描述	当用户将鼠标移到树中的报告上，以及当装入报告时，将显示这个简短描述。
模板	模板下拉列表提供可供应用程序使用的每个模板的名称。报告模板存储在应用程序的 <code>templates</code> 目录中。根据特定报告模板的不同，将需要选择其他参数。
保存	保存报告。如果无法验证报告参数，则可能会显示错误。
预览	在新的浏览器窗口中打开报告。如果无法验证参数，则可能会显示错误。

修改报告

要编辑报告，请左键单击树中的现有报告。这将装入“编辑报告”屏幕。

删除报告

要删除报告，请左键单击要删除的报告，然后单击工具栏中的“删除报告”按钮。

移动报告

要移动报告:

1. 将鼠标指针定位在要移动的报告旁边的报告图标上，然后按下鼠标左键不放。
2. 在按下鼠标按键不放的情况下，将指针移至树中的期望目标位置，然后放开鼠标按键。

管理文件夹

文件夹管理是一项基本操作，可以使用应用程序的用户界面来完成该操作。

创建文件夹

要创建文件夹，可以单击工具栏中的“创建文件夹”按钮。将装入“编辑文件夹”屏幕，该屏幕包含下列可编辑的字段:

字段	描述
名称	树中显示的报告名。
描述	当用户将鼠标移到树中的文件夹上时显示的值。
访问组	可用的 DB2 Alphablox 用户组的列表。可以选择多个用户组。如果未选择任何组，则所有用户都可以访问该报告（并且将显示警告）。
保存	保存文件夹。

修改文件夹

要编辑文件夹，请左键单击树中的现有报告。这将装入“编辑文件夹”屏幕，可以在该屏幕中进行更改。

删除文件夹

要删除报告，请左键单击要删除的报告，然后单击工具栏中的“删除报告”按钮。

移动文件夹

要移动文件夹:

1. 将鼠标指针定位在要移动的文件夹旁边的文件夹图标上，然后按下鼠标左键不放。
2. 在按下鼠标按键不放的情况下，将文件夹拖动到树中的期望目标位置，然后放开鼠标按键。

管理应用程序属性

要管理应用程序属性，请单击页面右上角的“全局设置”按钮。可以修改下列设置:

字段	描述
标题	应用程序的标题。此标题显示在用户应用程序顶部以及管理页面顶部。
徽标	显示作为用户应用程序中的徽标显示的文件的名称。
上传徽标图像	从任何本地文件（GIF 或 JPEG 格式）上载新的徽标图像。注意，在您单击“保存”之前，不会实际地发生上载。

主题	供您选择已安装的主题的下拉列表。注意，样本应用程序是仅为 <code>fastforward</code> 主题设计的。
页脚	更改显示在用户应用程序页面底部的文本。
管理电子邮件	设置用于接收应用程序反馈的电子邮件地址。如果未设置此地址，则会导致用户应用程序页面中的反馈按钮处于禁用状态。
保存	此选项保存上述编辑字段中的更改。

使用应用程序日志

FastForward 应用程序框架使用它自己的日志文件来报告应用程序级别的信息和异常。可以通过单击“查看日志文件”按钮来查看和清除此日志文件。

附录. OLAP 术语和概念

本节阐述对于理解多维分析 (MDA) 和 OLAP (联机分析处理) 来说十分关键的术语和概念。以二维分析中使用的术语开始, 我们定义了它们在 MDA 中的延伸等效术语。然后, 我们阐述 OLAP 数据库中使用的相关术语。

要了解更多的 OLAP 定义, 另请参阅 DB2 Alphablox 词汇表和 OLAP 参考词汇表。

- 『二维分析』
- 『多维分析』
- 第 139 页的『OLAP 数据库术语』

二维分析

使用诸如电子表格和报告之类的工具来执行二维数据分析的用户熟悉下列术语, 这些术语后面的表对它们作了阐述:

- **行**包含一组相关的数据。下表包含两个数据行。
- **行标号**显示在数据值的左边。行标号是深蓝色的。
- **列**也包含一组相关的数据。下表包含 4 个数据列。
- **列标号**显示在数据值上方。列标号是栗色的。
- **数据点** (也称为**单元格**) 是行与列的交点。数据点具有灰色背景。
- **数据值**是位于特定数据点处的元素。数字 10 至 30 是数据值。

二维销售表

	Qtr 1 Units	Qtr 2 Units	Qtr 3 Units	Qtr 4 Units
Diet Cola	10	15	20	25
Cola	12	18	24	30

多维分析

在**多维分析**中, 这些术语和概念更为复杂。MDA 不仅仅涉及行、列以及它们的交点。尽管用于显示多维数据的介质通常是二维网格, 但是 MDA 的能力涉及维、层次结构、成员、标题、值和实例以及行、列和数据点。

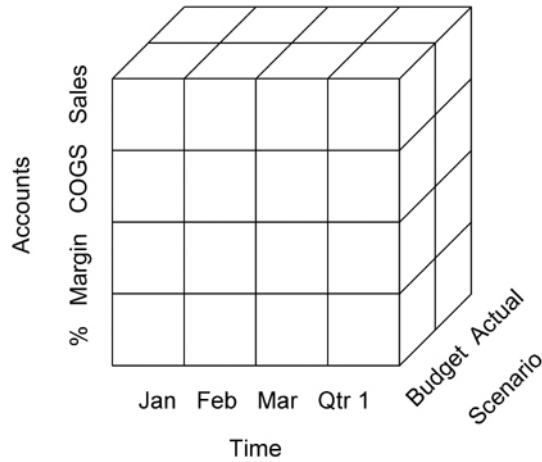
下列定义适用于多维数据, 这些定义后面的多维矩阵对这些定义作了阐述:

- **维**是数据立方体的结构属性, 并且由相关的分层成员组成。例如, Time 维可以包含下列成员: years、quarters、months 和 weeks。Geography 维可以具有下列成员: regions、countries 和 cities。
- **维成员**是维中的元素。例如, Quarter 和 Month 是 Time 维的成员。
- **维层次结构**将维成员组织成父代/子代关系。例如, Month 是 Quarter 的子代 (Month 属于 Quarter); Quarter 是 Year 的子代, 依此类推。
- **维标题**就是维的名称, 如 Time 或 Geography。
- **维成员标题**就是维成员的名称, 如 Month 或 Region。
- **维成员值**是维成员的实例; 1998 是维成员 Year 的值。

- **数据点**是多个维的交点。以下多维销售矩阵中的数据点具有灰色的背景。
- **数据值**位于数据点处。例如，在多维销售矩阵中，灰色区域中的每个数字都是数据值。

数据立方体

下图显示了已格式化为具有三个维（每个维具有若干个成员）的立方体的数据：



维	成员
Accounts	Sales、Cost of Good Sold (COGS)、Margin 和 Margin Percent
Time	Quarter 1, 包含: January、February 和 March
Scenario	Budget 和 Actual

多维销售矩阵

下图显示了销售数据的多维分层视图，在此视图中，Time 维的两个成员（Year 和 Quarter）显示在列轴上，Inventory 维的两个成员（Category 和 Product）显示在行轴上。

		列轴						
		Qtr 1						
		Jan		Feb		Mar		
		Budget	Actual	Budget	Actual	Budget	Actual	
行轴	Accounts	Sales	300	350	300	325	325	325
	COGS		175	185	175	175	185	190
	Margin		125	165	125	150	140	135
	Percent		42	47	42	46	43	42

OLAP 数据库术语

下列术语帮助您标识、组织和检索 OLAP 数据库中的数据。例如，当用户从 Qtr1 向下钻取时，OLAP 操作是检索 Qtr1 的“子代”（Jan、Feb 和 Mar）。

样本维层次结构	术语和定义
<ul style="list-style-type: none">[-] Year<ul style="list-style-type: none">[-] Qtr1<ul style="list-style-type: none">[+] Jan[+] Feb[+] Mar[-] Qtr2<ul style="list-style-type: none">[+] Apr[+] May[+] Jun[-] Qtr3<ul style="list-style-type: none">[+] Jul[+] Aug[+] Sep[-] Qtr4<ul style="list-style-type: none">[+] Oct[+] Nov[+] Dec	<p>根: 层次结构中的最顶部位置。在示例中，根是 Year。</p> <p>后代: 比根至少低一代的成员。示例中的所有其他维都是根的后代。Jan 也是 Qtr1 和 Year 的后代。</p> <p>子代: 刚好比另一个成员低一代的成员。Jan 是 Qtr1 的子代。</p> <p>父代: 刚好比另一个成员高一代的成员。Qtr1 是 Jan 的父代。</p> <p>同代: 与另一个成员位于同一代的成员。Jan、Feb 和 Mar 是同代。</p> <p>祖代: 比另一个成员高一代或多代的成员。Qtr1 和 Year 都是 Jan 的祖代。</p>

注: 用作这些术语中的其中一个术语的前缀的大写字母“**I**”（IPARENT 和 ICHILD 等等）指示“包括”。例如，在进行父代（Parent）向上钻取时，结果将仅显示父代的数据。在进行 IPARENT 向上钻取时，结果既包括父代的数据也包括子代的数据。

词汇表

本词汇表中提供的术语与 DB2 Alphablox 相关。此外，请参阅第 137 页的『OLAP 术语和概念』中的讨论。在因特网上，诸如 A Web of Online Dictionaries 和 WhatIs 之类的站点提供了更详尽的词汇表。

2 维 (2D)： 指的是使用行和列来以二维方式显示信息。示例包括简单列表、报告和二维电子表格。

3 维 (3D)： 指的是使用页、行和列维来以多维方式显示信息。GridBlox 提供这样的显示。

A

ACL： 访问控制表。被允许访问特定服务的用户的列表。

DB2 Alphablox 应用程序 (DB2 Alphablox application)： 下列各项的组合：(1) 包含 Blox 的 JSP 文件，(2) 应用程序所使用的数据的访问信息，以及 (3) 用于定义页面如何向用户显示的属性集合。应用程序位于 Web 服务器上，并显示在浏览器窗口中以响应用户请求。

祖代 (ancestor)： 比另一个维成员高一代或多代的维成员。请参阅“OLAP 术语和概念”中的“祖代”以获取示例。

API： 应用程序接口，即程序员在编写源代码时使用的接口。API 由库调用的名称以及它们所接收的自变量的数目和类型组成。在 DB2 Alphablox 中，API 由可供 Java 和 JavaScript 处理的 Blox 方法和属性组成。

applet： 请参阅 Java applet。

应用程序数据库 (application database)： 位于服务器上并且是为了支持特定应用程序而存在的数据库。可以从生产数据库中抽取应用程序数据库以使应用程序能够快速地访问相关数据。DB2 Alphablox 使用多维数据库和关系数据库来作为应用程序数据的来源。

应用程序页面 (Application page)： 作为 DB2 Alphablox 应用程序一部分的单个 HTML 页面。

认证 (authentication)： 验证个人或进程的标识的过程。通常，认证要求用户输入有效的标识 / 密码对。

轴 (axis)： 用来组织和显示数据的坐标。轴可以显示一个或多个维。对于多维数据，轴还被定义为立方体的其中一条边。

- 折线图或条形图沿水平轴 (X) 和垂直轴 (Y) 显示数据。

- 网格使用行轴、列轴和页面轴来显示多维数据。另外，DB2 Alphablox 可以显示一条名为“其他”的轴，该轴显示当前未使用的维。将一个维放在“其他”轴上时，将从当前视图中除去该维，但是保持该维可供使用。

B

Blox： 可复用的软件组件，这些软件组件有助于您为基于内部网 / 因特网的企业级分析解决方案快速组装应用程序。要了解有关如何将 Blox 组装成完整的应用程序的信息，请参阅《开发者指南》。要了解有关每个 Blox 的详细信息，请参阅 *Developer's Reference*。

C

级联样式表 (Cascading Style Sheets, CSS)： Web 页面开发者使用的用于定义 HTML 页面元素外观的技术。此技术与字处理程序使用的样式表类似。要了解更多信息，请参阅 Hakon Wium Lie 和 Bert Bos 撰写的 *Cascading Style Sheets: Designing for the Web* (Addison Wesley Longman 1997)，或者访问 <http://www.w3.org/Style/>。

目录 (catalog)： 数据源结构中的组织级别。在 DB2 OLAP Server 或 Essbase 中，这称为“应用程序”。通常，数据立方体分组为模式，而模式又分组为目录。

CGI： 公共网关接口。用于在 Web (HTTP) 服务器上执行程序的编程接口。CGI 提供了用于将数据从服务器传递至服务器的网关程序（此程序执行处理）以及用于将结果从网关程序返回到 Web 服务器再返回到请求客户机的结构。与 ISAPI 或 NSAPI 对照。

子代 (child)： 刚好比另一个维成员低一代的维成员。请参阅“OLAP 术语和概念”中的“子代”以获取示例。

折叠 (collapse)： 将详细信息合并到单个总结值中。与展开 (expand) 对照。

列轴 (column axis)： 一个用于查看多维数据的方向。列轴是三个可查看的轴的其中之一（另外两个轴是页面轴和行轴）。列显示维成员的数据值。通过将维放在列轴上，用户就能够查看该维的成员的数据值。例如，将 Time 维放在列轴上将使用户能够查看成员标识（如 Year 和 Quarter）以及相关的数据值（如销售量）。

cookie： 一种对象，它包含客户机上存储的关于用户会话的信息（如会话标识）。

CSS 类 (CSS class)：一种用来定义对元素的正常样式的扩展或覆盖的 CSS 技术。

CSV：逗号分隔的值。一种文件格式，它包含由换行符定界的行，并且各行中的数据值由逗号分隔。

D

DAA：请参阅动态应用程序组装 (Dynamic Application Assembly)。

数据项 (data item)：在多维分析中，这指的是位于两个或更多个维的交点的值。

数据库连接器 (database connector)：一个编程接口，它提供了一种用于通过网络访问数据库的公共语言。示例包括用于 Java 实现的 JDBC (Java 数据库连接) 以及用于 Microsoft 和 Apple Macintosh 应用程序的 ODBC (开放式数据库连接标准)。

数据集市 (data mart)：从运作过程以及其他来源收集的数据的存储库，用于满足信息工作者的分析和表示需求。此数据可以派生自企业级数据库或数据仓库，并且通常是以特定信息工作者小组所熟悉的形式表示的。数据集市强调易于访问以及适用于特定用途 (如销售分析)。

数据仓库 (data warehouse)：由企业的业务系统收集的数据的中央存储库。数据仓库通常位于企业服务器上。对来自各种联机事务处理 (OLTP) 应用程序的数据进行抽取和组织，以供分析应用程序和用户查询使用。

不推荐 (deprecated)：用于参数、类和方法，以表示它们将来可能不再存在。随着类的不断发展及其 API 发生变化，就会产生不推荐使用的需求：参数和方法将被重命名以保持一致性，并且将增加新的参数和方法。旧的 API 将保留，直到组装人员转而采用新的 API，但一旦不推荐使用某个 API，组装人员就不应该使用该 API。每当遇到不推荐的项时，DB2 Alphablox 就会将消息写至客户机控制台。

后代 (descendant)：比根 (维层次结构中的最顶部位置) 至少低一代的维成员。请参阅“OLAP 术语和概念”中的“后代”以获取示例。

维 (dimension)：立方体的结构属性，它可以帮助定义立方体的层次结构。维列出成员，用户将所有那些成员视为具有类似类型的数据。例如，Time 维可以包括年份成员、季度成员、月份成员和星期成员。Market 维可以包括国家或地区成员、区域成员、州成员以及城市成员。维可以帮助组织多维数据源中的数据，从而能够进行多维分析。

DLL：动态链接库。一种应用程序文件，它作为应用程序的进程和地址空间的一部分运行。DLL 文件是在应用程序

启动时被装入的，并且，需要多久就可以保留多久，从而消除了 CGI 应用程序所需的重复的应用程序检索时间。

向下钻取 (drill down)：在维成员的层次结构中向下移动。例如，用户可以查看 North America 的财务数据，然后向下钻取通过 US、Western states、California、Northern California 并且最后到达 Silicon Valley。

向上钻取 (drill up)：在维成员的层次结构中向上移动。例如，用户可以查看 Silicon Valley 的财务数据，然后向上钻取通过 Northern California、California、Western States 并且最后到达 US。

DSS：决策支持系统。一种信息和规划系统，它支持使用各种信息源进行的特别查询、分析和预测。

动态应用程序组装 (Dynamic Application Assembly)：从一组行为定义良好的组件快速组装应用程序的方法。

E

展开 (expand)：显示支持总结值的详细信息。与折叠 (collapse) 对照。

外部网 (extranet)：可供外部 (因特网) 世界访问但只能通过特权机制 (如注册的密码) 进行访问的企业 Web 站点。外部网有助于与伙伴、客户和其他具有预定义关系的实体进行交互。

F

胖客户机 (fat client)：使用必须安装在客户机上的客户机常驻应用软件来执行重要处理的客户机。与瘦客户机 (thin client) 对照。

过滤器 (filter)：一种限制查询所返回的数据的方法。例如，过滤器可以指定查询仅返回具有特定 Year 值的数据。

框架集 (frameset)：浏览器窗口中的两个或更多个可独立控制的部分 (框架) 的集合，每个部分都显示一个单独的 HTML 文件。显示在一个框架中的文件可以包含指向将要显示在另一个 (或同一个) 框架中的文件的链接。通常，一个框架包含用于进行导航和选择的用户工具，而其他框架显示用户操作的结果。

G

GIF 图像 (GIF image)：图形交换格式。一种受 HTML 支持的图形格式。

网格 (grid)：信息的多维显示，它使用行轴、列轴和页面轴。网格使用户能够对数据显示向下钻取 (和向上钻取)、切片和旋转。GridBlox 在网格中显示数据。

组 (group): 多个行项或列项的重复标题的压缩。另外, 组中数据项的总计值是自动生成的。程序员可以通过 SQL GROUP BY 子句来在关系数据库中定义组。程序员通过定义数据库层次结构来在多维数据库中定义组。

分组 (grouping): 这指的是在显示数字数据时将该数据格式化为通常由逗号或点标记的数字组 (如千)。

H

散列表 (hashtable): 一个 Java 数据结构, 它将键映射至值。下列示例取自 JavaSoft Java API。第一个示例创建一个数字散列表并使用字母表中的字母作为键 (因此, A 指向 1, B 指向 2, 等等)。第二个示例使用值的键 (B) 来检索特定的值 (2)。

```
Hashtable letters = new Hashtable();
numbers.put("A", new Integer(1));
numbers.put("B", new Integer(2));
numbers.put("C", new Integer(3));

Integer n = (Integer)numbers.get("B");
if (n != null) {
    System.out.println("B = " + n);
}
```

层次结构 (hierarchy): 将维的成员组织成父代/子代关系, 通常, 父代成员表示它的子代的合并。例如, Time 维中的层次结构可能是年份、季度和月份。每个月份都是特定季度的子代, 每个季度都是特定年份的子代。

HSB: 一种颜色模型, 它使用色调、饱和度和亮度来描述颜色。

HTML: 超文本标记语言。一种文本标记语言, 它指定文本在浏览器窗口中的显示方式。HTML 将屏幕文本连接至可能位于因特网上其他位置中的其他文本、图像和其他对象, 如 Java applet。

HTML 页面 (HTML page): 一个文本文档, 它可以包含标准的 HTML 标记、文本、图形、多媒体对象以及 Blox。HTML 页面是在 Web 浏览器中显示的。

HTML 标记 (HTML tag): 嵌入在 HTML 文档中的语言代码, 此代码告诉浏览器如何显示文档内容。HTML 标记包括名称和可选的属性。

HTTP: 超文本传输协议。一组用来通过万维网或通过内部网交换文件 (文本、图像、声音、视频和多媒体) 的规则。

I

实例化 (instantiate): 创建特定类的对象。例如, 当用户调用 DB2 Alphablox 应用程序时, DB2 Alphablox 就将该应用程序实例化 (创建该应用程序的实例)。

因特网 (Internet): 计算机的全球网络, 这些计算机使用一组公共的联网协议和软件协议。因特网允许用户跨多种平台共享发布至 Web 页面的数字信息。另请参阅外部网 (extranet) 和内部网 (intranet)。

内部网 (intranet): 供内部使用的企业网络, 此网络提供的服务与因特网提供的那些服务类似。另请参阅外部网 (extranet)。

ISAPI: 因特网服务器应用程序接口。一组特定于 Microsoft Internet Information Server 的 Microsoft Windows 程序调用, 此接口使您能够编写运行速度比 CGI (公共网关接口) 应用程序更快的 Web 服务器应用程序。

J

Java applet: 使用由 Sun Microsystems 开发的 Java 语言编写的程序。Java applet 嵌入在 Web 页面中, 并且在浏览器显示该页面时执行。

JavaScript: 一种基于对象的脚本语言, 用于 Web 应用程序的功能扩展及用户界面的增强。和 Java applet 一样, JavaScript 可以嵌入在 Web 页面中。

JDBC: Java 数据库连接。用于 Java 中的关系数据库访问的主要 API。JDBC 基于结构化查询语言 (SQL) 调用级接口。它允许程序使用 SQL 请求来访问数据库, 而不必使用那些数据库的专用接口。每个数据库各自的模块或驱动程序负责处理该 SQL 请求以及将该请求转换为各个数据库系统所能够理解的请求。

JPEG 图像 (JPEG image): 联合图像专家组。一种受 HTML 支持的图形格式。为了符合三字符文件扩展名这一 DOS 需求, JPEG 文件通常具有 .eps 或 .jpg 扩展名。

JRE: JavaSoft Java 运行时环境, 它由 Java 虚拟机、Java 核心类和支持文件组成。它是 J2SE Software Development Kit (JDK) 的运行时组件。JRE 是组成标准 Java 平台的一组最小的可执行文件和其他文件。

JVM: Java 虚拟机。负责解释和执行 Java 代码的实体。

L

旧数据库 (legacy database): 一种数据库, 它通常位于大型机上, 并且通常使用专利技术。它通常是企业重要的信息保险库。

M

元数据 (metadata): 用来描述其他数据的数据。数据库模式和数据字典包含元数据。

MDA: 多维分析。通过对数据的查询、向下钻取、切片、旋转以及定义数据的计算方法来浏览并显示数据库子集，以完成数据分析。

中间件 (middleware): 负责跨异构网络管理应用程序之间的交互的软件。例如, JDBC 驱动程序就是中间件, 它们允许程序访问数据库, 而不必使用那些数据库的专用接口。

MIME: 多用途因特网邮件扩展。允许计算机系统使用因特网邮件标准来以音频、图形、视频和文本格式交换多媒体信息。

多维数据库 (multidimensional database): 请参阅 OLAP 数据库。

多 SQL (multi-SQL): 多个链接到一起以形成单个复杂查询的 SQL 语句。相关的术语包括相关查询、子查询和嵌套查询。

多层访问 (multi-tier access): 请参阅 N 层访问 (N-tier access)。

N

N 层访问 (N-tier access): 跨多个计算层访问数据的过程。简单的客户机 / 服务器体系结构使用两层的访问, 即客户机请求服务器上存储的数据, 服务器将结果传递给该客户机。内部网体系结构通常涉及三层的访问, 即客户机浏览器通过 Web 服务器请求数据, 而后者又将该请求传递至数据库服务器。亦称为多层访问 (*multi-tier access*)。

本机 SQL (native SQL): 给定的数据库附带提供的关系数据库编程语言。本机 SQL 通常包括针对该数据库进行了优化的功能。请参阅 SQL。与 ODBC SQL 对照。

NSAPI: Netscape 应用程序接口。此接口使您能够编写运行速度比 CGI (公共网关接口) 应用程序更快的 Web 服务器应用程序。

O

ODBC: 开放式数据库连接标准。一个数据库编程接口, 此接口使得有可能从任何应用程序中访问任何数据, 而不考虑处理该数据的数据库管理系统 (DBMS)。ODBC 在应用程序与 DBMS 之间插入了一个中间层 (称为数据库驱动程序)。这个中间层将应用程序的数据查询转换为 DBMS 所理解的命令。

ODBC SQL: 用于访问网络上的关系数据库的开放的公共编程语言。为了提供接近通用的数据库访问, ODBC SQL 可能未包括数据库的本机 SQL 中的某些功能。请参阅 SQL。与本机 SQL (native SQL) 对照。

OLAP 数据库 (OLAP database): 联机分析处理数据库。一种数据库, 它使用特殊的建立索引技术来提供对总结数据以及该数据的多维视图的快速查询处理。DB2 OLAP Server 就是 OLAP 数据库的一个示例。

P

页面轴 (page axis): 一个用于查看多维数据的方向。页面通过多维数据库来定义切片。通过将维放在页面轴上, 用户就能够从该维中选择成员并且仅查看该成员的数据。例如, 通过将 Market 维放在页面轴上, 用户就能够从该维中选择 New York 并且仅查看 New York 的数据。

父代 (parent): 刚好比另一个维成员高一代的维成员。请参阅“OLAP 术语和概念”中的“父代”以获取示例。

持久性 (persistence): 编程语言的一项属性, 即已创建的对象和变量在程序的各次运行之间将一直存在并保留它们的值。

旋转 (pivot): 将维从一个轴移至另一个轴。组织人员可以通过 Essbase 报表规范或通过 JavaScript 来定义旋转。用户可以通过用户界面来执行旋转。

代理服务器 (proxy server): 位于客户机与服务器之间的服务器。代理服务器拦截所有前往服务器的请求以确定它是否应该自行满足那些请求。如果不是的话, 它将请求转发至该服务器。在因特网 / 内部网环境中, 代理服务器可以通过拦截、保存和提供频繁请求的页面来显著提高 Web 服务器的性能。

R

报表规范 (report specification): 在 DB2 OLAP Server 或 Hyperion Essbase 中, 这指的是定义结果集以供应用程序使用。报表规范包含数据抽取命令 (这些命令定义要从数据库中检索的数据) 和格式化命令 (这些命令定义要在网格中的页面轴、列轴和行轴上显示的维)。

存储库 (repository): 一个信息数据库, 它标识对象并使它们可以被复用。DB2 Alphablox 存储库存储用户对象和应用程序对象、访问控制信息以及配置参数。

RGB: 一种颜色模型, 它使用红色、绿色和蓝色来描述颜色, 这三种颜色的光可以混合以生成任何其他颜色。

根 (root): 维层次结构中的最顶部位置。请参阅“OLAP 术语和概念”中的“根”以获取示例。

行轴 (row axis): 一个用于查看多维数据的方向。行轴是三个可查看的轴的其中之一 (另外两个轴是页面轴和列轴)。行显示维成员的数据值。通过将维放在行轴上, 用户就能够查看该维的成员的数据值。例如, 将 Product 维放

在行轴上将使用户能够查看成员标识（如 Product ID 或它的别名）以及相关的数据值（如销售量）。

S

模式 (schema)：数据源结构中的组织级别。在 DB2 OLAP Server 或 Essbase 中，这称为“数据库”。通常，数据立方体分组为模式，而模式又分组为目录。

同代 (sibling)：与另一个维成员位于同一代的维成员。请参阅“OLAP 术语和概念”中的“同代”以获取示例。

切片 (slice)：通过将一个或多个维放在页面轴上来过滤一组数据。例如，销售数据的立方体可以将产品放在行轴上，将月份放在列轴上并将年份放在页面轴上。选择年份将对数据进行切片，从而仅显示所选年份的数据。

SQL：结构化查询语言，它提供了用于访问和检索关系数据库管理系统 (RDBMS) 中的数据的用户接口。另请参阅 ODBC SQL 和本机 SQL (native SQL)。

T

TCP/IP：传输控制协议 / 因特网协议 (Transmission Control Protocol/Internet Protocol)。在因特网上用来连接主机的基本通信语言 (协议)。也指 Transport Control Protocol/Internet Protocol。

层 (tier)：计算层次结构中的一个级别。因特网 / 内部网环境通常实现三层的层次结构：第 1 层是运行在客户机上的浏览器；第 2 层是运行在单独机器上的 Web (HTTP) 服务器；第 3 层是运行在另一台机器上的数据库服务器。

主题 (theme)：可以应用于 HTML 页面的设计元素（如字体和图像）的集合。对若干个页面使用同一个主题是一种简单易行的确保外观一致的方法。

瘦客户机 (thin client)：执行重要的应用程序处理但没有常驻应用软件（而是在需要该软件时才下载它）的客户机。支持 Java 的 Web 浏览器就是瘦客户机。浏览器可以显示 Web 页面和执行嵌入的 applet。与胖客户机 (fat client) 对照。

线程 (thread)：可以独立于其他程序部件执行的程序部件。支持多线程的操作系统允许开发者设计具有可以并发执行的线程部件的程序。

线程安全 (Threadsafe)：可以具有多个同时进行而又不会相互干扰的交错或嵌套调用的代码，或者是由于受到某种形式的互斥的保护而不会并发执行多次的代码。

元组 (tuple)：在 DB2 Alphablox 中，这指的是用来定义立方体子集的一组成员（这些成员来自若干个维中的每一个维）。在下面的网格中，列轴上的元组包括 South、East

和 West；行轴上的元组包括 VCR QTR1、TV QTR1 和 TV QTR2。

		South	East	West
VCR	QTR 1	10	20	30
TV	QTR 1	20	30	40
TV	QTR 2	30	40	50

U

URL：统一资源定位符。因特网上用于指定对象（如文件或新闻组）的草稿标准。HTML 文档使用 URL 地址来指定超链接的目标。

V

向量 (Vector)：称为元素的对象的可扩展数组。向量中的元素数目可以按照程序逻辑的指示动态地增加或减少。元素数目是由向量的大小定义的（按向量的容量表示的存储容量）。

W

Web 服务器 (web server)：一个服务器进程，它发出 HTML 页面以响应远程浏览器的请求。此术语亦指运行该进程的机器。

声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

*IBM World Trade Asia Corporation, Licensing,
2-31 Roppongi 3-chome, Minato-ku,
Tokyo 106-0032, Japan*

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

*IBM Corporation,
J46A/G4, 555 Bailey Avenue,
San Jose, CA 95141-1003
U.S.A.*

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本文档中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，与实际商业企业所用的名称和地址的任何雷同纯属巧合。

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。用户如果是为了按照 IBM 应用程序编程接口开发、使用、经销或分发应用程序，则可以任何形式复制、修改和分发这些样本程序，而无须向 IBM 付费。

商标

下列各项是 International Business Machines Corporation 在美国和 / 或其他国家或地区的商标或注册商标:

DB2

IBM

DB2 OLAP Server

WebSphere

DB2 Universal Database™

Alphablox 和 Blox 是 Alphablox Corporation 的商标或注册商标。

Intel® 和 Pentium® 是 Intel Corporation 在美国和 / 或其他国家或地区的商标。

Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

Java、JavaBeans、JavaScript 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

Linux 是 Linus Torvalds 在美国和 / 或其他国家或地区的商标。
其他公司、产品或服务名称可能是其他公司的商标或服务标记。

索引

[A]

安全模型

- Alphablox 登录模块 56
- Alphablox 8.4.1 55
- JNDI 域 56

安全性

- 定制实现 91
- 对 DB2 Alphablox 启用 71
- 基于 Web 的, 设置 62
- 基于 Web 的, 与 DB2 Alphablox 57
- 角色, 使用 53
- 目录内容, 访问 63
- 认证方式 55
- 使用 Microsoft IIS 的 Windows 认证 58
- 系统属性, 指定 70
- 用户帐户, 自动生成 62
- 域和应用程序访问 57
- admin 权限与用户权限 57
- Microsoft Analysis Services 38
- Sun iPlanet 选项 58
- 请参阅 角色

[B]

- 报告, FastForward 133
- 别名, Essbase 127

[C]

- 存储库管理器 9, 105
- 存储库转换实用程序 105
 - 操作描述 110
 - 启动 106
 - 使用 106
 - 数据库到文件 108
 - 文件到数据库 107
 - 语法, 命令行 110
 - 自变量 111
- 存储库, DB2 Alphablox
 - 存储库转换实用程序 106
 - 概述 105
 - 关系的优点 105
 - 管理器 9
 - 类型, 检查 106
 - 配置 105
 - 配置实例以使用现有的 109

[D]

- 单点登录 91
- 定制属性
 - 缺省值 73
 - 已定义 73
 - 应用程序属性, 定义 74
 - 应用程序属性, 更改 75
 - 应用程序属性, 删除 75
 - 用户属性, 定义 73
 - 用户属性, 更改 74
 - 用户属性, 删除 74
- 端口
 - 指定 71
- 对命令进行高速缓存
 - DB2 OLAP Server 127
 - Essbase 127

[F]

- 访问控制表,
 - 请参阅 角色
- 服务管理器 8
- 服务器空闲持续时间,
 - 请参阅 Alphablox Analytics
- 服务, DB2 Alphablox 作为 40

[G]

- 概述
 - DB2 Alphablox 1
- 个性化引擎,
 - 请参阅 可扩展用户管理器
- 关系立方体
 - 请参阅 Cube Server 管理员指南
- 关系数据库
 - 存储库, 用作 105
 - JDBC 跟踪 41
 - JDBC 驱动程序, 更新 42
 - Sybase JConnect, 设置环境 41
- 管理器, 存储库 105
- 管理注释集合 75

[H]

- 会话管理器 8

[J]

- 基于 LDAP 的用户管理器, 配置 87

- 级联样式表 (CSS)
 - 设置与 DHTML 客户机 25
- 集群
 - 集群管理器 10
 - 命令列表, 集群 120
 - 启动 119
- 集群管理器 10
- 集群, DB2 Alphablox
 - 概述 119
- 计算
 - 扩展 DB2 Alphablox 65
- 角色
 - 成员资格, 更改 53
 - 定义 53
 - 删除 54
 - 组和用户, 更改 47, 54

[K]

- 可扩展性
 - Blox UI 模型 66
 - DHTML 客户机 66
- 可扩展用户管理器 86
 - 编程接口 90
 - 扩展 DB2 Alphablox 65
- 可信用户, 已定义 58
- 空闲持续时间, 服务器 70
- 控制台
 - 访问 16, 123
 - 管理器 10
 - 集群命令 120
 - 命令列表, DB2 Alphablox 124
 - 命令文件, 使用 69
 - 命令语法 123
 - 缩写 124
 - 特定于 DB2 OLAP Server 的命令 127
 - 特定于 Essbase 的命令 127
 - 文本文件, 运行 130
 - 消息级别 129
 - DB2 Alphablox, 停止, 使用 21
 - HTML 123
 - Telnet 123

[L]

- 类路径 66
- 类路径设置
 - JDBC 驱动程序, 修改 42
- 类文件 18
- 连接池
 - 关系数据源 116
 - 关系, DB2 Alphablox 使用 116
 - BEA WebLogic 117
 - DB2 Alphablox 存储库 117
 - DB2 Alphablox 数据源 116

- 连接池 (续)
 - DB2 OLAP Server 113
 - Hyperion Essbase 113
 - Microsoft Analysis Services 113
- 轮廓高速缓存,
 - 请参阅 对命令进行高速缓存

[M]

- 目录浏览, 禁用 63

[Q]

- 启动属性
 - 服务器空闲持续时间 70
 - 命令文件名 69
 - 配置 69
 - 日志文件名 69
 - 实例名 69
 - 消息级别, 缺省值 70
 - 消息级别, DB2 Alphablox 控制台, 缺省值 70
 - DB2 Alphablox 日志, 启用 70
- 请求管理器 8
- 驱动程序, JDBC, 更新 42
- 全局设置, FastForward 135

[R]

- 认证
 - 安全方式 55
 - 安全域 57
 - 对 DB2 Alphablox 启用 71
 - DB2 Alphablox 与 Web 服务器 57
- 日志
 - 应用程序日志, FastForward 136
- 日志文件
 - 重命名 80
 - 管理 80
 - 启动属性, 定义 69
 - 启用 DB2 Alphablox 以将消息写入 70
 - 消息级别, 指定 70, 129
- 日志文件, DB2 Alphablox 79
 - 日志文件滚动时间间隔设置 79
 - 请参阅 日志文件

[S]

- 实例名
 - 存储库转换, 指定 107
 - 为 DB2 Alphablox 指定 69
- 实用程序, 存储库转换 105, 106
- 数据管理器 9
- 数据库
 - 数据适配器 3
 - 数据源, 定义 37

数据库 (续)
 Microsoft Analysis Services 安全性 38
 OLAP 术语 139
 Sybase JConnect 关系驱动程序设置 41
数据适配器 3
数据源
 定义 37
 定义, 更改 38
 定义, 删除 38
 Microsoft Analysis Services 安全性 38
属性
 定制,
 请参阅 定制属性
 启动, 配置 69
 系统 70
 用户 45
缩写, 控制台命令 124

[T]

弹出窗口拦截软件 25
体系结构
 DB2 Alphablox 7
停止 DB2 Alphablox 21
头链接
 设置 29

[W]

文本文件, 通过控制台运行 130
我的概要文件
 访问 45
 链接 18

[X]

系统属性
 配置 70
 Web 服务器 URL 前缀, 指定 70
系统需求
 FastForward 应用程序 132

[Y]

已授权的客户机列表, 指定 71
应用程序
 重命名 31
 定义 28
 定义, 更改 31
 定义, 删除 32
 定义, WebLogic 集群 32
 对 WebSphere 定义 29
 概述 6
 管理器 9
 类型 11

应用程序 (续)
 目录结构 27
 认证 57
 在 Apache HTTP Server 上注册 33
 在 Microsoft IIS 上注册 34
 在 Sun iPlanet 上注册 34
应用程序管理器 9
应用程序名 28
应用程序迁移实用程序
 访问 17
用户
 创建 45
 定制属性, 更改 74
 概要文件, 编辑 45
 更改 45
 角色, 更改成员资格 54
 删除 46
 属性, 更改 45
 自动生成帐户 62, 71
 组成员资格 46
用户管理器 8
 个性化引擎 86
 扩展 65
 Telnet 控制台命令 89
用户管理器, LDAP 87
用户属性
 定义 73
 定制, 更改 74
 更改 45

[Z]

帐户, 用户
 请参阅 用户
主题
 缺省值, 指定 71
注释集合
 创建和管理 75
转换实用程序, 存储库 105, 106
子组
 创建 49
 已定义 49
组
 成员资格 46
 创建 49
 更改 50
 角色, 更改成员资格 54
 删除 51
 子组, 了解 49
 组和用户成员资格, 更改 46

[特别字符]

“管理”选项卡 15
“以操作系统方式操作”用户权限 39

“应用程序”选项卡 15, 105, 116, 119
“组装”选项卡 18

A

ACL,
 [请参阅 角色](#)
ADD 命令 125
AlphabloxAuthenticatedUser 57
AlphabloxUser 57
Apache
 安全性 57
Apache HTTP Server
 注册应用程序 33
Apache Tomcat
 迁移应用程序 17
Application Studio
 概述 18
authenticate() 方法, IUser 接口 98
authorize() 方法, IUser 接口 99

B

Blox
 DB2 Alphablox 应用程序中的组件 12
Blox 样本程序
 “组装”选项卡上的示例链接 18
Blox 组件
 描述的 4
blox.tld 文件 28

C

CLUSTER SHUTDOWN 命令
 描述 121
CommentsBlox
 创建注释集合 75
containsGroup() 方法, IGroup 接口 101
containsUser() 方法, IGroup 接口 102
CREATE 命令 125

D

DB2 Alphablox
 安全性,
 [请参阅 安全性](#)
 存储库, 数据库 105
 定制属性,
 [请参阅 定制属性](#)
 端口, 指定 71
 访问 21
 服务器空闲持续时间 70
 服务, 已定义的 7
 概述 1
 管理器, 已定义的 7

DB2 Alphablox (续)
 管理任务 21
 基于 Web 服务器的安全性, 配置 62
 集群 119
 控制台,
 [请参阅 控制台](#)
 启动 21
 启动属性, 配置 69
 认证方式和安全方式 55
 体系结构 7
 停止 21
 系统属性, 配置 70
 系统需求,
 [请参阅 安装指南](#)
 用户属性,
 [请参阅 用户属性](#)
 用户, 创建 45
DB2 Alphablox 存储库,
 [请参阅 存储库, Alphablox Analytics](#)
DB2 Alphablox 立方体管理器
 最大立方体数, 指定 72
DB2 Alphablox 应用程序,
 [请参阅 应用程序](#)
DB2 Alphablox 主页
 “管理”选项卡 15
 “应用程序”选项卡 15, 105, 116, 119
 “组装”选项卡 18
DB2 Alphablox Cube Server
 数据源, 定义 37
 [请参阅 Cube Server 管理员指南](#)
DB2 OLAP Server
 客户机库, 更新 22
 控制台命令 127
 显示客户机库版本 127
DB2 UDB 数据源, 定义 37
DELETE 命令 125
DELETE OUTLINECACHE 命令, DB2 OLAP Server 128
DELETE OUTLINECACHE 命令, Essbase 128
DHTML 客户机
 已知问题 25
 级联样式表 (CSS) 25

E

Essbase
 客户机库, 更新 22
 控制台命令 127
 显示客户机库版本 127
Essbase 客户机库实用程序
 使用 22
EXIT 命令 125

F

FastForward

- 报告类型 133
- 创建新应用程序 132
- 管理 133
- 管理员角色, 修改 132
- 全局设置 135
- 系统需求 132
- 应用程序日志, 使用 136
- 应用程序属性, 管理 135
- 用户角色 131

findGroup() 方法, IUserManager 接口 95

findUser() 方法, IUserManager 接口 95

G

GET 命令 125

GET SERVICE CLUSTER 命令

- 描述 121
- 示例 120

getEmail() 方法, IUser 接口 99

getExternalProperties() 方法, IUserManager 接口 95

getFullName() 方法, IUser 接口 99

getName() 方法, IGroup 接口 102

getName() 方法, IUser 接口 100

getPassword() 方法, IUser 接口 100

getPrincipleUserName() 方法, IUserManager 接口 96

getPropertiesSubset() 方法, IGroup 接口 102

getPropertiesSubset() 方法, IUser 接口 100

Guest 用户

- 限制对应用程序的访问 57

H

hasExternalEditor() 方法, IUserManager 接口 96

HELP 命令 125

I

ISCLUSTERED 命令 121

isUserInRole() 方法, IUser 接口 101

J

J2EE 应用程序

- 导入 35

JAR 文件 18

JDBC

- 跟踪 41
- 驱动程序, 更新版本 42

JDBC 驱动程序

- 类路径设置, 修改 42
- 添加其他 42

JNDI 域

- 安全模型 56

K

KILL 命令 125

L

LDAP 用户管理器 87

LEADHOST 命令 121

LOAD 命令 125

LOCK 命令 126

M

MAXHOSTS 命令 121

MESSAGE 命令 126

Microsoft Analysis Services

- 连接池 113
- 认证, 设置 38
- 数据源, 定义 37
- 性能 113
- Windows 用户权限, 添加 39

Microsoft IIS

- 安全设置 58
- 安全设置, 启用 59
- 目录浏览, 禁用 63
- 目录权限, 设置 63
- 匿名用户权限, 限制 60
- 用户帐户, 自动生成 62
- 注册应用程序 34

Microsoft SQL Server

- 认证类型 38
- 数据源, 定义 37

N

N 层体系结构 6

O

OBJECTS 命令 126

OLAP 术语与概念 137

Oracle 数据源, 定义 37

P

PDF 报告

- 创建远程 PDF 处理器 78

Portal 主题实用程序

- 访问 16

PORTNUM 命令 121

R

refresh() 方法, IGroup 接口 103
refresh() 方法, IUser 接口 101
RELEASE 命令 126
REMOVE 命令 126
REPORT 命令 126
RESOLVEALIASSTOBASEMEMBERS 控制台命令 127
RESUME 命令 126
resume() 方法, IUserManager 接口 96
RUN 命令
 语法 126

S

SAVE 命令 126
SET 命令 126
SET SERVICE CLUSTER 命令 121
setCaseSensitiveGroups() 方法, IUserManager 接口 97
setCaseSensitiveUsers() 方法, IUserManager 接口 97
SHOW 命令 127
SHOW HOSTS 命令
 描述 121
 描述的 127
SHOW OUTLINECACHE 命令, DB2 OLAP Server 128
SHOW OUTLINECACHE 命令, Essbase 128
SMTP 服务器, 指定 71
SQL Server, Microsoft
 认证类型 38
START 命令 127
STARTUPTIME 命令 121
start() 方法, IUserManager 接口 97
STATISTICS 命令 127
STOP 命令 127
stop() 方法, IUserManager 接口 98
Sun iPlanet
 安全选项 58
 注册应用程序 34
SUSPEND 命令 127
suspend() 方法, IUserManager 接口 98
Sybase 数据源, 定义 37
Sybase SQL 脚本 41

T

Telnet 控制台 123
 端口, 缺省值 71

W

Web 服务器
 注册应用程序 33
 DB2 Alphablox, 配置安全性 62
 Microsoft IIS 安全设置 58
 URL 前缀, 指定的 70

WebLogic
 连接池 117
WebSphere
 应用程序, 定义 29
WebSphere Portal Server
 主题 16
WEB-INF 目录 28
web.xml 文件 28
Windows
 服务, 当使用 Microsoft Analysis Services 时配置 40
 用户权限, 为 Microsoft Analysis Services 设置 38



程序号: 5724-L14

Printed in China

S151-0143-03



Spine information:



IBM DB2 Alphablox

管理员指南

版本 8.4