版本信息

日期	版本	描述	作者
2004-03-01	v1.0	最初版本	LDAPChina.com

The LDAP URL Format
LDAP URL 格式

1、本备忘录的状态（Status of this Memo）

本文档定义了一个用于 Internet 通讯的 Internet 标准跟踪协议，为了发展的需要讨论和建议。对于这个协议的状况和地位请参照 Internet 官方协议标准（"Internet Official Protocol Standards"（STD 1））的当前版。这个备忘录的传播是不受限制的。

版权提示（Copyright Notice）

IESG 提示（IESG Note）

本文档描述将一种同时提供读和更新访问的目录访问协议。更新访问需要安全认证，但这个文档并不强制实现任何安全认证机制。

与 RFC2026 的 4.4.1 节相同，本规范正在被 IESG 批准期间，作为被提议的标准，可能并不限于本文档所述内容。原因如下：

a、 鼓励在它们被发布前，实现和交互测试这些协议（带有或没有更新访问）；

b、 鼓励在只读的应用程序中配置和使用这些协议。（例如，在某些应用程序中，目录的更新访问使用某些其它安全的机制而不是 LDAP，而使用 LDAPv3 被作为对目录的查询语言）；

c、 避免阻碍别的 Internet 标准追踪协议的发展和发布。（这些协议需要 LDAPv3 的目录服务器的查询能力，而不是更新能力）

需要警告读者的是，直到强制的验证机制被标准化之前，根据本规范编写的客户端和服务器实现了更新功能的话，它们的互操作性可能是不可靠的，或者仅提供在认证需要极度弱化的时候的互操作性。

因此在具有强制认证的 LDAPv3 未成为一个 RFC 而被批准或发布之前，不鼓励实现者发布一个实现了更新功能的 LDAPv3 的客户端和服务器。

目 录

2、摘要（Abstract）

LDAP 是轻目录访问协议，在参考文档[1]，[2]和[3]中定义。本文档描述了 LDAP 统一资源地址（URL）的格式。该格式描述了执行从一个 LDAP 目录检索信息的 LDAP 查询操作。本文档替代了 RFC 1959。它为 LDAPv3 更新了 LDAP URL 格式，并解释了如何处理 LDAP URL。本文档也为 LDAP URL 定义了扩展机制，这样未来的文档能扩展它们的功能，例如，若定义了一个新的 LDAPv3 扩展，就可以提供对它们的访问。

在本文档中使用的关键字"**MUST**"，"**MAY**"和"**SHOULD**"的含意与参考文档[6]中描述的相同。

3、URL 定义（URL Definition）

一个 LDAP URL 使用协议的前缀"ldap"开始，并使用下面的语法定义。

```
ldapurl       = scheme "://" [hostport] ["/"
                    [dn ["?" [attributes] ["?" [scope]
                    ["?" [filter] ["?" extensions]]]]]]
scheme        = "ldap"
attributes = attrdesc *("," attrdesc)
scope         = "base" / "one" / "sub"
dn            = distinguishedName from Section 3 of [1]
hostport      = hostport from Section 5 of RFC 1738 [5]
attrdesc      = AttributeDescription from Section 4.1.5 of [2]
filter        = filter from Section 4 of [4]
extensions = extension *("," extension)
extension     = ["!"] extype ["=" exvalue]
extype        = token / xtoken
exvalue       = LDAPString from section 4.1.2 of [2]
token         = oid from section 4.1 of [3]
xtoken        = ("X-" / "x-") token
```

前缀"ldap"声明了存在于 LDAP 服务器中的一个或多个条目，该 LDAP 服务器运行在给定主机的给定端口号上的。缺省的 LDAP 端口号是 TCP 端口 389。如果示给出主机端口号，客户端必须有一些预备知识，以连接到一个适合的 LDAP 服务器。

dn 是使用字符串格式的 LDAP 分辨名（参考文档[1]中描述）。它标识了 LDAP 查询的基准对象（base object）。

attributes 结构用于标识那些属性应该从条目中（或多个条目中）返回。单个 attrdesc 名称与参考文档[2]中 AttributeDescription 的定义相同。如果 attributes 部分被忽略，条目（或

多个条目）的所有用户属性应该被请求（例如，通过设置 LDAP 查询请求中的 attributes 域的 AttributeDescriptionList 为一个 NULL 列表，或（在 LDAPv3 里）通过请求特殊的属性名"*"）。

scope 部分用于指定在给定的服务器上执行的查询的范围。允许的范围如下："base"指基准对象（base object）查询，"one"指一层子条目查询，或者"sub"指子树（subtree）查询。如果 scope 被忽略，则认为是"base"范围。

filter 被用于指定查询过滤器，该过滤器在查询时应用到指定范围内的条目上。它的格式在参考文档[4]中定义。如果 filter 被忽略，则认为过滤器是"(objectClass=*)"。

extensions 部分提供了带有扩展机制的 LDAP URL，它允许 URL 的能力在未来可以被扩展。扩展是一个简单的由逗号分隔的（comma-separated）type=value 对儿列表，=value 部分对于不需要它的选项**可以**（MAY）被忽略。每一个 type=value 对儿是一个扩展。这些 LDAP URL 扩展不必与某个 LDAPv3 扩展机制相关。处理该 URL 的客户端可以支持，也可以不支持扩展。以'!'（ASCII 33）字符为前缀的 extension 是关键（critical）扩展。不以'!'（ASCII 33）字符为前缀的扩展是非关键（non-critical）扩展。

如果某个扩展被客户端所支持，若该扩展是关键扩展，则客户端**必须**（MUST）遵守该扩展。客户端**应该**（SHOULD）遵守它所支持的非关键扩展。

如果某个扩展不被客户端支持，若该扩展是关键扩展，客户端**绝不**（MUST NOT）能处理该 URL。如果不支持的扩展是非关键扩展，客户端**必须**（MUST）忽略该扩展。

如果一个关键扩展不能被客户端成功处理，客户端**绝不**（MUST NOT）能处理该 URL。如果一个非关键扩展不能被客户端成功处理，客户端**应该**（SHOULD）忽略该扩展。

带有"X-"或"x-"前缀的扩展类型保留，在通讯个体之间的双边协议（bilateral agreements）中使用。其它扩展类型**必须**（MUST）在本文档中定义，或在其它标准追踪文档中定义。

本文档的下一节中将定义一个 LDAP URL 扩展。其它文档或本文档的未来版本**可能**（MAY）定义其它扩展。

应注意，任何 URL 非法字符（URL-illegal character）（例如，空格），URL 特殊字符（URL special character）（类似于 RFC 1738 的 2.2 节中定义的）和保留字符'?'（ASCII 63）出现在 dn 中、出现在 filter 中、或出现在 LDAP URL 中的其它元素中，**必须**（MUST）使用在 RFC 1738（参考文档[5]）中描述的%方法进行转义。如果一个','（逗号字符）出现在一个 extension 值的中，该字符也**必须**（MUST）使用%方法进行转义。

4、Bindname 扩展（The Bindname Extension）

本节定义了一个 LDAP URL 扩展，该扩展代表了一个客户端使用的分辨名，客户端在解析一个 LDAP URL 期间认证到一个 LDAP 目录时使用该分辨名。客户端**可以**（MAY）实现该扩展。

该扩展类型是"bindname"。extension 值是认证时使用的目录条目的分辨名，与上面语法里描述的 dn 的格式相同。dn 可以是空字符串，指定为非认证访问。该扩展可以是关键扩展（以'!'字符为前缀）或非关键扩展（不以'!'字符前缀）。

如果 bindname 扩展是关键扩展，解析 URL 的客户端**必须**（MUST）使用给定的分辨名和一个适当的认证方法认证到目录。注意对于一个 NULL 分辨名，**可能**（MAY）需要不绑定（no bind），以获得对目录的匿名访问。如果该扩展是非关键扩展，客户端**可以**（MAY）使用给定的分辨名绑定到目录。

5、URL 处理（URL Processing）

本节描述客户端**应该**（SHOULD）如何处理一个 LDAP URL。

首先，客户端得到 URL 所标识的 LDAP 服务器的连接，或者如果没有 LDAP 服务器被显示地指出，客户端得到一个由客户端选择的 LDAPP 服务器的连接。该连接**可以**（MAY）专门为解析 URL 的目的而打开，或者客户端也**可以**（MAY）重用一个已经打开着的连接。该连接**可以**（MAY）提供保密性、完整性或其它服务，例如，使用 TLS。如果在 URL 中未指明，则是否使用安全服务由客户端决定。

下一步，客户端向 LDAP 服务器认证它自己。除非在 URL 中包含了一个非空值的 bindname 扩展的关键扩展，否则这一步是可选的。如果给出了一个 bindname 扩展，客户端根据上一节所述进行处理。

如果 bindname 扩展未被指定，客户端**可以**（MAY）使用它自己选择的适当的 dn 和认证方法绑定到服务器（包括 NULL 认证）。

下一步，客户端执行 URL 中指定的 LDAP 查询。在 LDAP 协议的查询请求中的附加域，例如在 URL 规范中 sizelimit，timelimit，deref 和任何在 URL 中未指定的或非缺省的域，**可以**（MAY）根据客户端的判断进行设置。

一旦查询执行完成，客户端**可以**（MAY）关闭到 LDAP 服务器的连接，或者客户端**可以**（MAY）保持该连接为打开状态以便将来使用。

6、范例（Examples）

下面是一些使用上面定义的格式的LDAP URL的例子。第一个例子是一个指向Michigan大学的条目的 LDAP URL，该条目在客户端选择的 LDAP 服务器上有效：

ldap:///o=University%20of%20Michigan,c=US

下一个例子是一个指向特定 LDAP 服务器上的 Michigan 大学条目的 LDAP URL：

ldap://ldap.itd.umich.edu/o=University%20of%20Michigan,c=US

两个 URL 均对条目"o=University of Michigan, c=US"使用"(objectclass=*)"过滤器进行基准对象（base object）查询，并请求所有的属性。

下一个例子是一个仅指向 Michigan 大学条目的 postalAddress 属性的 LDAP URL：

ldap://ldap.itd.umich.edu/o=University%20of%20Michigan,
c=US?postalAddress

在上一个例子中，对应的 LDAP 查询操作与前面的例子相同，但仅请求 postalAddress 属性。

下一个例子中，LDAP URL 指向一个条目集，该条目集通过执行下列查询得到：给定6666 端口的服务器，基准对象为 Michigan 大学，范围是所有子树，过滤器为 common name（译者注：公共名，即 cn）为"Babs Jensen"的条目，并返回所有的属性：

ldap://host.com:6666/o=University%20of%20Michigan,
c=US??sub?(cn=Babs%20Jensen)

下一例子是指向 c=GB 条目的所有一级子条目的 LDAP URL：

ldap://ldap.itd.umich.edu/c=GB?objectClass?one

ObjectClass 属性被请求，并与条目一起返回，使用缺省的过滤器"(objectclass=*)"。

下一个例子是检索"o=Question?,c=US"条目的 mail 属性的 LDAP URL，演示说明了对保留字符'?'使用转义机制。

ldap://ldap.question.com/o=Question%3f,c=US?mail

下一个例子演示了 LDAP 和 URL 引用机制之间的交互。

ldap://ldap.netscape.com/o=Babsco,c=US??(int=%5c00%5c00%5c00%5c04)

这个例子中，过滤器使用 LDAP 转义机制 \，对值中的三个 0 和 null 字节进行编码。在 LDAP 中，过滤器将被写成（int=\00\00\00\04）。因为\字符必须在 URL 中被转义，\'s 字符在 URL 编码中被转义为%5c。

最后一个例子展示了 bindname 扩展的使用，它用于指定在解析 URL 时客户端应该用来证明身份的 dn。

ldap:///??sub??bindname=cn=Manager%2co=Foo
ldap:///??sub??!bindname=cn=Manager%2co=Foo

两个 URL 是相同的，但第二个标记 bindname 扩展为关键扩展。注意在 bindname 扩展中，使用了%编码方法对分辨名值中的逗号进行编码。

7、安全考虑（Security Considerations）

总体 URL 安全考虑在参考文档[5]中关于 LDAP URL 部分。

在处理 LDAP URL 时安全机制的使用需要特别的小心，因为客户端可以通过 URL 遇到许多不同的服务器，同时 URL 很可能是自动处理的，没有用户的干预。一个客户端**应该**（SHOULD）有一个用户可配置策略（user-configurable policy），该策略决定使用哪个安全机制去连接哪个服务器，并且**不应该**（SHOULD NOT）产生与该策略不一致的连接。

发送认证信息时，无论什么机制都可能破坏一个用户的隐私要求。当不存在允许认证信息发送到服务器的特定策略时，客户端应该使用匿名连接。（注意符合以前的 LDAP URL 规范的客户端（所有的连接是匿名的和不受保护的）与本规范是一致的，它们具有简单的缺省安全策略。）

某些认证方法，特别是发送到服务器的重用密码，可能把易于滥用（easily-abused）的信息暴露给远程服务或传输中的监听者，除非得到了策略的显示地允许，否则不应该在 URL 的处理中被使用。认证信息在经过用户批准后使用是非常适当的。而不暴露敏感信息的健壮的认证方法的使用更受到欢迎。

LDAP URL 格式允许在评估 LDAP URL 时执行任意 LDAP 查询操作。追踪一个 LDAP URL 可能导致不可预测的结果，例如，大量数据的检索，长时间的（long-lived）查询初始化，等等。解析 LDAP URL 时需要考虑的安全问题与处理 LDAP 查询请求时的基本相同。

8、感谢（Acknowledgements）

LDAP URL 格式最初是 Michigan 大学定义。这个材料是基于国家科学基金（National Science Foundation under Grant No. NCR-9416667）支持的工作之上。对 Michigan 大学和国家科学基金的支持均表示诚挚的谢意。

几位人士对本文档提出了有价值的意见。特别是 RL "Bob" Morgan 和 Mark Wahl 的贡献应得到特别的感谢。

9、参考书目（References）

[1] Wahl, M., Kille, S., and T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", RFC 2253, December 1997.

[2] Wahl, M., Howes, T., and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[3] Wahl, M., Coulbeck, A., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997.

[4] Howes, T., "A String Representation of LDAP Search Filters", RFC 2254, December 1997.

[5] Berners-Lee, T., Masinter, L. and M. McCahill, "Uniform Resource Locators (URL)," RFC 1738, December 1994.

[6] Bradner, S., "Key Words for use in RFCs to Indicate Requirement Levels," RFC 2119, March 1997.

10、作者地址（Authors' Addresses）

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Rd.
Mountain View, CA 94043
USA
Phone: +1 415 937-3419
EMail: howes@netscape.com

Mark Smith

Netscape Communications Corp.
501 E. Middlefield Rd.
Mountain View, CA 94043
USA
Phone: +1 415 937-3477
EMail: mcs@netscape.com

11、完整的版权声明（Full Copyright Statement）

略