



# JNDI

Java Naming and Directory Interface

*Allen Long*

*Email: allen@huihoo.com*

<http://www.huihoo.com>

2004-04



JAVA

# 内容安排



- 命名和目录服务
- JNDI一览
- 功能和代码事例
- JNDI提供者
- 资源
- 引用



# Naming Service



关联名字和对象 (bind)

通过名字检索对象(resolve)

举例:

- RMI Registry
- CORBA Naming Service (COSNaming)
- Domain Name Service (DNS)
- Filesystem



# 上下文(Contexts)



- 包含一系列绑定和查询操作
- 拥有自己的命名约定
- 举例
  - Filesystem directory: “/”
  - DNS domain: “edu”
  - LDAP: “c=us”
- 命名系统是一个上下文集合 (相同类型)
  - names in a particular system = namespace



JAVA

# Names



- 关联一个对象的逻辑标识
- Localhoat=>127.0.0.1
- 命名约定通过命名系统定义
  - /dir/dir2/file for UNIX
  - drive:\dir\string for DOS
  - cn=dan, o=ISP for LDAP
  - aplcenmp.apl.jhu.edu for DNS



JAVA

# Names



- 原子(Atomic)Name
  - name used in a binding
  - only meaningful in a context
  - Ex. filename in a directory
- 复合(Compound)Name
  - sequence of atomic names
  - /usr/bin/l
  - conforms to naming convention of name space



# Names



- 混合(Composite)Name
  - Spans multiple naming systems
  - `http://www.apl.jhu.edu/~weimer/`
    - URL scheme id: `http`
    - DNS: `www.apl.jhu.edu`
    - UNIX and webserver:  
`/usr/weimer/public_html`
- 分解(Resolution)  
Object o = `ctx.lookup("usr/weimer/public_html");`



# 目录服务



- 目录对象表示一个对象
- 对象可能给出属性(Objects may be given attributes)
- 能获得一个对象的属性和通过属性找到对象
  
- 举例
  - X.500 - ISO standard. Complex.
  - LDAP - Simplified X.500 over TCP/IP
  - NDS - Novell Directory Services.
  - NIS - Directory service for Solaris





# JNDI的目标



- 提供一个一致的API去访问不同的命名和目录服务  
客户端只需要学习一种API
- 不同的命名和目录服务能被组合进一个逻辑系统
- 不需要修改客户端就能加入新的命名服务实现



# JNDI 架构



Java Client Application

JNDI API

JNDI Naming Manager

JNDI Service Provider Interface

DNS

LDAP

RMI

CORBA

Anything



# JNDI 架构



- 应用程序接口 (API)
  - API for client programmer
  - 统一接口(Unifying interface)
- Service Provider Interface (SPI)
  - For vendors enabling JNDI access to their naming/directory service



# JNDI API



- Included in Java 2 SDK v 1.3
- Have to download for JDK 1.1 and SDK 1.2
- Packages
  - javax.naming
  - javax.naming.directory
  - javax.naming.event
  - javax.naming.ldap
  - javax.naming.spi



# 服务提供者(Service Providers)



- Implementation for underlying products
- Included in Java 2 SDK 1.3
  - LDAP
  - COSNaming
  - JNDI over RMI Registry
- Have to download for previous versions
  - <http://java.sun.com/products/jndi/>



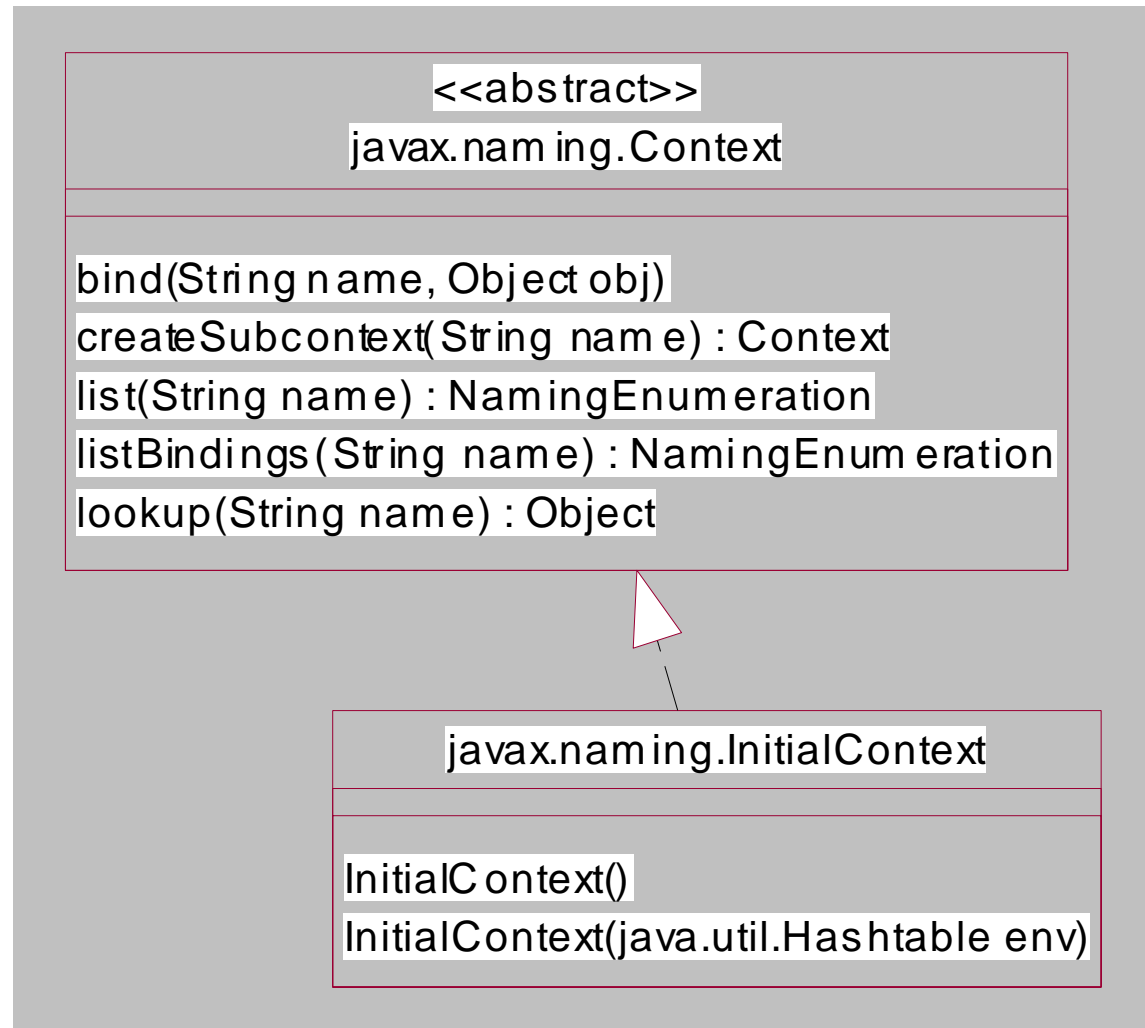
# 共同JNDI任务(Tasks)



- Obtain reference to initial context
- Context Operations
  - List children of a context
  - Bind names to objects
  - Lookup objects via name
  - Create/destroy contexts
  - Note: NamingException can be thrown from most Context operations



# JNDI 上下文(Contexts)



# 获得初始上下文



```
import javax.naming.*;
```

```
java.util.Properties props = new java.util.Properties();  
props.put(Context.INITIAL_CONTEXT_FACTORY,  
           "com.sun.jndi.fscontext.RefFSContextFactory");  
props.put(Context.PROVIDER_URL, "file:///");
```

```
Context initContext = new InitialContext( props );
```



# 初始上下文



- Starting point in the namespace
- All operations performed are relative to the initial context
- Specify service provider with property

```
props.put(Context.INITIAL_CONTEXT_FACTORY”  
,  
“com.sun.jndi.fscontext.RefFSContextFactory” );
```



JAVA

# 初始上下文



JAVA

- Specify provider-specific properties
- LDAP
  - props.put(Context.PROVIDER\_URL, “ldap://host:port”);
  - props.put(Context.SECURITY\_PRINCIPAL, “user” );
  - props.put(Context.SECURITY\_CREDENTIALS, “password”);
- File System
  - props.put(Context.PROVIDER\_URL, “file://tmp/”);
- Create
  - InitialContext initContext = new InitialContext( props );

# 列举children

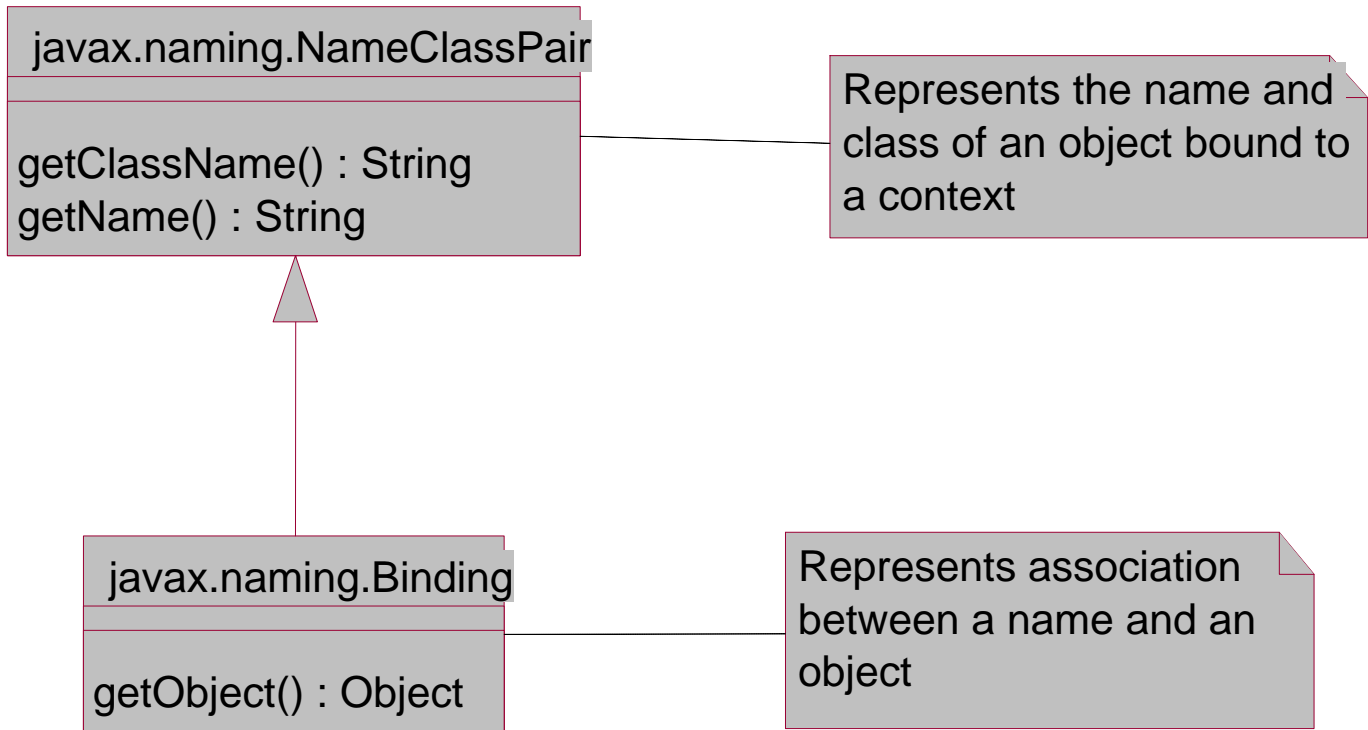


```
NamingEnumeration children = initContext.list("");
while( children.hasMore() )
{
    NameClassPair nc = (NameClassPair) children.next();
    System.out.println(nc.getName());
    System.out.println(nc.getClassName());
}
```

- list() returns a NamingEnumeration of NameClassPair objects
- listBindings() returns a NamingEnumeration of Binding objects



# NameClassPair and Binding



# NamingEnumeration



- Extends `java.util.Enumeration`
- Throws exception when no more entries in the enumeration
- Allows partial results to be returned and exception throws upon access where the problem occurred
  - i.e. `hasMore()` will throw the exception
- Limit of search with search controls



# 绑定名字到一个对象



- Bind name to object within a context

```
File f = new File("/tmp/dan");  
tmpDirContext.bind("dan", f );
```

- `NameAlreadyBoundException` occurs if “dan” is already bound in the tmp context
- Can use `unbind()` or `rebind()`



# 对象查询



## Lookup object in a context

- String name; Object o = `initContext.lookup(name)`;
- name can be compound
  - “/usr/tmp/dir/myfile”
  - Separator is not standardized !
- Class returned is up to provider !
- Filesystem provider returns
  - File for files
  - [Ref]FsContext for directories
- Novell NDS can return `OrganizationalDirContext`
- Containers should implement `Context`



JAVA

# 上下文生命周期操作



- `ctx.createSubcontext(String name)`
  - JNDI Provider chooses actual implementation of new context.
  - Ex. File System Provider will create a directory
  - No way to create a file
- `ctx.destroySubcontext(String name)`
  - Can not destroy current object with name of “”
  - Ex. Remove directory in a filesystem



# 目录操作



`javax.naming.directory.DirContext`

extends `javax.naming.Context`

examine/search attributes associated with a directory object

`javax.naming.directory.Attribute`

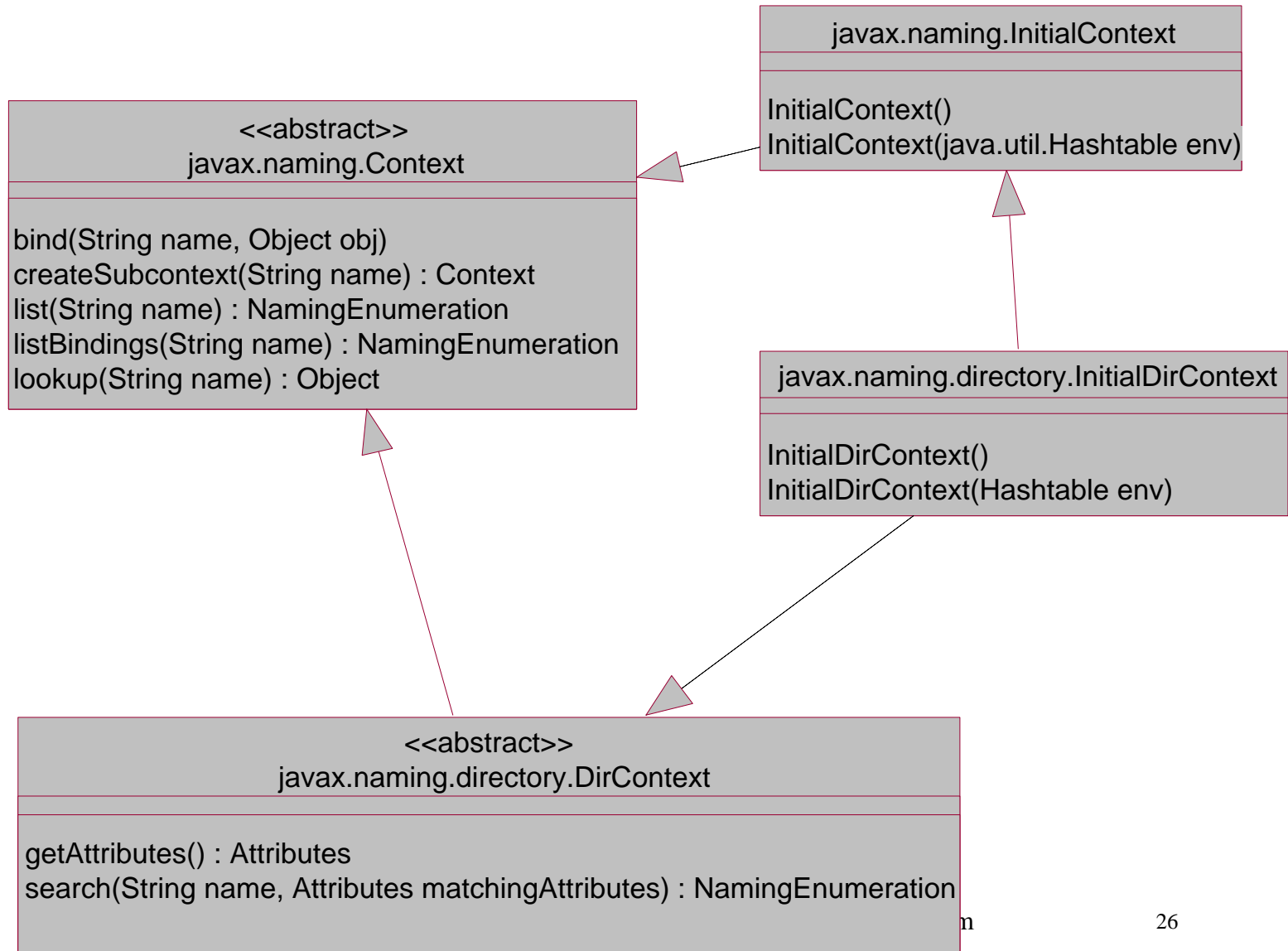
name and set of values

`getAttributes( String name ) : Attributes`

`modifyAttributes( String name, .... )`



# 目录操作



# 属性



- DirContext contains an Attributes object for each contained binding

- Common LDAP Attributes

c = Country

o = Organization

ou = Organizational Unit

cn = Common Name (typically first or full name)

sn = User's surname

- Attribute Name

–referred to as attribute id

- determines type of attribute (attribute type definition)
- attribute syntax definition specifies the syntax for the attribute's value and whether it can have multiple values

- reverse lookup; content-based searching



# JNDI and RMI



- Can put standard interface on RMI registry

```
Properties props = new Properties();
props.put(Context.INITIAL_CONTEXT_FACTORY,
    "com.sun.jndi.rmi.registry.RegistryContextFactory");
props.put(Context.PROVIDER_URL, "rmi://host:port");
Context ctx = new InitialContext(props);
ctx.rebind("Teller", someRMIObject );
.....
Teller t = (Teller) ctx.lookup("Teller");
```



# JNDI and JDBC



JAVA

- JDBC Data Source
  - 发展(Evolution) of the JDBC Driver Manager
  - Typically stored in a JNDI tree
  - connection pooling support
  - Methods
    - `ds.getConnection(String name, String password )`
    - `conn.close()`

# JNDI and EJB



- Home objects for beans are stored in JNDI
- In order to create a bean:

```
InitialContext ic = new InitialContext( props );
```

```
TellerHome th = (TellerHome)
```

```
    ic.lookup("TellerHome");
```

```
th.transfer( ... );
```

```
// Remote stub typically returned to client
```

# JNDI and J2EE Applications



- InitialContext
  - supplied by Container
  - accessed by Component using default ctor
    - `InitialContext rootCtx = new InitialContext();`
- Properties
  - supplied to Container in deployment descriptor
  - accessed by Component through special context
    - `Object object = rootCtx.lookup("java:comp/env/myObject");`

# JNDI 1.2 功能



- 事件通告(Event Notification)
  - LDAPv3 Extensions and Controls
  - Service Provider Support
- `javax.naming.spi`



# JNDI Providers



- LDAP
  - `com.novell.naming.service.nds.NdsInitialContextFactory`
- NIS
  - `com.sun.jndi.nis.NISCtxFactory`
- NIS+ (future)
- RMI
  - `com.sun.jndi.rmi.registry.RegistryContextFactory`
- COSNaming
- Files
  - `com.sun.jndi.fscontext.[Ref]FSContextFactory`

# LDAP Products



## Publicly accessible

-ldap://ldap.Bigfoot.com

-ldap://ldap.four11.com

-ldap://ldap.InfoSpace.com



# 总结



- JNDI客户端能访问以下资源:
  - printers
  - fax machines
  - databases (JDBC 2.0 extensions)
  - user credentials
  - object references
- contained in multiple underlying naming service implementations using the same API



JAVA

# 参考资料



- <http://java.sun.com/products/jndi>  
SUN公司的JNDI站点
- <http://www.huihoo.org/jfox/jfoxns/>  
JFoxNS (JFox命名服务)
- <http://www.huihoo.com>  
国内一个关于中间件的专业站点



# 结束



## 谢谢大家！

Allen@huihoo.com

<http://www.huihoo.com>

