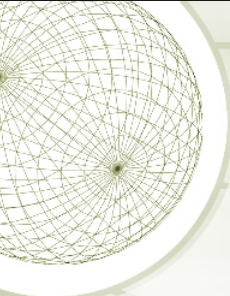


***DNS-SD in Java***

**Info 341 Networking and  
Distributed Applications**



***Service Discovery  
Browsing Services***

## DNS-SD Command

```

dns-sd -E                (Enumerate recommended registration domains)
dns-sd -F                (Enumerate recommended browsing domains)
dns-sd -B <Type> <Domain> (Browse for services instances)
dns-sd -L <Name> <Type> <Domain> (Look up a service instance)
dns-sd -R <Name> <Type> <Domain> <Port> [<TXT>...] (Register a service)
dns-sd -P <Name> <Type> <Domain> <Port> <Host> <IP> [<TXT>...] (Proxy)
dns-sd -Z                <Type> <Domain> (Output results in Zone File format)
dns-sd -Q <FQDN> <rrtype> <rrclass> (Generic query for any record type)
dns-sd -C <FQDN> <rrtype> <rrclass> (Query; reconfirming each result)
dns-sd -X udp/tcp/udptcp <IntPort> <ExtPort> <TTL> (NAT Port Mapping)
dns-sd -G v4/v6/v4v6 <Hostname> (Get address information for hostname)
dns-sd -V                (Get version of currently running daemon / system service)
dns-sd -A                (Test Adding/Updating/Deleting a record)
dns-sd -U                (Test updating a TXT record)
dns-sd -N                (Test adding a large NULL record)
dns-sd -T                (Test creating a large TXT record)
dns-sd -M                (Test creating a registration with multiple TXT records)
dns-sd -I                (Test registering and then immediately updating TXT record)
dns-sd -S                (Test multiple operations on a shared socket)

```

## Parameters

- ★ Name
  - ★ A string, text to name the service
- ★ Domain
  - ★ “local” is the special domain for mDNS and is the only one supported
- ★ Port
  - ★ Protocol port for the service



## Parameters

### ★ Type

- ★ Special strings that indicate a service
- ★ Must be registered
  - ★ <http://www.dns-sd.org/ServiceTypes.html>

### ★ Examples

```
_http._tcp  
_daap._tcp  
_ftp._tcp  
_ichat._tcp  
_ldap._tcp  
_printer._tcp
```



## Browse

### ★ `dns-sd -B <Type> <Domain>`

- ★ `dns-sd -B _http._tcp`
- ★ `dns-sd -B _printer._tcp`
- ★ `dns-sd -B _ftp._tcp`
- ★ `dns-sd -B _smb._tcp`



## *Register*

- ★ `dns-sd -R <Name> <Type> <Domain> <Port>`
- ✦ `dns-sd -R "Service 1" _fake._tcp local 25001`
- ✦ `dns-sd -R "Service 1" _fake._tcp local 25002`
- ✦ `dns-sd -R "Service 2" _fake._tcp local 25003`



## *Java APIs*



## *DNSSD Classes*

- ★ In package `com.apple.dnssd`

```
import com.apple.dnssd.*;
```
- ★ Two key classes (and several interfaces)
  - ★ `DNSSD`
    - ★ Mostly static factory class
  - ★ `TXTRecord`
    - ★ Create and manage DNS TXT records
- ★ Almost all DNSSD calls are asynchronous
  - ★ They create a *\*new\** Thread



## *Three Critical Activities*

- ★ Register Service
  - ★ Make a service announcement
- ★ Browse for Services
  - ★ Find services on the network
- ★ Resolve Service
  - ★ Given a service name, find host & port

## Browsing

### ★ Mechanism for finding services

```
DNSSDService browser = null;
browser = DNSSD.browse(sd_service, myBrowseListener);
```

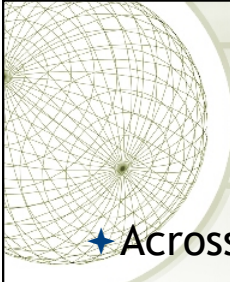
- ★ “browser” is a thread that calls methods on myBrowseListener
  - ✦ serviceFound()
  - ✦ serviceLost()

## Browse Listener Interface

### ★ BrowseListener

- ✦ The more complex part of browsing

```
public class MyBrowseListener implements BrowseListener {
    public void serviceFound (DNSSDService br, int flags, int ifIndex, String serviceName,
        String regType, String domain) {
        ...
    }
    public void serviceLost (DNSSDService br, int flags, int ifIndex, String serviceName,
        String regType, String domain) {
        ...
    }
    public void operationFailed (DNSSDService service, int errorCode) {
        ...
    }
    <other methods>
}
```



## *Things to Notice*

- ★ Across Register, Browse, & Resolve
  - ◆ Initiate asynchronous services (threads)
    - ◆ Must eventually call stop()
  - ◆ Pass some form of Listener object
  - ◆ Listener interfaces implement callback methods
  - ◆ All Listeners implement “operationFailed”