

Ember+ Service Discovery

The Ember+ group made the commitment (in April 2012) to use **Zero Configuration Networking** for service discovery of Ember+ devices. This only applies if service discovery is required at all.

Service discovery shall answer the following questions:

- Which devices provide a certain service?
- How can these services be reached?

A service provider must register its services by the DNS daemon/server which answers the requests. A service consumer may then browse for services in the network. After browsing and resolving a certain service, the consumer is able to connect to the provider (by using the host name or the IP address) and use the service.

The service type to use for Ember+ is `_ember._tcp` and/or `_ember._udp`.

Two operating modes are possible:

- Multicast communication is used (mDNS) and requests are answered directly by the local DNS daemon.
- Unicast communication is used and requests are answered by a DNS server. The local DNS daemon must register the services by the DNS server (which must be enabled / supported).

The operating mode is typically given by the domain, either `.local` (mDNS) or `.<my-domain>` (DNS).

In the following sample scenario the wireless microphone hubs are expected to appear in a list of the mixing console when they are ready. The user may choose to connect (or even automatically connect) to the available wireless microphone hubs and display information about battery state, frequencies, etc.

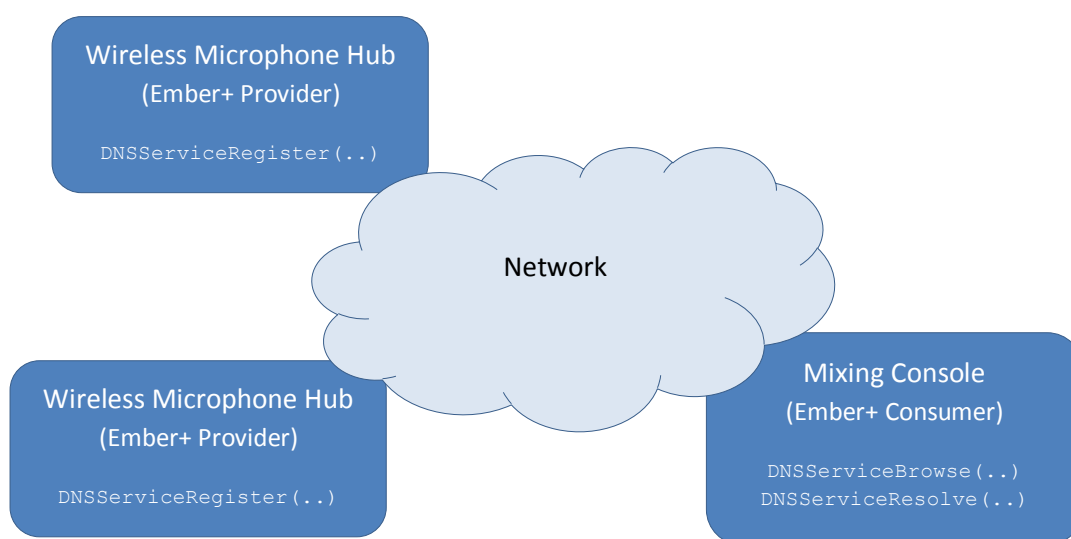


Image 1: Sample scenario with Ember+ devices. The API calls at the bottom are examples of the Bonjour SDK on Windows.

Zero Configuration Networking implementations

mDNSResponder is a multi-platform library which implements the service discovery part of Zero Configuration Networking. **Avahi** is another implementation for Linux based platforms. It implements much more than only service discovery and provides also a compatibility layer for the mDNSResponder library. This makes it easy to support both libraries.

A daemon process is usually running in the background, responding to queries sent by peers and issuing queries to gather information from others in the network. Multiple applications may then use the daemon to register its services or browse for services provided by others.

Note for Windows users:

It is sufficient to install the Bonjour SDK (see *Bonjour for Developers*) in order to install the Bonjour Windows service. It is **not** required to install for example iTunes!

References

- Zero Configuration Networking: The Definitive Guide, O'Reilly Media, ISBN-13: 978-0596101008
- DNS Service Discovery: <http://www.dns-sd.org/>
- Zero Configuration Networking on Wikipedia: http://en.wikipedia.org/wiki/Zero_configuration_networking
- Apple Bonjour on Wikipedia: [http://en.wikipedia.org/wiki/Bonjour_\(software\)](http://en.wikipedia.org/wiki/Bonjour_(software))
- Apple Open Source Portal: <https://developer.apple.com/opensource/>
- Bonjour for Developers: <https://developer.apple.com/bonjour/>
- mDNSResponder: <http://www.macosforge.org/>
- Avahi: <http://avahi.org/>