Windows Network Tools

Identifying the services on well-known ports and the process for name resolution can reveal a lot of information about the machines on the network. There are many tools we can use for Windows network awareness.

Tools	Description
SC	(Resource Kit) remotely creates and starts services from the command line.
net accounts	Updates user accounts and modifies password and logon requirements.
net computer	Adds or deletes computers from a domain.
net config	Displays configuration info for workstation or server services.
net continue	Reactivates a service that has been suspended by NET PAUSE.
net file	Closes a shared file and removes file locks.
net group	Adds, displays, or modifies global groups on servers.
net helpmsg	Displays info about network messages such as error, warning, and alert messages.
net localgroup	Modifies local groups on computers. net name adds or deletes an alias name for a computer.
net pause	Suspends a service or resource. Pausing a service puts it on hold.
net print	Displays print jobs and shared queues. For each queue, the display lists jobs, showing the size and status of each job, and the status of the queue.
net send	Sends messages to other users or computers. The Messenger service must be running to receive messages.
net session	Lists or disconnects sessions between the computer and other computers on the network.
net share	Makes resources available to network users. When used without options, it lists information about all resources being shared on the computer.
net start	Lists running services.
net statistics	Displays the statistics log for the local Workstation or Server service.
net stop	Stops services.
net time	Synchronizes the computer's clock with the domain.
net use	Connects a computer to a shared resource or disconnects a computer from a shared resource.
net user	Creates and modifies user accounts on computers. When used without switches, it lists the user accounts for the computer.
net view	Displays a list of resources being shared on a computer. When used without options, it displays a list of computers in the current domain or network.

Arp	Displays and modifies the IP-to-Physical address translation tables used byaddress resolution protocol (ARP).
Ipconfig	Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. Used without parameters, **ipconfig** displays the IP address, subnet mask, and default gateway for all adapters.
Ping	Verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) Echo Request messages. The receipt of corresponding Echo Reply messages are displayed, along with round-trip times. Ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution.
pathping	A route tracing tool that combines features of the **ping** and **tracert** commands with additional information that neither of those tools provides. The **pathping** command sends packets to each router on the way to a final destination over a period of time, and then computes results based on the packets returned from each hop.
Tracert	A route-tracing utility used to determine the path that an IP packet has taken to reach a destination.
Netstat	Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, **netstat** displays active TCP connections.
Nbtstat	Displays NetBIOS over TCP/IP (NetBT) protocol statistics, NetBIOS name tables for both the local computer and remote computers, and the NetBIOS name cache. **Nbtstat** allows a refresh of the NetBIOS name cache and the names registered with Windows Internet Name Service (WINS). Used without parameters, **nbtstat** displays help.
Browstat	Part of Winxp sp2 support tools. Intended as an administrative tool used to determine the root cause of a browsing issue and then help to fix it. We can use it to obtain a list of all computers in a non-domain environment.
Route	Displays and modifies the entries in the local IP routing table. Used without parameters, **route** displays help.
Nslookup	A command-line administrative tool for testing and troubleshooting DNS servers. This tool is installed along with the TCP/IP protocol through Control Panel. This article includes several tips for using Nslookup.exe.
Psexec	**(Sysinternals)** is a light-weight telnet-replacement that lets you execute processes on other systems, complete with full interactivity for console applications, without having to manually install client software. PsExec's most powerful uses include launching interactive command-prompts on remote systems and remote-enabling tools like IpConfig that otherwise do not have the ability to show information about remote systems.

Netsh	**(Article ID: 242468) -** is a tool an administrator can use to configure and monitor Windows-based computers at a command prompt. You can use the Netsh.exe tool to: configure interfaces, configure routing protocols, configure filters, configure routes, configure remote access behavior for Windows-based remote access routers that are running the Routing and Remote Access Server (RRAS) Service and to display the configuration of a currently running router on any computer. Use the scripting feature to run a collection of commands in batch mode against a specified router.
Ifids	Ifids is useful for looking at just one RPC, and it lets you talk to RPC servers that are not registered in the endpoint map.

Tasklist Commands

```
TASKLIST:  [/S system [/U username [/P [password]]]] [/M [module] | /SVC | /V] [/FI filter] [/FO format] [/NH]
```

Command	Description
/S system	Specifies the remote system to connect to.
/U [domain\]user	Specifies the user context under which the command should execute.
/P [password]	Specifies the password for the given user context. Prompts for input if omitted.
/M [module]	Lists all tasks currently using the given exe/dll name. If the module name is not specified all loaded modules are displayed.
/SVC	Displays services hosted in each process.
/V	Displays verbose task information.
/FI filter	Displays a set of tasks that match a given criteria specified by the filter.
/FO format	Specifies the output format. Valid values: "TABLE", "LIST", "CSV".
/NH	Specifies that the "Column Header" should not show in the output. Valid only for "TABLE" and "CSV" formats.

Listdlls Commands

```
listdlls [-r] [-v | -u] [processname|pid]
listdlls [-r] [-v] [-d dllname]
```

Command	Description
processname	Dump DLLs loaded by process (partial name accepted).
pid	Dump DLLs associated with the specified process id.
dllname	Show only processes that have loaded the specified DLL.
-r	Flag DLLs that relocated because they are not loaded at their base address.
-u	Only list unsigned DLLs.
-v	Show DLL version information.

Remote Procedure Call

Remote procedure call (RPC) is a network programming standard originally developed in the early 1980s. The Open Software Foundation (now The Open Group) made RPC part of the distributed computing environment (DCE) distributed computing standard. Although there is a second RPC standard, SunRPC, the Microsoft RPC implementation is compatible with the OSF/DCE standard. RPC builds on other networking APIs, such as named pipes or Winsock, to provide an alternate programming model that in some sense hides the details of networking programming from an application developer. Fundamentally, RPC provides a mechanism for creating programs that are distributed across a network, with portions of the application running transparently on one or more systems.

The main idea of an RPC is to allow a local computer (client) to remotely call procedures on a remote computer (server). It provides a common interface for communications between applications and serves as a go-between for client/server communications. It is designed to make client/server interaction easier and safer by factoring out common tasks such as security, synchronization, and data flow handling into a common library. It allows the client computer program to cause a subroutine or procedure to execute in another computer's (the server's) address space. Additionally, the programmer does not explicitly code the details for this remote interaction.

It is important to understand that RPCs are an application-layer communication mechanism. This means that RPCs use other network communication mechanisms, such as NetBIOS, named-pipes, or Windows Sockets, to establish the communication path.

RPC Components:

Client: A process that requests a service provided by another program. The client process uses the requested service without having to "deal" with many working details about the other program of service.

Server: A process that responds to requests from a client.

Endpoint: The name, port, or group of ports on a host system that is monitored by a server program for incoming client requests. The endpoint is a network-specific address of a server process for remote procedure calls. The name of the endpoint depends on the protocol sequence being used.

Endpoint Mapper (EPM):

The endpoint mapper service (commonly referred to as the portmapper service) creates a database of RPC services available on a system. As the RPC services initialize, the register with the endpoint mapper service to indicate they are up and running. The endpoint mapper can then be queried, either by the local computer or remote computers, for available RPC services.

Another purpose of the endpoint mapper service is to resolve RPC interface identifiers to transport endpoints. RPC interfaces serve as a network contact for calling a set of remote procedures. RPC services are called using the UUID (Universally Unique Identifier) assigned to its interface. The portmapper service is an RPC service that listens on different endpoints:

- **Ncalrpc:** epmapper LPC port
- **Ncacn_np:** epmapper named pipe
- **Ncacn_ip_tcp:** 135/tcp
- **Ncadg_ip_udp**: 135/udp
- **Ncacn_http:** 593/tcp

Typically, to discover the port on which an RPC service can be reached, a client will establish a TCP connection to port 135, asking for the port allocated to a given RPC service. Then, the client closes the connection to port 135 and opens a new connection to the port returned by the portmapper service.

The process by which an RPC connection is established is outlined in the steps below.

- When a computer's RPC services initialized, they register with the RPC endpoint mapper service. This allows the service to be made available to facilitate RPC connection requests from the local or a remote computer.
- The client connects to the endpoint mapper service and queries for a specific RPC service. If the service is not available, the client terminates the connection. If the service is available, the client sends a request to connect to the desired service.
- The server responds to the connection request with the protocols to be used for the connection to the RPC service.
- The client has the option to connect using one of the specified protocols, or it can attempt to connect using a different protocol. Generally, attempts to connection using other than the specified protocol returned by the server will fail. Otherwise the client connects using one of the specified protocols, and the RPC operation can continue.

RPC Operation:

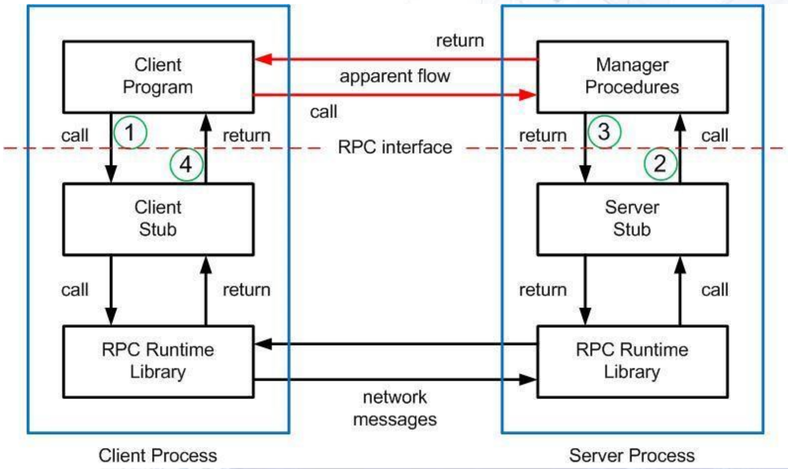
Client Stub: Module within a client application containing all of the functions necessary for the client to make remote procedure calls using the model of a traditional function call in a stand-alone application. The client stub is responsible for invoking the marshalling engine and some of the RPC application programming interfaces (APIs).

Server Stub: Module within a server application or service that contains all of the functions necessary for the server to handle remote requests using local procedure calls.

During an RPC connection, the RPC functions impersonate the credentials associated with the client process, to enable it run in the security context of the client process.

The figure below shows an example of a RPC operation and flow.

Figure: Remote Procedure Call Operation and Flow



1. The client application calls a local stub procedure instead of the actual code implementing the procedure. Stubs are compiled and linked with the client application. Instead of containing the actual code that implements the remote procedure, the client stub code:

   a. Retrieves the required parameters from the client address space.

   b. Translates the parameters as needed into a standard format for transmission over the network.

   c. Calls functions in the RPC client run-time library to send the request and its parameters to the server.

2. The server performs the following steps to call the remote procedure:

   a. The server RPC run-time library functions accept the request and calls the server stub procedure.

   b. The server stub retrieves the parameters from the network buffer and converts them from the network transmission format to the format the server needs.

   c. The server stub calls the actual procedure on the server.

3. The remote procedure then runs, possibly generating output parameters and a return value.

4. When the remote procedure is complete, a similar sequence of steps returns the data to the client.
    a. The remote procedure returns its data to the server stub.
    b. The server stub converts output parameters to the format required for transmission over the network and returns them to the RPC run-time library functions.
    c. The server RPC run-time library functions transmit the data over the network to the client computer
5. The client completes the process by accepting the data over the network and returning it to the calling function.
    a. The client RPC run-time library receives the remote-procedure return values and returns them to the client stub.
    b. The client stub converts the data to the format used by the client computer. The stub writes the data into the client memory and returns the results to the calling program on the client.
    c. The calling procedure continues as if the procedure had been called on the same (local) computer.

The run-time libraries are provided in two parts: the Import Library (which is linked to the application) and the RPC Run-time Library (which is implemented as a dynamic-link library (DLL)).

The server application contains the calls to the server run-time library functions which register the server's interface and allows the server to accept remote procedure calls. The server application also contains the application-specific remote procedures that are called by the client applications.

HOSTS and LMHOSTS

Understanding how the files, HOSTS and LMHOSTS, are implemented within Windows networking functionality is important. Take some time and read the recommended Internet sites.

Recommended Internet Sites

- **HOSTS and LMHOSTS File** https://web.archive.org/web/20160721111603/https://www.svrops.com/svrops/documents/hostsdoc.htm
- **LMHOSTS File** https://web.archive.org/web/20160721110826/https://technet.microsoft.com/en- us/library/cc977602.aspx
- **LMHOSTS File Information and Predefined Keywords** https://web.archive.org/web/20170601173503/http://montgomeryminds.com/blog/lmhosts-file-information-and-predefined-keywords/
- **What is RPC?** https://web.archive.org/web/20170608111144/https://technet.microsoft.com/en-us/en-%20us/library/cc787851(v=ws.10).aspx
- **DNS Name Resolution**

  https://web.archive.org/web/20160722144450/https://technet.microsoft.com/en-us/library/cc775637(v=ws.10).aspx

  https://web.archive.org/web/20170608111808/http://www.tcpipguide.com/free/t_DNSNameResolutionProcess-2.htm https://web.archive.org/web/20160721111845/http://www.itgeared.com/articles/1354-domain-name-system- dns-tutorial-overview/

https://web.archive.org/web/20160721111922/http://compnetworking.about.com/od/dns_domainnamesystem/f/what-is-a-dns-cache.htm

Please contact the Course Coordinators if you are unable to access any of the Recommended Internet Sites.