Проверка доступности TCP и UDP портов с помощью PortQryv2

http://winitpro.ru/index.php/2017/10/05/proverka-dostupnosti-tcp-i-udp-portov-s-pomoshhyu-portqryv2/

Утилита **Portqry.exe**, входящая в состав Support Tools для Windows 2003, является удобным инструментом проверки доступности TCP / UDP портов на удаленном сервере при диагностике проблем, связанных с функционированием различных сервисов, а также наличием файерволов и межсетевых экранов в TCP/IP сетях. Первая версия Portqry для Windows Server 2003 некорректно работает с более новыми ОС (Windows Server 2008 и выше), поэтому в дальнейшем была выпущена вторая версия утилиты **PortQryV2**. Именно эту версию и стоит использовать (скачать утилиту PortQryV2 можно по ссылке).

Первоначально утилита Portqry была исключительно консольным инструментом. К примеру, чтобы проверить доступность DNS сервера с клиента, необходимо проверить открыты ли на нем 53 порты TCP и UDP. Формат команды такой:

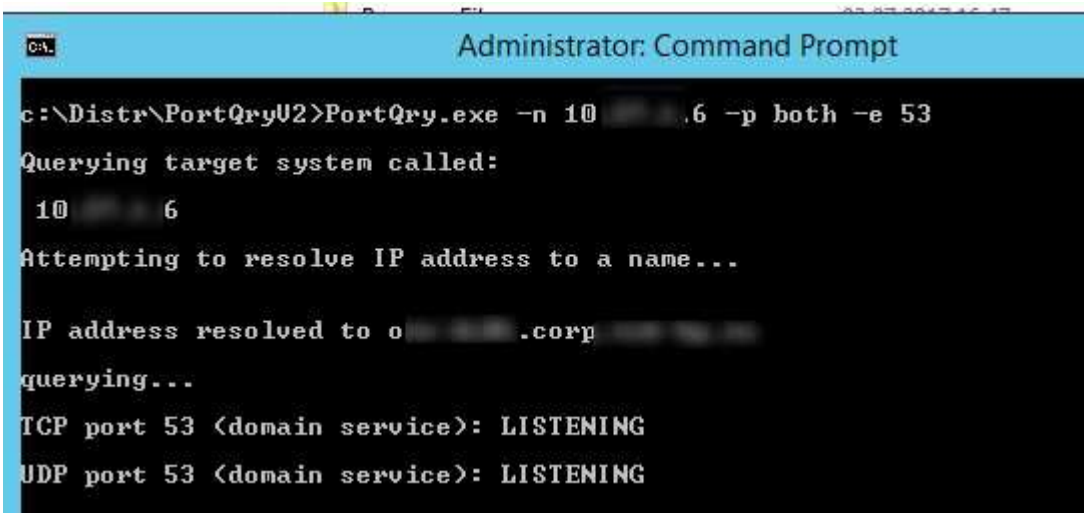
```
PortQry -n server [-p protocol] [-e || -r || -o endpoint(s)]
```

-n – имя или IP адрес сервера, доступ к которому нужно проверить
-e – порт для проверки (от 1 до 65535)
-r – диапазон портов для проверки (например, 1:80)
-p – по какому протоколу выполняется проверка. Это может быть TCP, UDP или BOTH (по умолчанию используется TCP).

Примечание. В отличии от комадлета PowerShell Test-NetConnection, который может использоваться только для проверки доступности TCP портов, утилита PortQry поддерживает и TCP и UDP протоколы.

В нашем примере команда будет такой:

```
PortQry.exe -n 10.1.10.6 -p both -e 53
```



Утилита Portqry для каждого указанного порта вернет один из трех статусов проверки его доступности

- **Listening** – означает, что порт открыт (принимает соединения)
- **Not Listening** – на целевой системе отсутствует процесс, который бы принимал подключения на указанном порту
- **Filtered** – утилита PortQry не получала ответа от указанного порта либо ответ был отфильтрован. Т.е. на целевой системе либо не слушается данный порт, либо доступ к нему ограничен файерволом или настройками системы.

В нашем примере, DNS сервер доступен с клиента и по TCP и по UDP.

```
TCP port 53 (domain service): LISTENING
UDP port 53 (domain service): LISTENING
```

С помощью атрибута **-o**, можно указать последовательность портов, доступность которых нужно проверить.

```
portqry –n 10.1.10.6  –p tcp –o 21,110,143
```

Следующий запрос выполнит сканирование диапазоны «известных» TCP портов и вернет список портов, которые принимают подключения (своеобразный TCP сканер открытых портов):

```
portqry –n 10.1.10.6  –r 1:1024 | find ": LISTENING"
```

В утилите PortQry имеется встроенная поддержка некоторых сетевых служб. Это LDAP, Remote Procedure Calls (RPC), почтовые протоколы SMTP, POP3 и IMAP4, SNMP, FTP/ TFTP, NetBIOS Name Service, L2TP и других. Помимо проверки доступности этих портов, утилита выполняет специфические для конкретного протокола запросы для получения статуса сервисов.

Например, с помощью следующего запроса мы не только проверим доступность службы RPC endpoint mapper (TCP/135), но и получим список имен зарегистрированных в системе конечных точек RPC (в том числе их имя, UUID, адрес к которому они привязаны и приложение, с которым они связаны).

```
portqry –n 10.1.10.6  –p tcp –e 135
```

TCP port 135 (epmap service): LISTENING
Using ephemeral source port
Querying Endpoint Mapper Database…
Server's response:
UUID: d95afe72-a6d5-4259-822e-2c84da1ddb0d
ncacn_ip_tcp:10.1.10.6 [49152]
UUID: 897e215f-93f3-4376-9c9c-fd2277495c27 Frs2 Service
ncacn_ip_tcp:10.1.10.6 [5722]
UUID: 6b5bd21e-528c-422c-af8c-a4079be4fe48 Remote Fw APIs
ncacn_ip_tcp:10.1.10.6 [63006]
UUID: 12345678-1234-abcd-ef22-0123456789ab IPSec Policy agent endpoint
ncacn_ip_tcp:10.1.10.6 [63006]
UUID: 367abb81-9844-35f1-ad32-91f038001003
ncacn_ip_tcp:10.1.10.6 [63002]
UUID: 50abc2a3-574d-40b3-1d66-ee4fd5fba076
ncacn_ip_tcp:10.1.10.6 [56020]
……..
UUID: 3c4428c5-f0ab-448b-bda1-6ce01eb0a6d5 DHCP Client LRPC Endpoint
ncacn_ip_tcp:10.1.10.6 [49153]
Total endpoints found: 61
==== End of RPC Endpoint Mapper query response ====
portqry.exe -n 10.1.10.6 -e 135 -p TCP exits with return code 0x00000000.

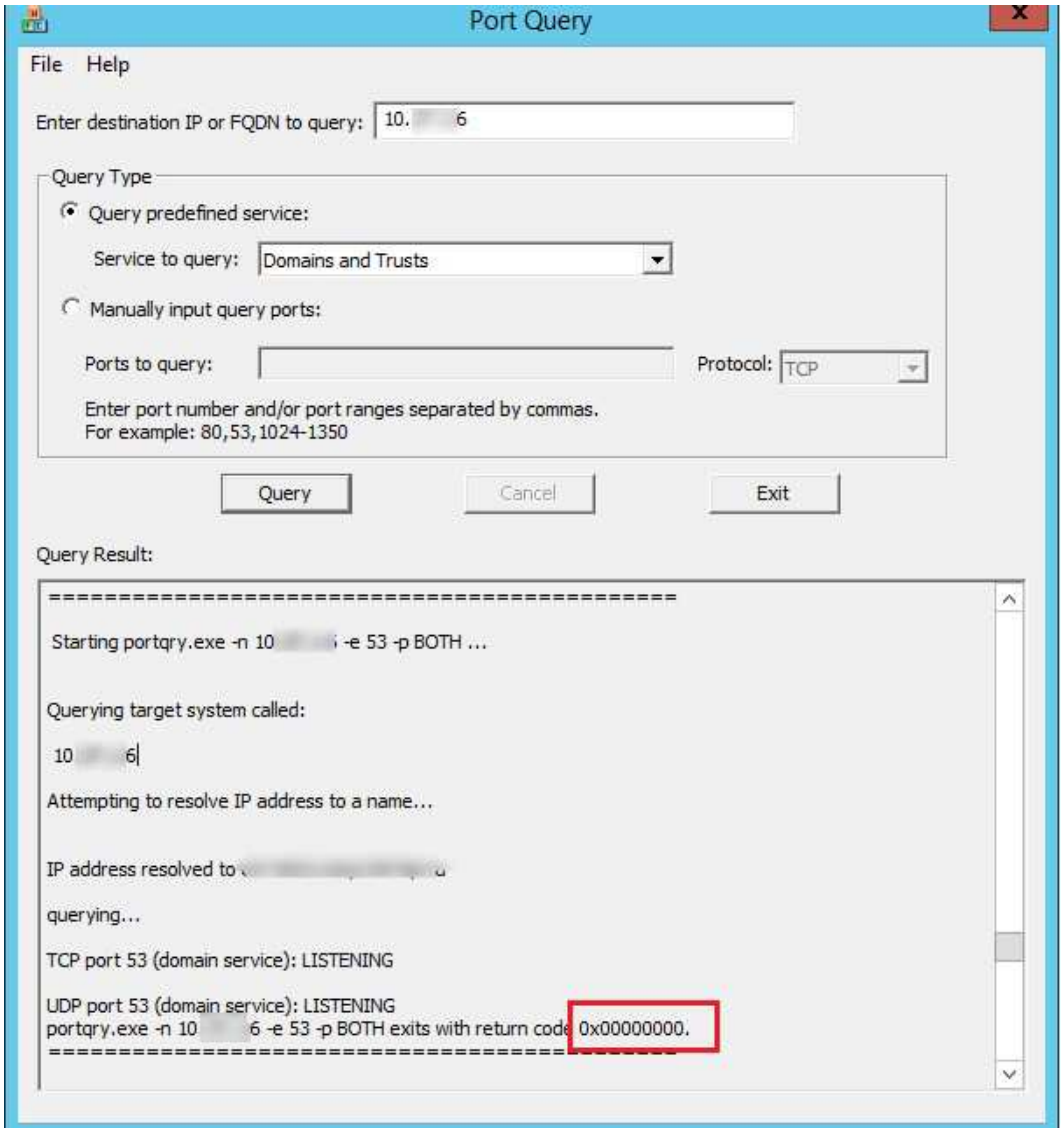
Для удобства пользователей, которые не дружат с командной строкой, MSFT разработало и простейший графический интерфейс для утилиты portqry – **PortQueryUI**. Скачать PortQueryUI можно с сайта загрузок Microsoft http://download.microsoft.com/download/3/f/4/3f4c6a54-65f0-4164-bdec-a3411ba24d3a/PortQryUI.exe

PortQueryUI по сути представляет собой графическую надстройку над portqry для формирования командной строки, и возврата результата графическое окно.

Кроме того, в PortQueryUI заложено несколько заранее предопределенных наборов запросов для проверки доступности популярных служб Microsoft:

- Domain and trusts
- IP Sec
- Networking
- SQL Server
- Web Server
- Exchange Server
- Net Meeting

Думаю, особых комментариев к интерфейсу PortQueryUI давать не нужно. Все должно быть понятно из скриншота ниже.



Стоит прокомментировать возможные коды возвратов в PortQueryUI (выделен на скриншоте).

- **0** – означает, что соединении успешно установлено и порт доступен
- **1** – указанный порт недоступен или отфильтрован
- **2** – это нормальный код возврата при проверке UDP подключения, т.к. не возвращается ACK ответ