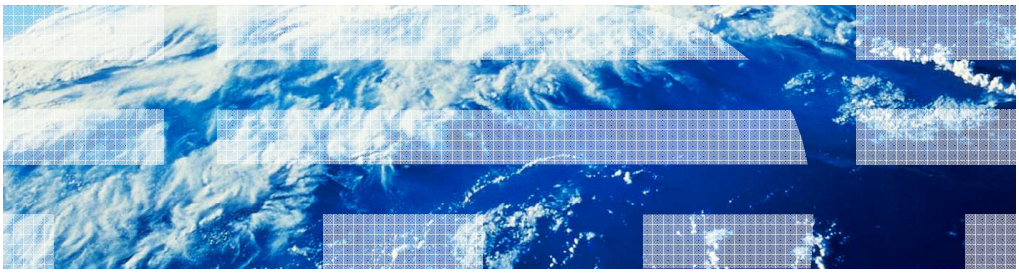
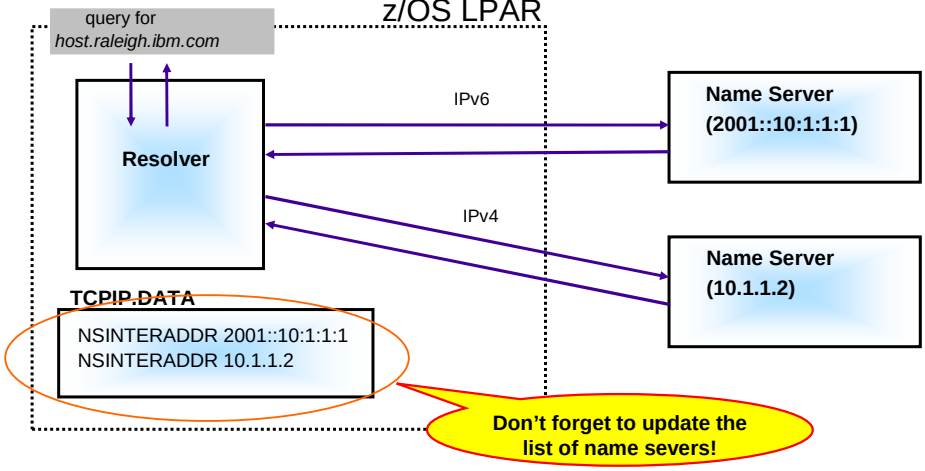
z/OS Communications Server Resolver enhancements



This presentation describes how the z/OS resolver can now send queries to DNS name servers over IPv6 connections in z/OS V1R12 Communications Server.

Resolver can communicate using IPv6

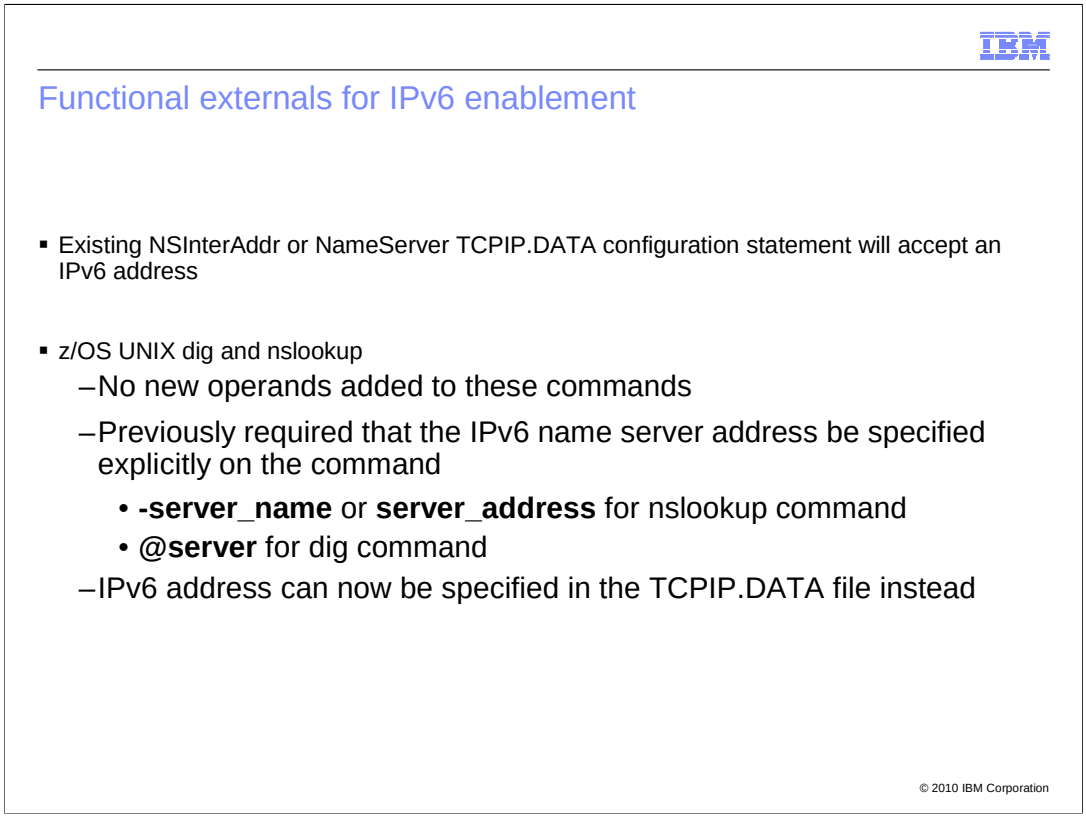
- Resolver can now communicate directly to a name server using IPv6
  - Eliminates one reason to define a local DNS name server



Starting in z/OS V1R12, the resolver is able to communicate to name servers using IPv6 connections. Before z/OS V1R12, in order to communicate with a DNS name server using IPv6, you were required to configure a local name server and have the resolver communicate with the local name server using IPv4.

In this diagram, you see that the local name server has been eliminated from the configuration, since the resolver can now directly forward the query to the IPv6 name server. If necessary, the resolver can still communicate using IPv4 connections to name servers, as shown with the second remote name server at IP address 10.1.1.2. IPv6 connections can be used to the local name server, if there were other reasons you had implemented a name server on the z/OS LPAR.

One subtle point to note is that when the local name server is removed, the TCPIP.DATA definitions of what name servers to query must be modified. Previously, the IPv4 loopback address (127.0.0.1) was the only IP address specified in the list of name servers. Now, with the local name server removed, two IP addresses are specified, representing the two name servers available to be queried.



You use the NSInterAddr, or the equivalent NameServer, configuration statement in TCPIP.DATA to define which name servers the resolver should contact. You continue to use this statement to define IPv6 addresses to be used for contacting a name server.

The existing limitations for these configuration statements exist. Only the first 16 name servers defined are used by the resolver, and no more than four name servers can be defined on a given statement. Any mix of IPv4 and IPv6 addresses can be specified as the list of 16 addresses, and any mix of IPv4 and IPv6 addresses can be coded on a given statement.

Trace resolver res_init( ) output

```
res_init Resolver values:
 Global Tcp/Ip Dataset  = SYS1.TCPPARMS(RESGLOBL)
 Default Tcp/Ip Dataset = TPOUSER.RESOLVER.DEFAULT.DATA
 Local Tcp/Ip Dataset   = USER55.TCPIP.DATA
 Translation Table      = Default
 UserId/JobName         = USER55
 Caller API             = TCP/IP Rexx Sockets
 Caller Mode            = EBCDIC
 System Name            = MVS000 (from VMCF)
 (L) DataSetPrefix = USER55
 (G) HostName = MVS026
 (L) TcpIpJobName = TCPCS
 (G) Search = tcp.raleigh.ibm.com
 (G) SortList = 0.0.19.0/0.0.255.0
 (G) NameServer = 2001::9:43:80:52
                  EDNS0 Support = Unknown
 (G) NsPortAddr = 53            (G) ResolverTimeout = 3
 (*) ResolveVia = UDP           (G) ResolverUdpRetries = 1
 (*) Options NDots = 1
 (*) SockNoTestStor
 (*) AlwaysWto = NO             (L) MessageCase = MIXED
 (*) LookUp = DNS LOCAL
 (*) Cache
 (G) LoadDbcsTable = BIG5
res_init Succeeded
```


You can examine the trace resolver res_init() output to determine the list of name servers to be searched. This list represents the full list of name servers, and not just the name servers in the base res_state control block.

An additional serviceability enhancement was added to trace resolver for res_init( ) processing. The system name in effect, and the source for that value, are now included in the list of information. The system name value can be coded on TCPIP.DATA configuration statements as a mechanism for having certain settings be active only on certain systems. The information is now included in the trace output to help identify situations where the incorrect setting was applied.

Trace resolver formatting changes

```
res_send Name Server Capabilities
 Name server 2001::9:43:80:52 EDNS0 Support = unknown
res_send Sending query to Name Server 2001::9:43:80:52
DNS Communication Started: 2009/06/18 15:51:51.816987
 BPX1SOC: RetVal = 0, RC = 0, Reason = 0x00000000, Type=IPv6
 No OPT RR record sent on request to 2001::9:43:80:52
 BPX1STO: RetVal = 38, RC = 0, Reason = 0x00000000
 BPX1AIO Sched: RetVal = 1, RC = 0, Reason = 0x00000000
 BPX1AIO RECVMSG :  From 2001::9:43:80:52
                    RetVal=502, RC=0, Reason=0x00000000
 UDP Data Length: 502
res_send received data via UDP
```


Additional trace resolver changes were made to accommodate the larger IPv6 addresses now possible. In this example, the trace entry for the BPX1AIO calls were converted into a two-line format, with the IP address on one line and the return code information on the second line.

There is one other change of note in trace resolver. When the resolver opens a socket, using the BPX1SOC interface, the trace entry now indicates if the socket opened was IPv6 or IPv4.

IPCS RESOLVER * DETAIL data

- The res_state IPv6 extension is included in the IPCS output

```
Resolver Process Data for USER33 Asid=0025 Tcb@=006FF560:
. . .
RPID_STATE NSADDR_LIST contains only IPv4 name server addresses
RPID_STATE_IPV6EXT STAT_NSADDR_LIST contains the full set of name server addresses
. . .
RRST: 7F6EB0D0
+0000 RETRANS.. 00000001 RETRY.... 00000002 OPTIONS.. 000002C1 NSCOUNT.. 00000002
       -- Array elements –
+0010 NSADDR_LIST........ 10020035 090B192E 00000000 00000000
+0020 NSADDR_LIST........ 10020035 090A192E 00000000 00000000
+0030 NSADDR_LIST........ 00000000 00000000 00000000 00000000
. . .
-- End of array --
. .
RRS6: 7F6EB520
+0000 STAT_NSCOUNT..... 00000004
       -- Array elements –
+0004 STAT_NSADDR_LIST. 1C130035 00000000 50C9C2D4 00000001 00090042 01050195 00000000
+0020 STAT_NSADDR_LIST. 10020035 090B192E 00000000 00000000 00000000 00000000 00000000
+003C STAT_NSADDR_LIST. 10020035 090A192E 00000000 00000000 00000000 00000000 00000000
+0058 STAT_NSADDR_LIST. 1C130035 00000000 00000000 00000000 00000000 00000001 00000000
+0074 STAT_NSADDR_LIST. 00000000 00000000 00000000 00000000 00000000 00000000 00000000
. . .
-- End of array --
```


The IPCS RESOLVER output was enhanced to show the contents of the IPv6 extension. The data is both displayed unformatted (not shown on the slide), but also formatted as shown here. In this example, a total of four IP addresses were defined to be searched. The first and fourth addresses are IPv6 sockaddr structures, while the other two are IPv4 sockaddr structures. The contents of the base res_state list of name servers shows only the two IPv4 addresses. As a reminder of this situation, text was added to the output to indicate the different content to expect in the two lists of name servers. This reminder is only included if IPv6 addresses are actually defined and appear in the IPv6 extension.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_Resolver.ppt

This module is also available in PDF format at: ../Resolver.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information