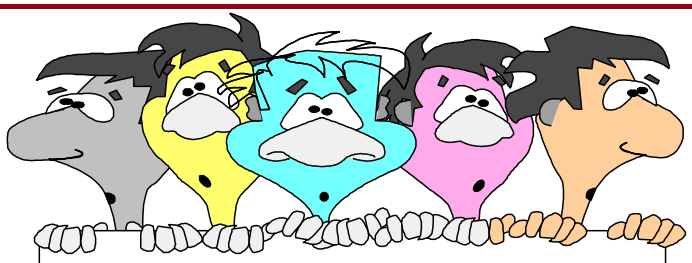
e-business

Communications Server z/OS® V1R5 and V1R6 Technical Update

OMPROUTE for z/OS V1R6

Topics

e-business



z/OS V1R6

- OSPFv3 - OSPF for IPv6

e-business

OMPROUTE Support for OSPFv3 (OSPF for IPv6) - z/OS V1R6

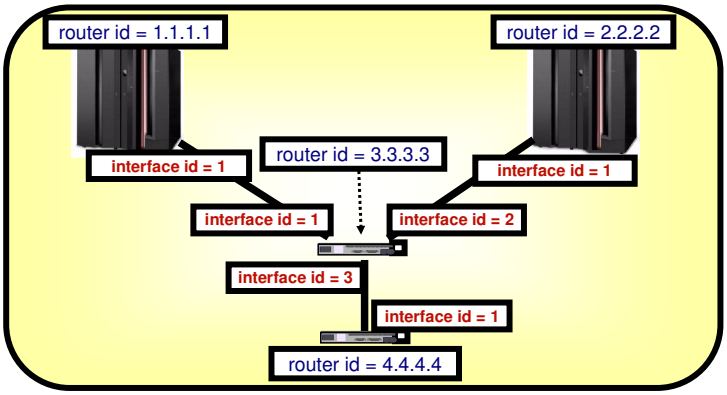
IBM

What is OSPF for IPv6?

- Extension of OSPF for IPv6
- Officially known as OSPFv3, though it's referred to as IPv6 OSPF in CS for z/OS publications
  - OSPF for IPv4 is officially known as OSPFv2
- Idea was to make it as protocol-independent as possible
  - could conceivably be used for architectures other than IPv6, though it isn't today
- IP addressing and topology semantics have been separated where possible
  - (many LSAs do not carry IP addresses at all, only abstract topology information)
- New LSA types added
  - (to carry addressing and link-local information)
- Concept of Flooding Scope added
  - (scopes are: link, area, autonomous system)
- Support for Unknown LSA types is added
  - makes the protocol more extensible
- Multiple OSPF instances supported on a link
- "Subnet" loses its importance, replaced by "Link"
  - since multiple IPv6 prefixes per link are allowed and expected, routing by subnet/prefix makes less sense

Differences: separation of IP addressing and topology

- In IPv6 OSPF, the topology elements are routers and links
  - Like IPv4, routers are identified by a router ID
    - Unlike OMPROUTE's IPv4 implementation, the router ID is not an IP address -- it's simply a abstract 32-bit number written in dotted-decimal format
  - Links are identified by a combination of owning router and interface ID
    - Interface ID is not an IP address -- it's a 32-bit integer.
    - Interface ID is an OSPF ID, it's not the same ID used by the stack to autoconfigure addresses on the interface
    - Only unique within a router, which is why both router AND interface ID is required for identification



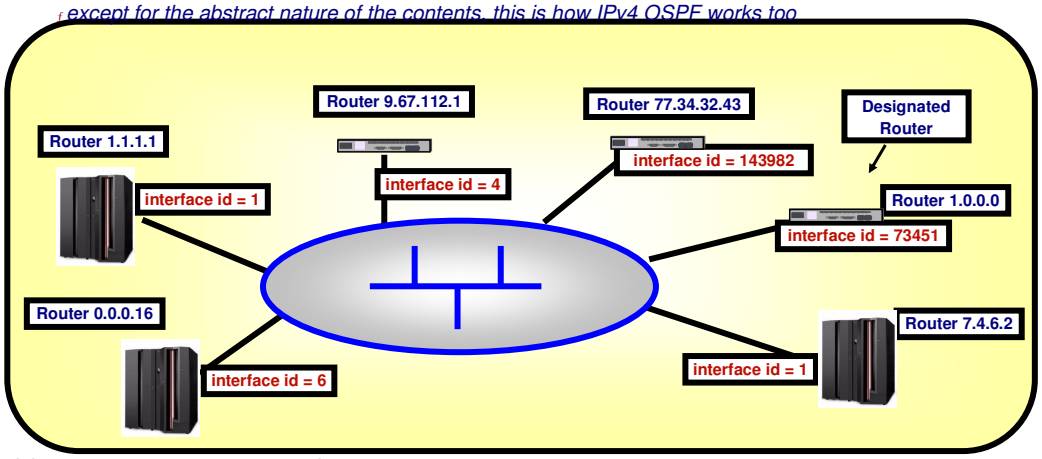
Notes on picture

NOTES

- The picture on the previous page illustrates the representation of topology in OSPFv3.
- For example, to get from router 1.1.1.1 to router 2.2.2.2, the topology path would be:
  - first hop to interface 1 on router 3.3.3.3
  - second hop to interface 1 on router 2.2.2.2
- Note that none of these numbers is an IPv6 address
- Router 3.3.3.3 would originate a router LSA that simply says:
  - connected to router 1.1.1.1's interface 1 via my interface 1
  - connected to router 2.2.2.2's interface 1 via my interface 2
  - connected to router 4.4.4.4's interface 1 via my interface 3
- Router 4.4.4.4's router LSA would read:
  - connected to router 3.3.3.3's interface 3 via my interface 1
- And so on....

How LANs are represented

- Like IPv4 OSPF, in IPv6 OSPF the designated router advertises a network LSA for a LAN.
  - *However it is expressed in abstract terms*
- Other routers on the LAN advertise their attachment to a LAN in their router LSAs
  - *except for the abstract nature of the contents, this is how IPv4 OSPF works too*



LAN picture notes

NOTES

- In the LAN pictured on the previous page, 1.0.0.0 is the designated router. That router would originate a network LSA for the LAN as follows:
  - This is for the Lan owned by router 1.0.0.0, interface ID 73451.
  - The following routers are attached to it:
    - 1.0.0.0
    - 1.1.1.1
    - 0.0.0.16
    - 9.67.112.1
    - 77.34.32.43
    - 7.4.6.2
- Each router would then advertise its attachment to the LAN by referring to the LAN's designated router and interface ID on the designated router. For example:
  - router 1.1.1.1 would include in its router LSA a link that says "My interface 1 is attached to the LAN owned by 1.0.0.0, which it refers to as interface ID 73451"
  - router 0.0.0.16 would include in its router LSA a link that says "My interface 6 is attached to the LAN owned by 1.0.0.0, which it refers to as interface ID 73451"
  - etc.
- Note that while some of the router IDs may look like IPv4 addresses, that is simply because the convention is to write them as dotted-decimal numbers. However it may be possible that the system adminstrator of one or more routers decides to use the same router ID for IPv4 and for IPv6 (this is allowed and supported by OMPROUTE).

Differences: New LSA types

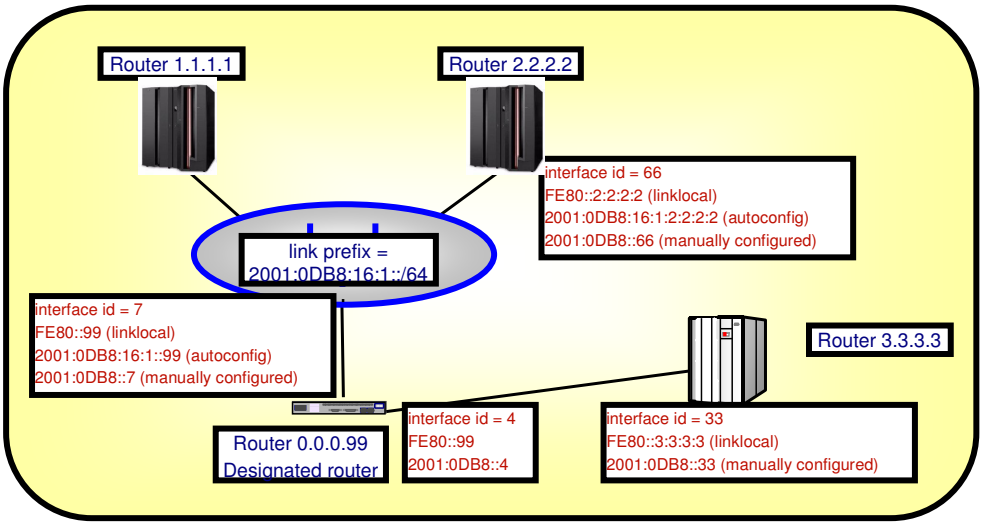
- Since the router and network LSAs no longer contain IP addressing information, new LSA types have been added to provide that function

- Intra-area prefix LSA
  - Used to let other routers in an area know what prefixes (or addresses) are associated with topology elements
    - an intra-area prefix LSA references a topology LSA (router or network) and then lists the prefixes or addresses associated with that topology element
    - multiple intra-area prefix LSAs can be originated for a single topology element, but an intra-area prefix LSA can only reference one topology element

- Link LSA
  - used to let other routers on the same link know a router's link-local address.
  - also used by routers to share the list of prefixes associated with a link and to assert options that they want the designated router to carry on the network LSA it originates for the network

Differences: New LSA types, example

- The following picture gives an example of how the new LSA types work. For readability only the addresses and identifiers needed to explain the example are shown here.



Notes on new LSA types example

NOTES

- Router 0.0.0.99 would advertise the following LSAs:
  - Network LSA for the network owned by 0.0.0.99 interface id 7:
    - attached routers: 0.0.0.99, 1.1.1.1, 2.2.2.2
  - Router LSA for router 0.0.0.99 has the following interfaces:
    - interface 7 attached to the network owned by 0.0.0.99 interface id 7
    - interface 4 attached point to point to 3.3.3.3's interface 33
  - Intra-area prefix LSA: the following prefixes are associated with the network owned by 0.0.0.99, interface id 7: 2001:0DB8:16:1::/64
  - Intra-area prefix LSA: the following prefixes and addresses are associated with router 0.0.0.99:
    - 2001:0db8::7/128, 2001:0db8::4/128, 2001:0db8:16:1::99/128
      - note link local addresses are NOT included here
  - Link LSA, sent out on the LAN and not forwarded to other links, with the following information:
    - router ID 0.0.0.99, my interface ID on this link is 7, my link-local address on this link is fe80::99, and the following prefixes are associated with this link: 2001:0db8:16:1::/64
  - Link LSA sent out on the point to point network and not forwarded to other links, with the following information
    - router ID 0.0.0.99, my interface ID on this link is 4, my link-local address on this link is fe80::99, and I have no prefixes to associate with this link
      - note that the link-local address only has to be unique on a link, so it is legal for a router to use the same link local address on multiple interfaces, as long as they are attached to different links

Notes on new LSA types example (continued)

NOTES

- Router 3.3.3.3 would advertise the following LSAs:
  - Router LSA, router 3.3.3.3 has the following links:
    - interface id 33 attached point to point to 0.0.0.99's interface 4
  - Intra-area prefix LSA indicating that the following IP addresses are associated with router 3.3.3.3:
    - 2001:0db8::33
  - Link LSA sent out on the point to point network and not forwarded to other links, with the following information
    - router ID 3.3.3.3, my interface ID on this link is 33, my link-local address on this link is fe80::33, and I have no prefixes to associate with this link

Differences: flooding scope

- IPv6 OSPF formalizes the concept of flooding scope, which is indicated using high order bits of the LSA type
  - link scope: the LSA is only flooded on one link
    - for example, link LSA
  - area scope: the LSA is only flooded within one OSPF area
    - for exampe, router, network, intra-area prefix, inter-area router, inter-area prefix
      - note: intra-area prefix LSA is the IPv6 equivalent of the IPv4 type-3 summary LSA
      - intra-area router is the IPv6 equivalent of the IPv4 type-4 summary LSA
  - autonomous system scope: the LSA is flooded throughout the autonomous system
    - for example, AS External LSA
- IPv4 OSPF already had this concept but:
  - there was no link scope
  - the concept wasn't formalized or indicated by bits within LSAs
    - the added advantage of the bit indicators is that unknown LSAs can be properly handled
      - more on unknown LSAs on the next page

Differences: handling of unknown LSAs

- In IPv4 OSPF, if an LSA has an unrecognized type, it's ignored
  - unknown LSAs discarded when received
- IPv6 OSPF requires support for unknown LSAs
  - they use a standard LSA header for fields like scope, type, checksum, sequence number, etc.
  - they should be stored, forwarded, and flooded like any other LSAs
  - the LSA scope bits (see previous page) tell a router how to handle an unknown LSA
    - if the LSA has link scope, do not forward it
    - if the LSA has area scope, only forward it out interfaces in the same area as it was received
    - if the LSA has autonomous system scope, forward it out all interfaces
- This function makes migration easier, as routers supporting different functional levels can coexist well
  - you can even have a designated router who is lower level function than some other routers on the network

Configuration defaults

- Configuration defaults for IPv6 OSPF were chosen to be identical to those for IPv4 OSPF wherever practical
  - hello (10), dead router (40), and database exchange (40) intervals
  - interface costs (1)
  - designated router priority (1)
  - AS boundary routing and comparision values
  - etc
- One big difference: the router ID
  - since the router ID in IPv6 OSPF is not an interface address, it cannot be defaulted to an interface's home address like in IPv4
  - if IPv4 OSPF is also running, the default is to use the same router ID value that IPv4 OSPF is using
    - otherwise this the IPv6 Router ID MUST be specified.

Interaction with router advertisement routes

- The CS for z/OS TCP/IP stack receives router advertisement routes from routers
  - CS for z/OS does not create router advertisement routes
- The stack informs OMPROUTE of received router advertisement routes
- OMPROUTE treats them identically to replaceable static routes
  - they are external to the OSPF autonomous system
  - their advertisement into the OSPF autonomous system is controlled by the IMPORT_ROUTER_ADVERTISEMENT_ROUTES setting on the IPv6_AS_BOUNDARY_ROUTING statement
  - if the prefix is learned by another means (e.g, link LSA), the router advertisment route is replaced
    - this implies that if a router is advertising a prefix on both router advertisements and link LSA, the link LSA advertisement will be used and the prefix will be considered internal to the OSPF AS
      - note: this is what CISCO routers do

Defining prefixes to OMPROUTE

- You can statically define prefixes to OMPROUTE by using the PREFIX parameter of the IPv6_OSPF_INTERFACE, IPv6_RIP_INTERFACE, or IPv6_INTERFACE statement
  - prefixes defined in this manner are not used for autoconfiguration
  - OMPROUTE will add direct routes to prefixes defined this way, and inform TCP/IP.
  - prefixes defined on IPv6_OSPF_INTERFACE statements are considered internal to OSPF
    - they will be advertised on the intra-area prefix LSA for that interface
    - they will also be advertised on the OSPF link LSA sent out the defining interface
  - prefixes defined on IPv6_RIP_INTERFACE statements are considered internal to RIP and external to OSPF
    - they will only be advertised to OSPF destinations if IPv6_AS_BOUNDARY_ROUTING IMPORT_RIP_ROUTES is enabled
  - prefixes defined on IPv6_INTERFACE statements are considered external to the OSPF autonomous system
    - they will only be advertised to OSPF destinations if IPv6_AS_BOUNDARY_ROUTING IMPORT_DIRECT_ROUTES is enabled

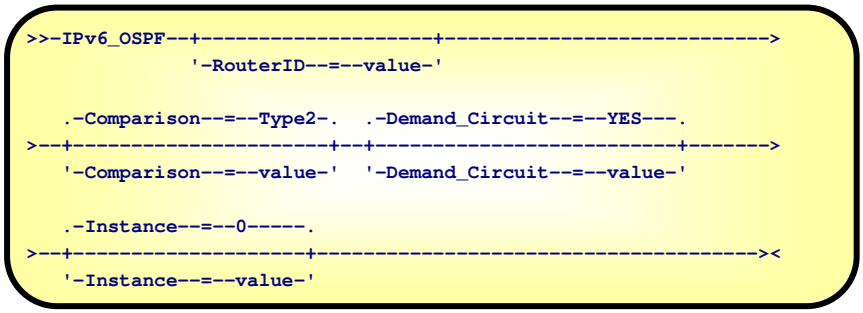
New OMPROUTE configuration statements

- New OMPROUTE configuration statements for IPv6 OSPF
  - IPv6_OSPF
  - IPv6_Area
  - IPv6_Range
  - IPv6_AS_Boundary_Routing
  - IPv6_OSPF_Interface
  - IPv6_Virtual_Link
- New OMPROUTE configuration statements for IPv4 OSPF
  - OSPF

IPv6_OSPF Syntax

NOTES

Used to specify various parameters that apply to the IPv6 OSPF autonomous system as a whole



Notes on IPv6_OSPF statement

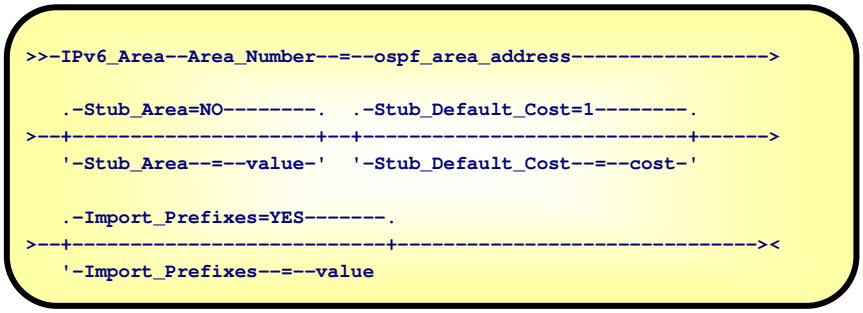
NOTES

- Router ID: a dotted-decimal 32-bit number.
  - If IPv4 OSPF is also running, defaults to the same router ID used by IPv4 OSPF
  - Must be specified if IPv4 OSPF is not also running.
- Comparison: Tells OMPROUTE where external routes fit in the IPv6 OSPF hierarchy. Similar in effect and values to the IPv4 OSPF **Comparison** statement
- Demand_Circuit: Enables or disables demand circuits for IPv6 OSPF. Similar in effect and values to the IPv4 OSPF **Demand_Circuit** statement
- Instance: Specifies the default instance value for all IPv6_OSPF_Interface statements. Values are 0-255.

IPv6_Area Syntax

NOTES

Used to define and set the parameters for an IPv6 OSPF area.



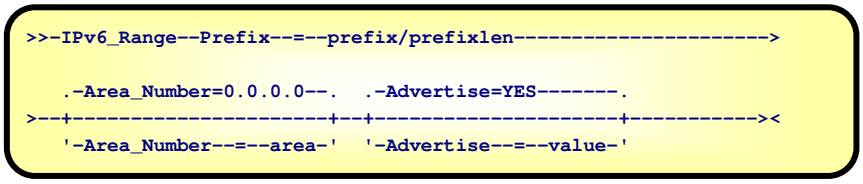
IPv6 Area syntax notes

- ospf_area_address: a 32-bit area number expressed in dotted-decimal notation (like IPv4 OSPF)
- Stub_Area: indicates whether or not this is a stub area. Works similarly to the same parameter on the Area configuration statement for IPv4
- Stub_Default_Cost: sets the cost of the default route this router advertises into the area if it's a stub area and this router is the area border router.
- Import_Prefixes: If this area is a stub area, indicates whether prefixes from neighboring areas will be imported A stub area with Import_Prefixes set to NO is commonly referred to in RFCs and other standards documentation as a Totally Stubby Area.

IPv6_Range Syntax

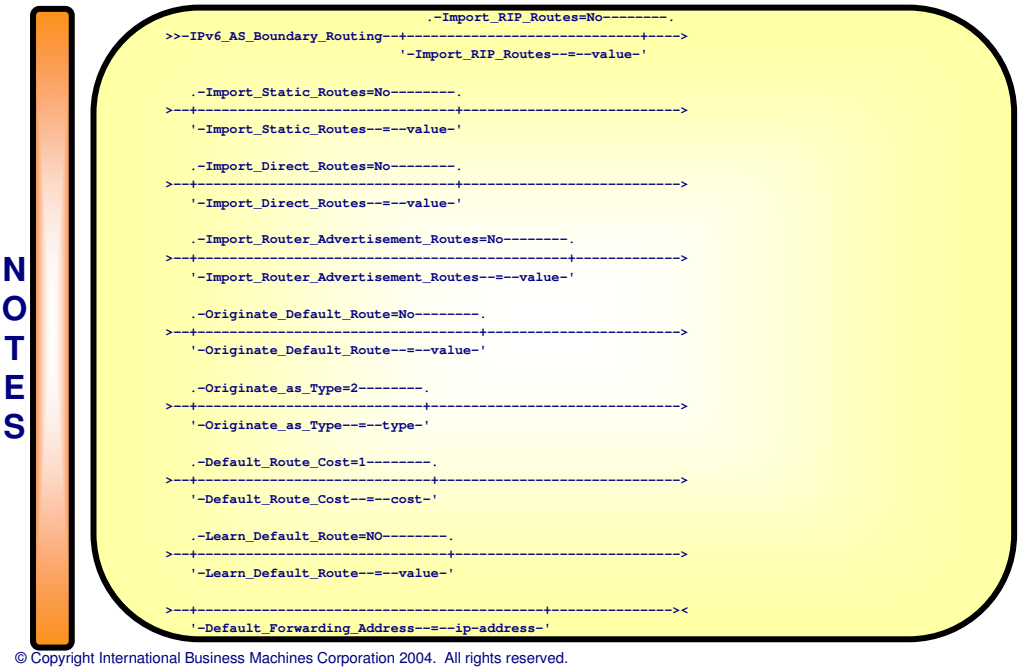
NOTES

Adds ranges to IPv6 OSPF areas. Similar in function to the Range statement for IPv4 OSPF



- prefix/prefix_len: Common prefix of IP addresses in this range
- Area_Number: Area number to which this range applies
- Advertise: Specifies whether this range will be advertised to other areas.
  - tip: this can be used to filter IP addresses between areas

IPv6_AS_Boundary_Routing Syntax



NOTES

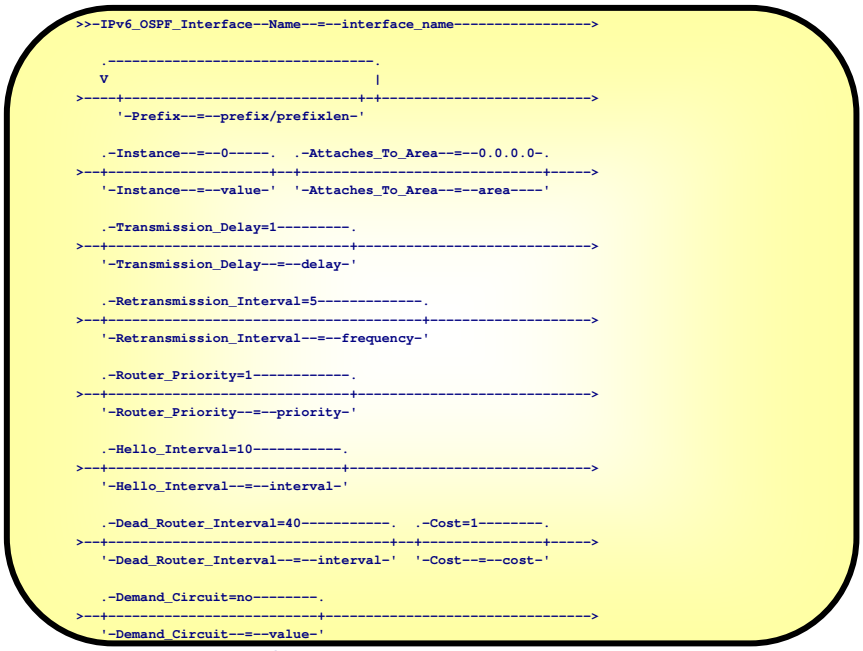
IPv6 AS_Boundary_Routing syntax notes

NOTES

- Import_RIP_Routes: Indicates whether or not routes from the IPv6 RIP autonomous system are to be advertised into IPv6 OSPF. Only applies if OMPROUTE is also running IPv6 RIP.
- Import_Static_Routes: Indicates whether or not IPv6 static routes are to be advertised into IPv6 OSPF
- Import_Direct_Routes: Indicates whether or not IPv6 direct routes are to be advertised into IPv6 OSPF
  - an example of an IPv6 direct route is a prefix defined on an IPv6_Interface statement
- Import_Router_Advertisement_Routes: Indicates whether or not IPv6 Router Advertisement routes are to be advertised into OSPF
  - note: if a router advertises a prefix on both router advertisements and the link LSA, it will be considered an OSPF prefix, not a router advertisement prefix.
- Originate_Default_Route: Specifies whether or not this host will originate an AS External default route into the IPv6 OSPF domain.
  - If YES and Default_Forwarding_Address is not also coded (or is coded to ::), this host will advertise itself as a default router
- Originate_as_Type: Specifies the external type assigned to the default route originated by this host if Originate_Default_Route is YES
- Default_Route_Cost: Specifies the cost that IPv6 OSPF associates with the default route originated by this host if Originate_Default_Route is YES
- Learn_Default_Route: Specifies whether IPv6 OSPF will learn default routes from inbound packets when their cost is higher than the cost of the default route originated by this host.
  - If this parameter is set to NO, then only default routes with lower cost than the one originated by this host will be learned.
- Default_Forwarding_Address: If Orignate_Default_Route is YES, this optional parameter may be used to specify that this host should originate a default route on behalf of a different router.
  - This parameter is not needed if this host is to advertise itself as the default router.
  - It should only be used when the default router is another router that this host can route to, which is not capable of advertising an IPv6 OSPF default route on its own behalf.

IPv6_OSPF_Interface Syntax

NOTES

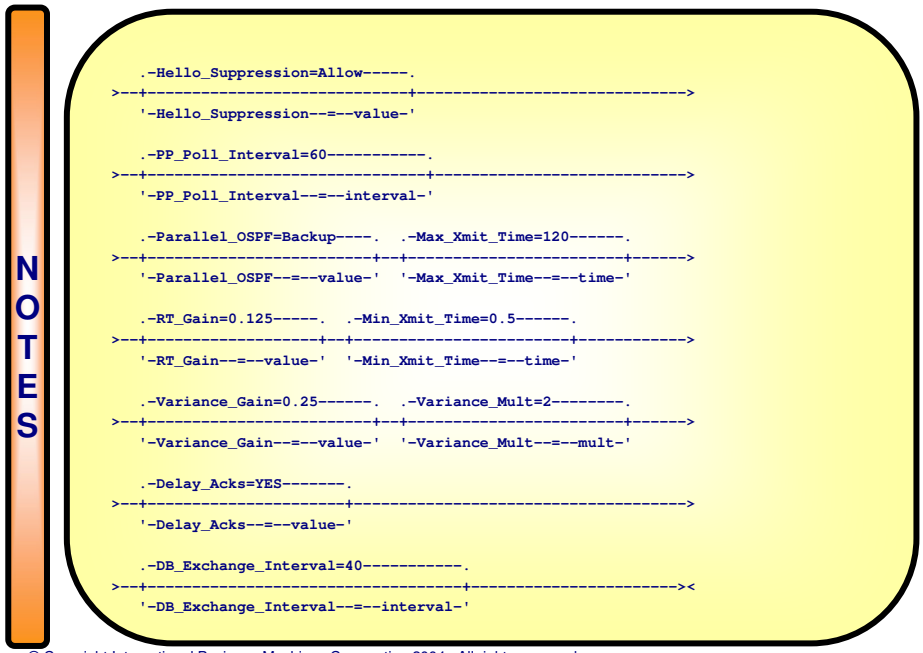


IPv6_OSPF_Interface

NOTES

- Name: the name of the interface. Must match the interface's name from the TCP/IP profile. Can be wildcarded in a manner similar to IPv6_RIP_Interface statements (i.e, trailing asterisk wildcards, for example VIPA* to match all interfaces whose name starts with the characters VIPA)
- Prefix: Specifies a prefix that is on the link to which the interface attaches. For each configured Prefix parameter, OMPROUTE will add a direct route to the defined prefix.
- Instance: Specifies the IPv6 OSPF protocol instance number for this interface. This value should be the same as the instance value of other IPv6 OSPF hosts or routers that OMPROUTE will be communicating with on this link
- Attaches_To_Area: IPv6 OSPF area to which this interface attaches
- Retransmission_Interval: Sets the frequency (in seconds) of retransmitting link-state update packets, link-state request packets, and database description packets
- Transmission_Delay: This parameter is an estimate of the number of seconds that it takes to transmit link-state information over the interface. As each link-state advertisement is sent out from this interface, it will be aged by this configured transmission delay.
- Router_Priority: This value is used for multiaccess networks to elect the designated router, with the highest priority router being elected. A value of 0 indicates that OMPROUTE cannot become designated router
  - tip: we recommend that OMPROUTE not be designated router if possible. If there are routers on the network, allow them to be designated router
- Hello_Interval: This parameter defines the number of seconds between IPv6 OSPF Hello packets being sent out this interface. This value must be the same for all routers attached to a common link.
- Dead_Router_Interval: The interval in seconds, after not having received an IPv6 OSPF Hello, that a neighbor is declared to be down.
- Cost: The IPv6 OSPF cost for this interface.
- Demand_Circuit: This parameter, when coded with YES, causes Link State Advertisements (LSAs) to not be periodically refreshed over this interface. Only LSAs with real changes will be advertised. In addition, coding this parameter to YES causes LSAs flooded over this interface to never age out

IPv6_OSPF_Interface Syntax *(continued)*



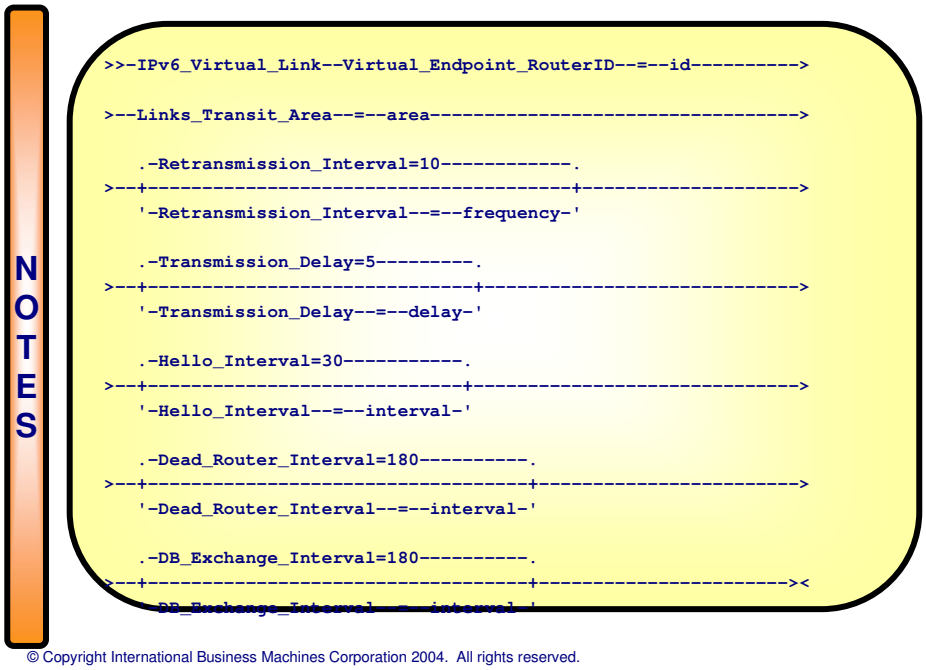
NOTES

IPv6_OSPF_Interface syntax notes, continued

NOTES

- Hello_Suppression: This parameter is meaningful only if Demand_Circuit is coded YES. This parameter allows you to configure the interface to request or allow Hello suppression.
  - only allowed with point to point or point to multipoint interfaces
- PP_Poll_Interval: This parameter specifies the interval (in seconds) that OMPROUTE should use when attempting to contact a neighbor to reestablish a neighbor relationship when the relationship has failed, but the interface is still available.
  - This parameter is meaningful only if Demand_Circuit is coded YES and Hello_Supression has been enabled
- Parallel_OSPF: This parameter designates whether the IPv6 OSPF interface is primary or backup when more than one IPv6 OSPF interface is defined to the same link.
- DB_Exchange_Interval: The interval in seconds that the database exchange process cannot exceed.
  - If the interval elapses, the database exchange process will be restarted.
- Retransmit Parameters: The following parameters are used by OMPROUTE to set values in the routes added to theTCP/IP route table, which use this interface. The values affect the TCP retransmit algorithms.
  - Max_Xmit_Time: Limits the TCP retransmission interval
  - Min_Xmit_Time: Sets a minimum retransmit interval.
  - RT_Gain: This value is the percentage of the latest Round Trip Time (RTT) to be applied to the smoothed RTT average.
    - The higher this value, the more influence the latest packet's RTT has on the average
  - Variance_Gain: This value is the percentage of the latest RTT variance from the RTT average to be applied to the RTT variance average.
    - The higher this value, the more influence the latest packet's RTT has on the variance average.
  - Variance_Mult: This value is multiplied against the RTT variance in calculating the retransmission interval.
  - Delay_Acks: The delay acknowledgments value to add to the routing table for routes that take this interface.

IPv6_Virtual_Link Syntax



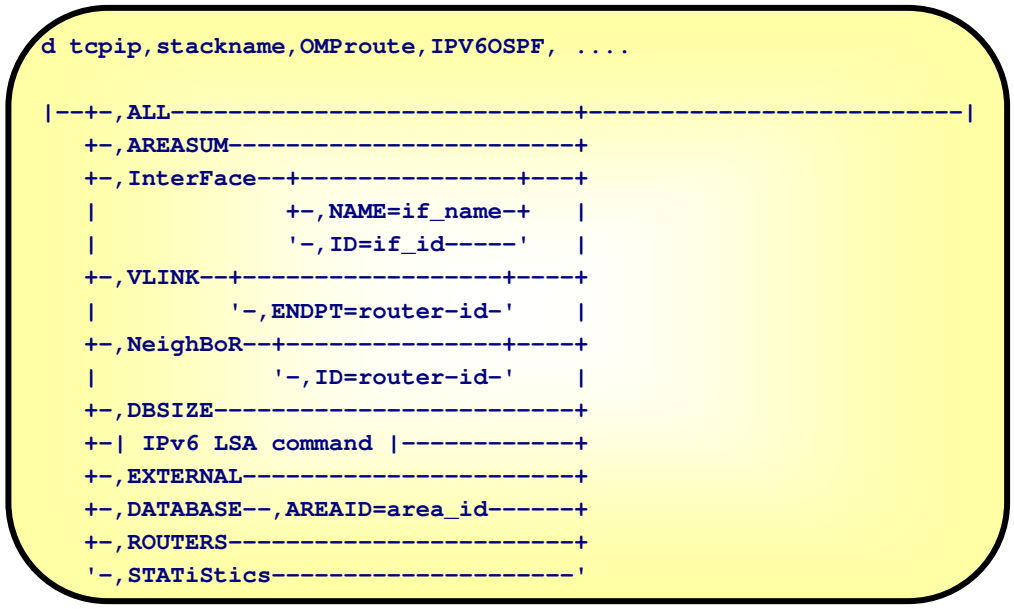
N O T E S

IPv6_Virtual_Link syntax notes

NOTES

- Virtual_Endpoint_RouterID: 32-bit IPv6 OSPF router ID of the virtual neighbor (other endpoint), specified in dotted-decimal notation.
- Links_Transit_Area: This is the nonbackbone, nonstub area through which the virtual link is configured.
- Retransmission_Interval: Sets the frequency (in seconds) of retransmitting link-state update packets, link-state request packets, and database description packets
- Transmission_Delay: This parameter is an estimate of the number of seconds that it takes to transmit link-state information over the virtual link. As each link-state advertisement is sent out over the virtual link, it will be aged by this configured transmission delay.
- Hello_Interval: This parameter defines the number of seconds between IPv6 OSPF Hello packets being sent out this virtual link. This value must be the same for both endpoint routers of the virtual link.
- Dead_Router_Interval: The interval in seconds, after not having received an IPv6 OSPF Hello, that the other endpoint router is declared to be down.
- DB_Exchange_Interval: The interval in seconds that the database exchange process cannot exceed.
  - If the interval elapses, the database exchange process will be restarted.

d tcpip,,omproute,IPV6OSPF command syntax



d tcpip,,omproute,IPV6OSPF,all command syntax sample output


```
EZZ7958I IPV6 OSPF INTERFACES
NAME             AREA            TYPE    STATE COST HELLO DEAD NBRS ADJS
VIPA1A6          6.6.6.6         VIPA     N/A     1  N/A   N/A  N/A  N/A
MPCPTP7TO5       0.0.0.0         P-2-MP    16     1   10    40    1    1
NSQDIO1L6        6.6.6.6         BRDCST    32     1   10    40    3    2
VL/0             0.0.0.0         VLINK     16     1   30   180    1    1

EZZ7972I IPV6 OSPF VIRTUAL LINKS
ENDPOINT         TRANSIT AREA    STATE COST HELLO DEAD NBRS ADJS
64.64.64.64      6.6.6.6           16     1   30   180    1    1

EZZ8129I IPV6 OSPF NEIGHBORS
ROUTER ID        STATE LSRXL DBSUM LSREQ HSUP RTR-PRI IFC
65.65.65.65        128     0     0     0  OFF        1 MPCPTP7TO5
64.64.64.64        128     0     0     0  OFF        1 NSQDIO1L6
63.63.63.63        128     0     0     0  OFF        1 NSQDIO1L6
68.68.68.68        128     0     0     0  OFF        1 NSQDIO1L6
64.64.64.64        128     0     0     0  OFF        1 *
```

Other new OMPROUTE function: OSPF configuration statement

- For IPv6 OSPF, miscellaneous parameters were consolidated under the IPV6_OSPF configuration statement
- We decided to implement this design for IPv4 OSPF also

```
                                  .-Comparison--=--Type2-.
>>-OSPF--+--------------------+--+----------------------+------->
         '-RouterID--=--value-'  '-Comparison--=--value-'

   .-Demand_Circuit--=--YES---.
>--+--------------------------+-------------------------------><
   '-Demand_Circuit--=--value-'
```

- The parameters on this new statement behave exactly the same as their standalone equivalents
  - *the older, standalone statements are still supported but no new ones will be added -- the OSPF statement will receive such new enhancements.*

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.