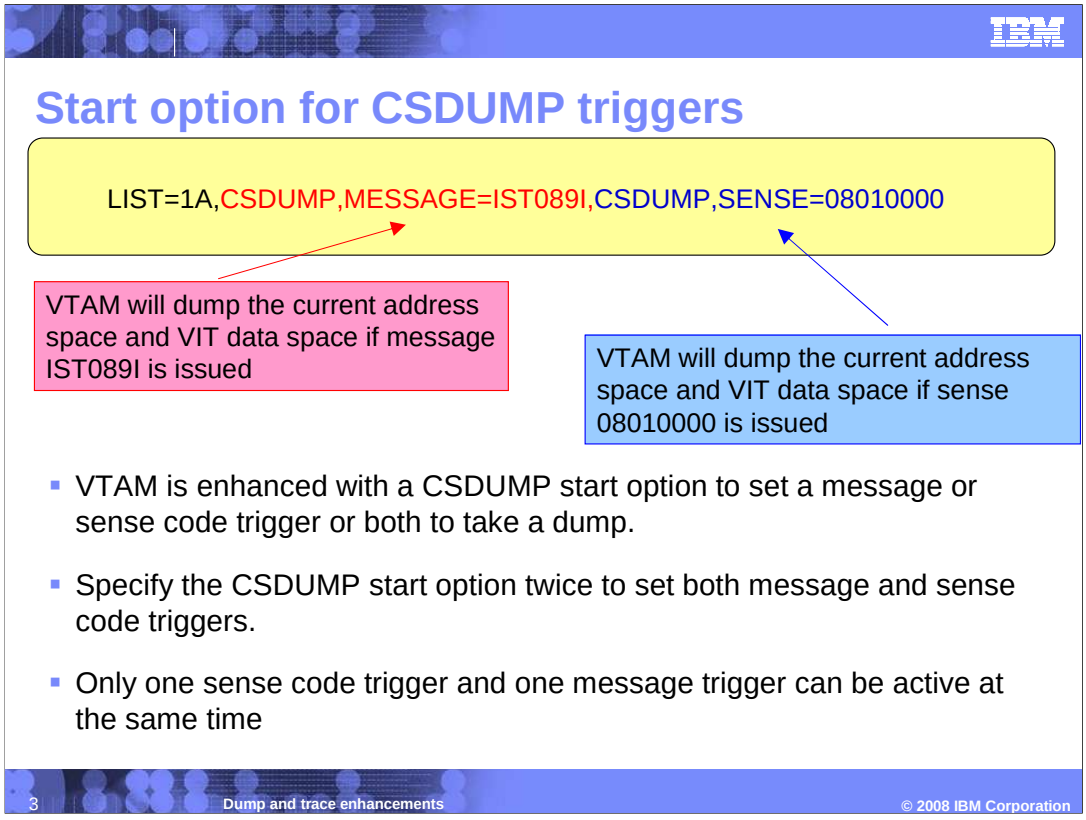
This presentation will discuss dump and trace enhancements added to z/OS V1R10 Communications Server

Start option for CSDUMP triggers - Background

- Modify CSDUMP command sets the message or sense code CSDUMP triggers to take a dump.
- VTAM® does not have any start option to set the message or sense code CSDUMP triggers to take a dump.
- User must re-issue Modify CSDUMP command to set the message or sense code CSDUMP triggers if VTAM is started again. It is prone to error.
- Problems can occur before the operator can enter a Modify CSDUMP command to set the trigger.

© 2008 IBM Corporation

V1R9 and earlier z/OS Communications Server has a current CSDUMP function. One limitation with the current CSDUMP implementation is that the triggers can only be applied by issuing VTAM's MODIFY CSDUMP command. For a problem that takes a long time to re-occur, this means that you must re-issue the command after any VTAM restart until the problem finally hits. If you did not re-issue the command after a VTAM restart for any reason, the problem can occur and you will not get any documentation for the problem.



A CSDUMP start option is added to make it easier to set the CSDUMP triggers.

In the CSDUMP start option, the CSDUMP sense code trigger has the syntax to set sense code trigger, and the CSDUMP message trigger has the syntax to set the message trigger. The syntax is similar to Modify CSDUMP command, including support for the MATCHLIM parameter. It is documented in the SNA Resource Definition manual under CSDUMP start option.

VTAM allows only one sense code trigger and only one message trigger to be active at the same time.

The CSDUMP start option can be added on the start command or the start option list as required. See z/OS V1R10 SNA Resource Definition for the syntax for both the start command and the start option list.

This slide shows an example of start option with both a sense code and a message CSDUMP trigger.

Start option for CSDUMP triggers: Display example

- Display CSDUMP displays CSDUMP triggers set by CSDUMP start option or MODIFY CSDUMP command.
- A MODIFY CSDUMP command for a given trigger will replace the same type of trigger entered as a start option.
- DISPLAY VTAMOPTS cannot be used to display CSDUMP

```
d net,csdump
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = CSDUMP TRIGGERS
IST1871I MESSAGE TRIGGER: MESSAGE = IST089I MATCHLIM = 1
IST1873I SENSE TRIGGER: SENSE = 08010000 RU = *ANY* MATCHLIM = 1
IST314I END
```


© 2008 IBM Corporation

Issue the Display CSDUMP command to verify the CSDUMP triggers set by either a CSDUMP start option or a MODIFY CSDUMP command. This command displays the current CSDUMP triggers that are active. If a CSDUMP trigger is set by the CSDUMP start option and the same trigger type is set by a MODIFY CSDUMP command, the trigger set by the MODIFY is the active trigger.

The DISPLAY VTAMOPTS cannot be used to display the CSDUMP start option values.

This slide shows an example of output from the CSDUMP display command.

Tracing enhancements - Data trace filtering

- A new *PORTNUM* filtering option has been added to the data trace
  - Allows data tracing to be filtered based upon a socket's source or destination port number
- This allows data capture to be limited to a single connection, even when there are multiple connections for a single application or between hosts
- Note that PORTNUM applies only to the TCP and UDP protocols
  - When PORTNUM is specified, packets using the RAW protocol are not traced

© 2008 IBM Corporation

The new PORTNUM keyword on the DATTRACE command will provide an additional level of filtering. This allows data capture to be limited to a single connection, even when there are multiple connections for a single application or between hosts. Note the port number applies to connected sockets.

Some UDP socket calls allow sending of a packet without a bind or connect function call. On these calls the port number is not saved in the required control blocks.

Also, since RAW protocols do not use ports, using this option will prevent tracing of RAW protocol data.

Tracing enhancements - Data trace collection

- The data trace has been enhanced to collect the domain, driver type, and protocol information
- For each record, the formatted data trace will output the these new fields:
  - The *Domain* field is set to either *AF_INET* or *AF_INET6*
  - The *Type* field is set to either *Stream, Datagram,* or *Raw*
  - The *Protocol* field is set to either *TCP, UDP,* or the protocol name

© 2008 IBM Corporation

Additional information is now collected by the TCP/IP data trace. Now collected is the domain (IPv4 or IPv6), the driver type (stream or datagram) and the protocol. These are arguments of the socket() function and are now captured with each data trace record. These new fields will also be formatted by the data trace formatter.

Tracing enhancements - Tracing discarded packets

- The PKTTRACE command has a new parameter to select packets by discard code.
  - `PKTTRACE DISCARD=NONE|nnnnn|*|ALL`
- The IPCS CTRACE subcommand for SYSTCPDA, SYSTCPOT, and SYSTCPIS can now select packets by discard code.
  - `OPTIONS(( DISCARD(nnnnn) ))`
  - `OPTIONS(( FLAG(DISCARD) ))`
- The EZBCTAPI Network Management Interface to SYSTCPDA formatting can used to select packets by discard code.
  - The EZBYCODE macro file describes the discard reason code values

© 2008 IBM Corporation

The OSA Express Network Traffic Analysis (OSAENTA) function allows customers to request that packets that are ordinarily silently discarded to be traced. With V1R10 the same function is added to the IP Communications Server. Packets that would otherwise be silently discarded by IP or TCP processing can now be traced with the PKTTRACE command.

Discarded packets can now be traced in several trace types. They can be traced in SYSTCPDA, which is the TCP/IP packet trace, SYSTCPOT, which is the OSA packet trace, and SYSTCPIS, which is the Intrusion Detection System Trace.

You can select all discarded packets, or packets which are discarded for a specific reason.

There are several IP reasons for discarding an example, here are some examples.

The IP header is malformed. For example, the length of header exceeds the size of the data received.

The IPv4 check sum value was incorrect.

An source or destination IP address that is not valid

The device interface was not active

There are several TCP reasons for discarding a packet, and here are some examples.

The TCP header is malformed.

The TCP check sum value was incorrect.

A duplicate SYN packet was discarded.

The PKTTRACE discard function involved changing the PKTTRACE command to add a parameter for the selection of discarded packets. NETSTAT was changed to display the new parameter on the PKTTRACE command (NETSTAT DEVLINKS). The SYSTCPDA packet trace records now contain a new field for the discard code which can be used the IPCS CTRACE command to select, format and count packets that have been discarded. This functionally was also added to the EZBCTAPI packet trace formatting Network Management Interface (NMI).

Tracing discarded packets - Limitations

- Only inbound packets that are discarded are traced. There are no outbound discarded packets.
- Only packets that are discarded at the IF, IP and TCP layers are traced.

© 2008 IBM Corporation

Only inbound packets that are discarded are traced. There are no outbound discarded packets.

Inbound packets discarded by the UDP and RAW layers are not traced.

To collect all discard packets the this command can be used:

VARY PKTTRACE,ON,DISCARD=ALL

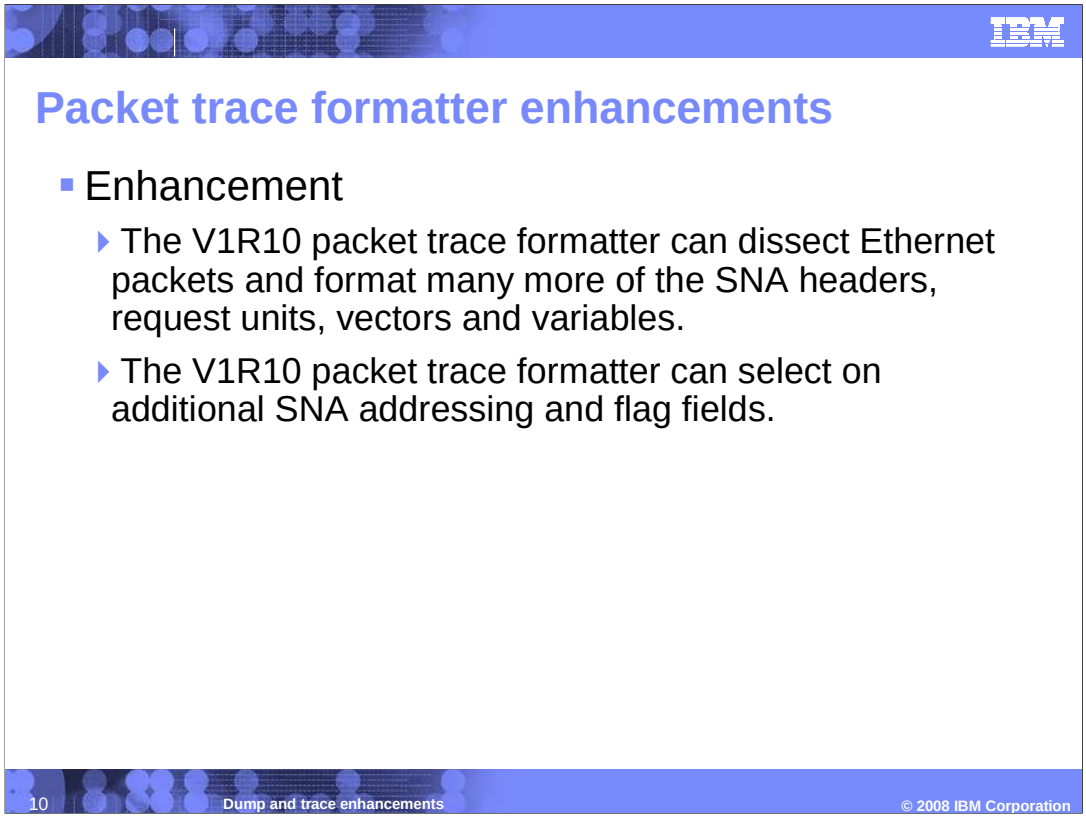
Tracing enhancements- Packet trace formatter

- Background Information:
  - In z/OS V1R6 formatting of Enterprise Extender (EE) packets was introduced. At that time only the HPR header data was formatted.
  - In z/OS V1R9 the OSA Express Network Traffic Analysis (OSAENTA) can capture true SNA packets from Communication Controller for Linux® (CCL).

© 2008 IBM Corporation

In z/OS V1R6 formatting of Enterprise Extender (EE) packets was introduced. At that time only the HPR header data was formatted. In z/OS V1R9 the OSA Express Network Traffic Analysis (OSAENTA) can capture true SNA packets from Communication Controller for Linux (CCL).

However, the V1R9 packet trace formatter does not format all of the SNA headers, Request Units, Control Vectors and GDS variables.



The z/OS V1R10 Communications Server includes enhancements to EE packet formatting. The packet trace formatter can dissect Ethernet packets and format many more of the SNA headers, request units, vectors and variables.

The packet trace formatter can select on additional SNA addressing and flag fields. These enhancements are available to the SYSTCPIP, SYSTCPIS and SYSTCPOT component traces. When using the NMI EZBCTAPI formatting macro, the packets will also reflect the additional SNA formatting.

IPCS formatter enhancements: Queued application data

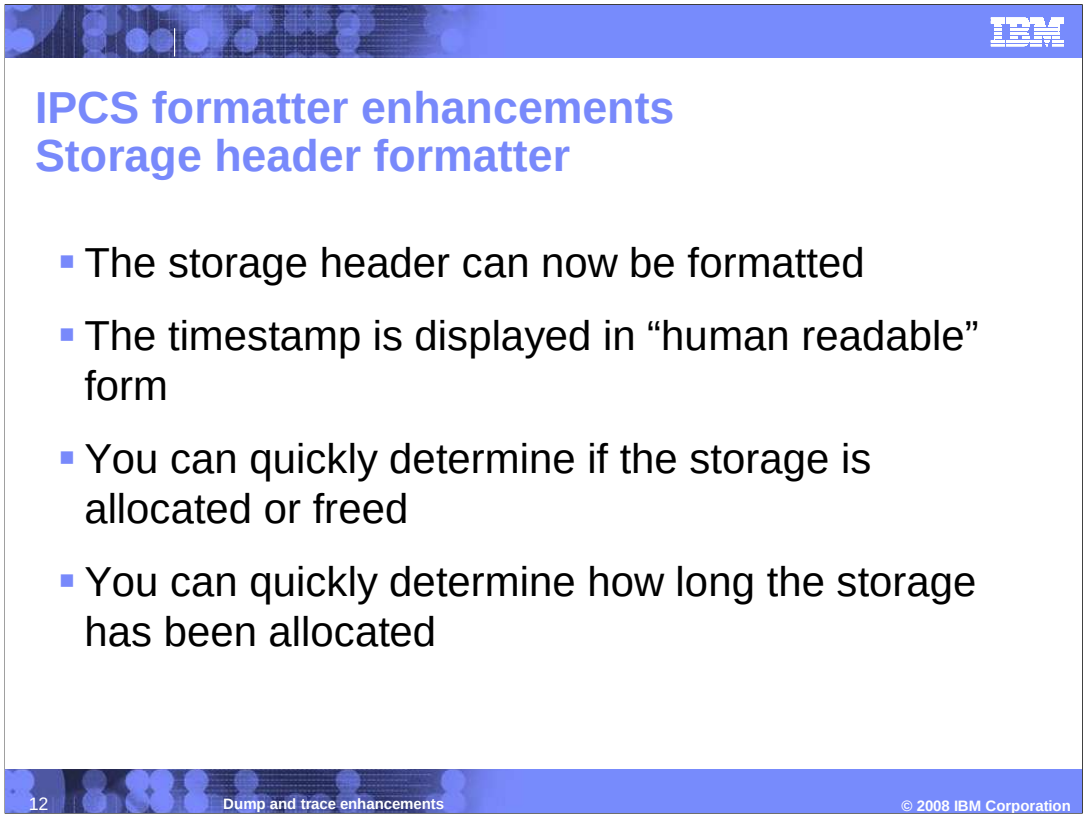
- A new operand, DATAQ, is added to TCPIPCS TCB, UDP, and RAW. This will format only the connection control blocks that have queued data.
  - TCPIPCS TCB(* DATAQ {DETAIL})
  - TCPIPCS UDP(* DATAQ {DETAIL})
  - TCPIPCS RAW(* DATAQ {DETAIL})
- If the DETAIL keyword is specified, DETAIL formatting is provided, but only for connection control blocks with queued data.

© 2008 IBM Corporation

Applications that do not RECEIVE their data in a timely manner can cause storage problems. Determining which applications are causing the problem can be difficult. For TCBs, data can also be queued to the SEND queue. Data that is not acknowledged can remain queued to the TCB SEND queue.

In V1R9 and earlier, the only way to find applications that are waiting on data is to run TCB or UDP or RAW or a combination and find every connection control block with a non-zero EVENT_QUEUE field. You then manually analyze the EVNT control block to determine what the application is waiting on. Also, the chain of EVNT control blocks has to be followed to determine the total number of EVENTs that the application is waiting on.

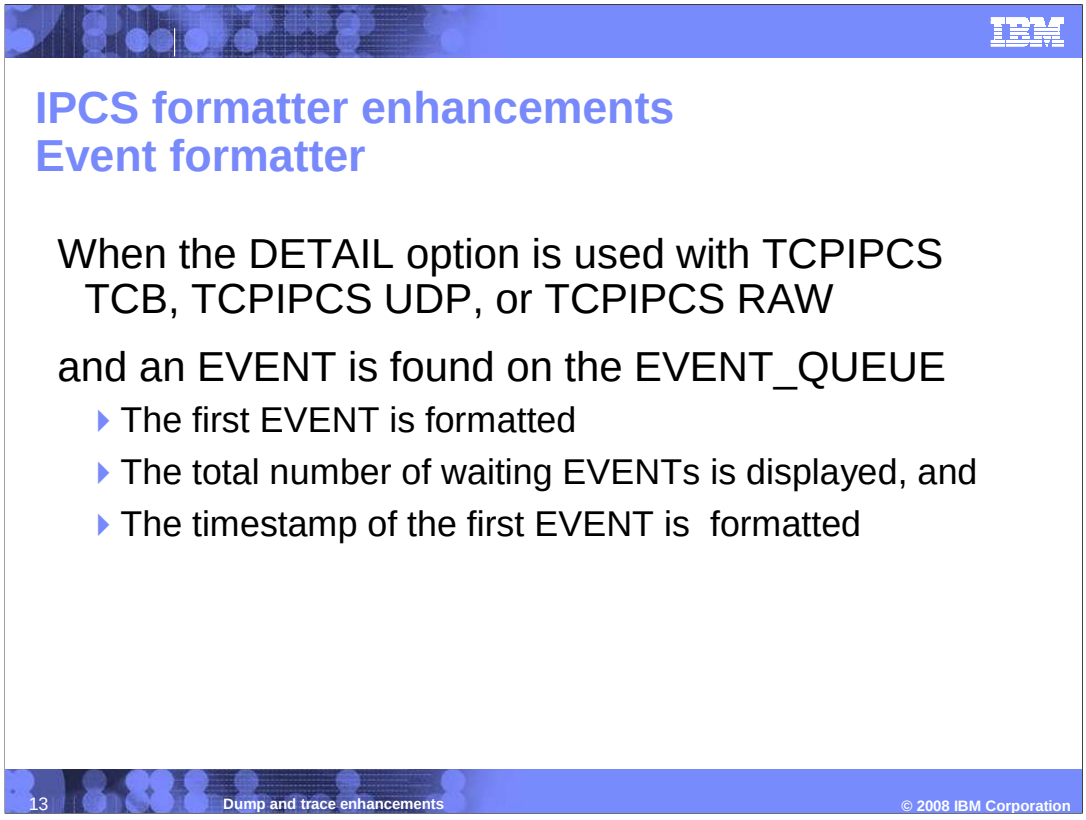
In V1R10 a new operand, DATAQ, is added to TCB, UDP, and RAW. When DATAQ is specified, only the connection control blocks with queued data is formatted and displayed. DATAQ works with or without the DETAIL option.



Most storage elements that are managed by TCPIP have a storage header that describes the storage attributes. Attributes include size, storage key, z/OS sub pool number, private or common, allocated or free, and storage pool

In V1R9 timestamps are in hex format which is not very useful and needs to be converted to “human readable” format in order to be understood. Also, for any storage element, you need to know if TCPIP considers it to be allocated or freed.

In V1R10, More fields of the TCPIP storage header are formatted. The timestamps no longer need to be converted to “human readable” format and it is easier to tell if the storage is allocated or freed.



If applications are waiting on events, an EVENT Control Block (EVNT) is queued to the connection control block. If EVNTs are not posted (meaning the EVENT does not complete) this can be an indication of a problem.

In V1R9, if an EVENT is queued to a connection control block, the EVNT has to be formatted “by hand”. Then the queue (chain) of EVNTs has to be manually counted to find how many events were queued.

In V1R10 Event formatting is part of DETAIL processing for options TCP, UDP, and RAW. The first queued event is formatted (including its timestamp) and the total number of waiting events is displayed.

IPCS formatter enhancements
SYSTCPIP session formatter using MIN options

- V1R10 z/OS Communication Server provides a session based formatter for SYSTCPIP traces containing PFSMIN and TCPMIN trace records
  - similar to that for the SYSTCPDA component,
- Enables viewing of flows on a per session basis if IPSEC is being used for a session

© 2008 IBM Corporation

Two of the more common SYSTCPIP component trace entries used in debugging problems are TCP and PFS. These trace entries provide a great deal of useful information for the PFS and TCP layers of the stack. However, there are times when a problem does not need such comprehensive data for debugging, but in fact needs key fields from the TCP and PFS layers using less storage and cpu cycles. This led to the development of the PFSMIN and TCPMIN SYSTCPIP component trace options in z/OS Communications Server V1R7. In V1R10 there is now a formatter for those options to make the results easier to understand.

The design and creation of the SYSTCPIP session trace formatter allows you to see the PFSMIN and TCPMIN trace entries in a readable, session based format. These options were used for the session formatter because they are more efficient than the existing TCP and PFS entries with respect to cpu cycles and storage. Also because they are invoked on packets before IPSEC encryption is performed and after IPSEC decryption is performed. Because of this, filtering on session IDs, IP addresses and ports can be performed even if IPSEC is used for the session.

IPCS formatter enhancements
SYSTCPIP session formatter using MIN options

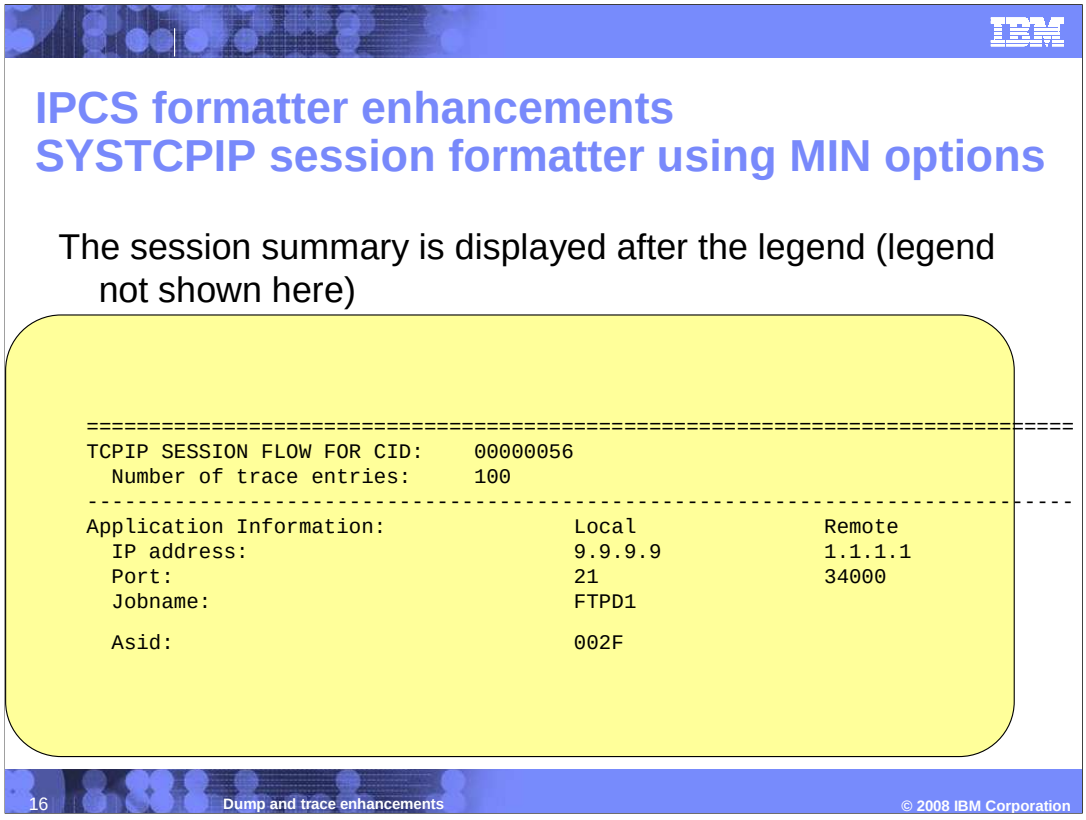
- IPCS formatter panel:

```
System      ===>           (System name or blank)
 Component   ===> SYSTCPIP  (Component name (required))
 Subnames    ===> TCPCS

 GMT/LOCAL   ===> G                      (G or L, GMT is default)
 Start time  ===>                        (mm/dd/yy,hh:mm:ss.dddddd or
 Stop time   ===>                         mm/dd/yy,hh.mm.ss.dddddd)
 Limit       ===> 0         Exception ===>
 Report type ===> FULL      (SHort, SUmmary, Full, Tally)
 User exit   ===>           (Exit program name)
 Override source ===>
 Options         ===> SESSION PORT(21)

 To enter/verify required values, type any character
 Entry IDs ===>   Jobnames ===>   ASIDs ===>   OPTIONS ===>   SUBS ===>

 CTRACE COMP(SYSTCPIP) SUB((TCPCS)) FULL OPTIONS((SESSION PORT(21)))
```

Select the session on port 21

© 2008 IBM Corporation

The trace is invoked the same way as the packet trace (SYSTCPDA) formatter. The formatter is invoked by adding the keyword 'SESSION' in the options field of the CTRACE DISPLAY PARAMETERS screen. You can also add other filtering criteria such as port and IP address in the same manner as is currently done with the SYSTCPDA formatter. On this slide the IPCS formatter panel is shown with the new SESSION keyword, selecting on port 21 (the FTP port).



The formatter always includes a legend displayed one time before the session output that describes the unique fields for the respective entries in the trace. There are ten different record types and a maximum of eight unique fields for each record type. The directional arrows are used to show the flow direction, from the application to the stack and the stack to the application. Certain letters within the directional arrows also have particular meanings and are also described in the legend.

The session summary information is the first thing presented with respect to the session. The information presented includes the local and remote IP address and port, and the local address space identifier and other session statistical information. It is the same output as presented in the SYSTCPDA session formatter.

IPCS formatter enhancements
SYSTCPIP session formatter using MIN options

- Sample session output

```
Session Summary:
RECTYP| TIME            |PFS|TCP|IP | MODID   UNIQUE-1 UNIQUE-2 UNIQUE-3 UNIQUE-4 . . .
------+-----------------+---+---+---+------  --------- -------- -------- -------- . . .
TCSN  | 17:29:47.003917 |   |   |-->| TCSND  01A6/0012 00000000 00000001 00000000 . . .
TCRV  | 17:29:47.011345 |   |   |<--| TCRD   68EE/0010 38D4716E 849DFA4A 00000000 . . .
PFSX  | 17:29:47.012033 |<--|   |   | PFACP   00000000 00000000 00000000 90000000 . . .
PFSE  | 17:29:47.028846 |-->|   |   | PFIOC   90000000     0000     0000
PFSE  | 17:29:47.855710 |-->|   |   | PFIOC   90000000     0000     0000
PFSE  | 17:29:47.857385 |S->|   |   | PFSDR   90280000     0000     0000
TCWE  | 17:29:47.857396 |   |-->|   | TCFWR            00000000 00000000 00000053 . . .
TCSN  | 17:29:47.857420 |   |   |-->| TCFWR  01A7/0010 00000000 00000001 00000053 . . .
TCWX  | 17:29:47.857548 |   |<--|   | TCFWR   00000053 00000000 00000000 00000053 . . .
PFSX  | 17:29:47.857550 |<-S|   |   | PFSDR   00000053 00000000 00000000 90280000 . . .
PFSE  | 17:29:47.870506 |S->|   |   | PFSDR   90280000     0000     0000
```


© 2008 IBM Corporation

After the legend and session summary, the session flows are displayed, as shown on this slide.

The session flows on this slide show the send socket call coming from the application, S→ , to the stack's PFS layer, it then passes from the PFS to TCP, and from TCP to IP. The call then traverses back through the same layers and the returns the send socket call to the application, ←S .

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 VTAM z/OS

A current list of other IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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 or changes in the products or programs 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 (for example, 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.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.