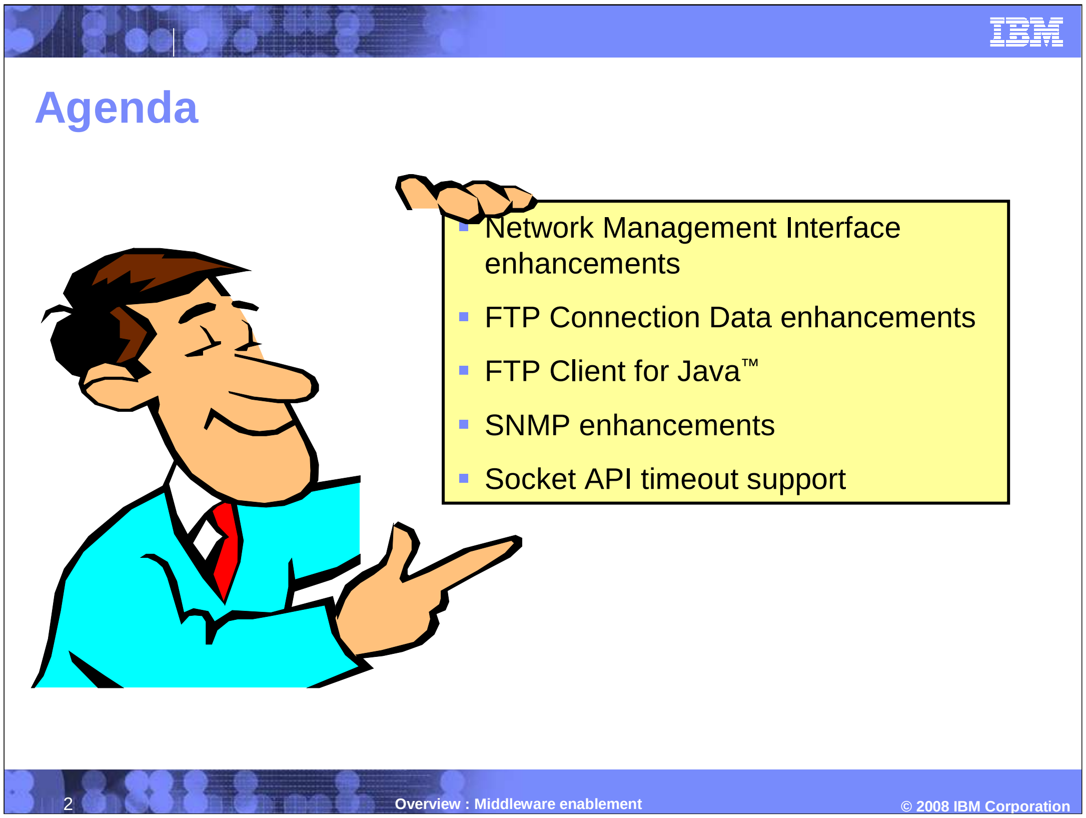
z/OS® V1R10 Communications Server

Overview : Middleware enablement

@business on demand software

This presentation is an overview of Middleware enablement enhancements for Communications Server for z/OS V1R10.



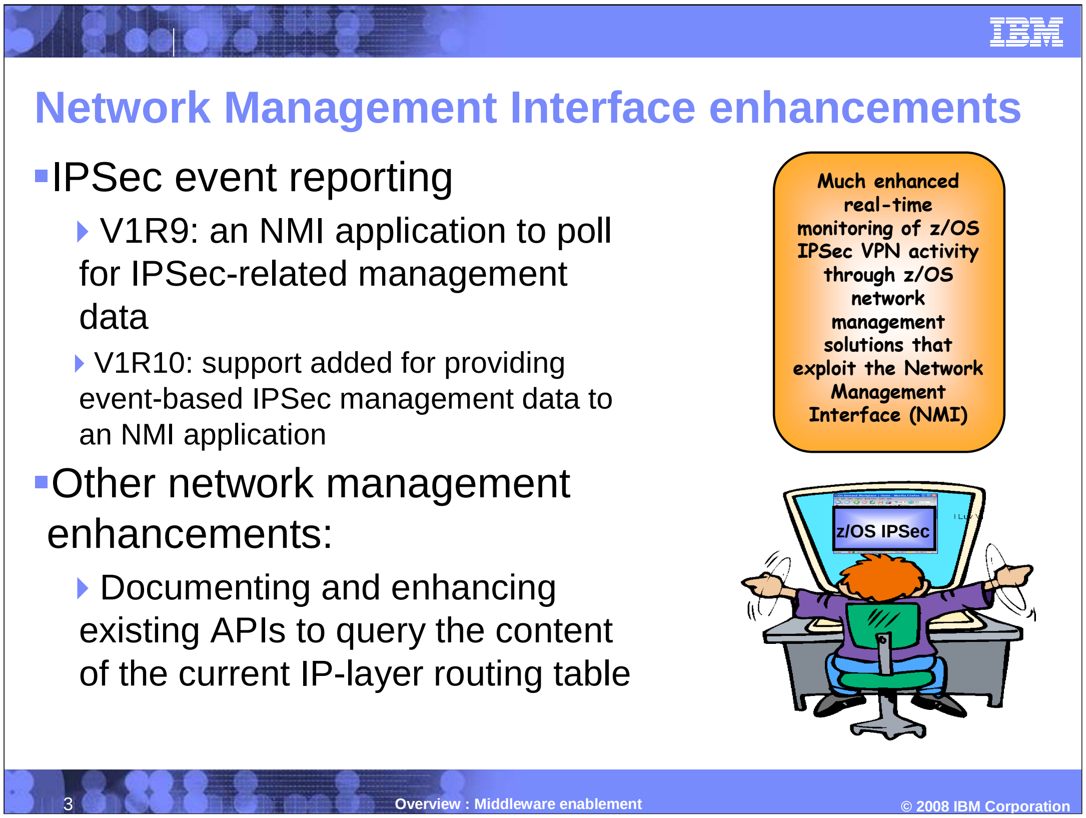
IBM

Agenda

- Network Management Interface enhancements
- FTP Connection Data enhancements
- FTP Client for Java™
- SNMP enhancements
- Socket API timeout support

2 Overview : Middleware enablement © 2008 IBM Corporation

This presentation introduces the Middleware enablement enhancement in CS z/OS V1R10. These include enhancements to the Network Management Interface, FTP Connection Data, FTP Client for Java, SNMP and Socket API timeout support.



Network Management Interface enhancements

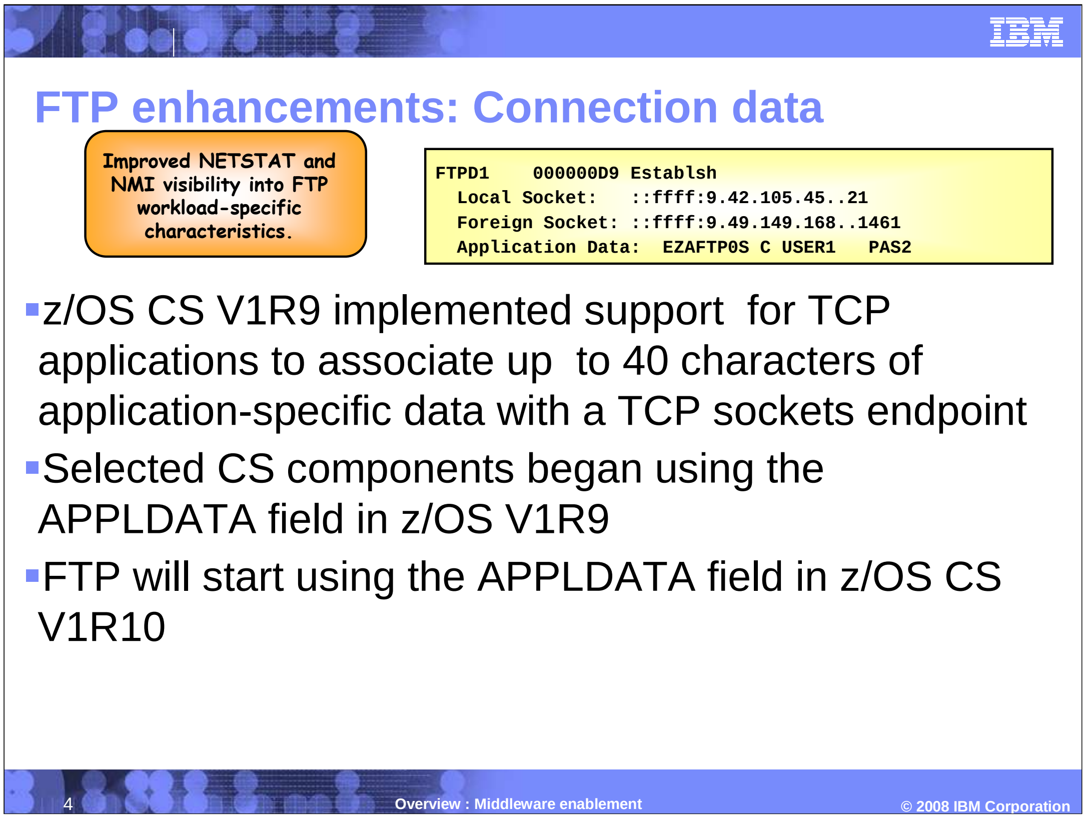
- IPSec event reporting
  - V1R9: an NMI application to poll for IPSec-related management data
  - V1R10: support added for providing event-based IPSec management data to an NMI application
- Other network management enhancements:
  - Documenting and enhancing existing APIs to query the content of the current IP-layer routing table

Much enhanced real-time monitoring of z/OS IPSec VPN activity through z/OS network management solutions that exploit the Network Management Interface (NMI)

The IPSec events are reported over the NMI interface in SMF119 record type format. These SMF119 records can also optionally be written to the SMF data sets for later post processing.

Event types were added for activation/deactivation phase 1 and phase 2 security associations, Refresh of phase 1 and phase 2 security associations, and manual and dynamic tunnel activation, deactivation, add and remove.

IP layer routing table contents are retrieved using an IO control call (SIOCGRTTABLE ioctl call).



IBM

FTP enhancements: Connection data

Improved NETSTAT and NMI visibility into FTP workload-specific characteristics.

```
FTPD1    000000D9 Establsh
  Local Socket:   ::ffff:9.42.105.45..21
  Foreign Socket: ::ffff:9.49.149.168..1461
  Application Data:  EZAFTP0S C USER1   PAS2
```

- z/OS CS V1R9 implemented support for TCP applications to associate up to 40 characters of application-specific data with a TCP sockets endpoint
- Selected CS components began using the APPLDATA field in z/OS V1R9
- FTP will start using the APPLDATA field in z/OS CS V1R10

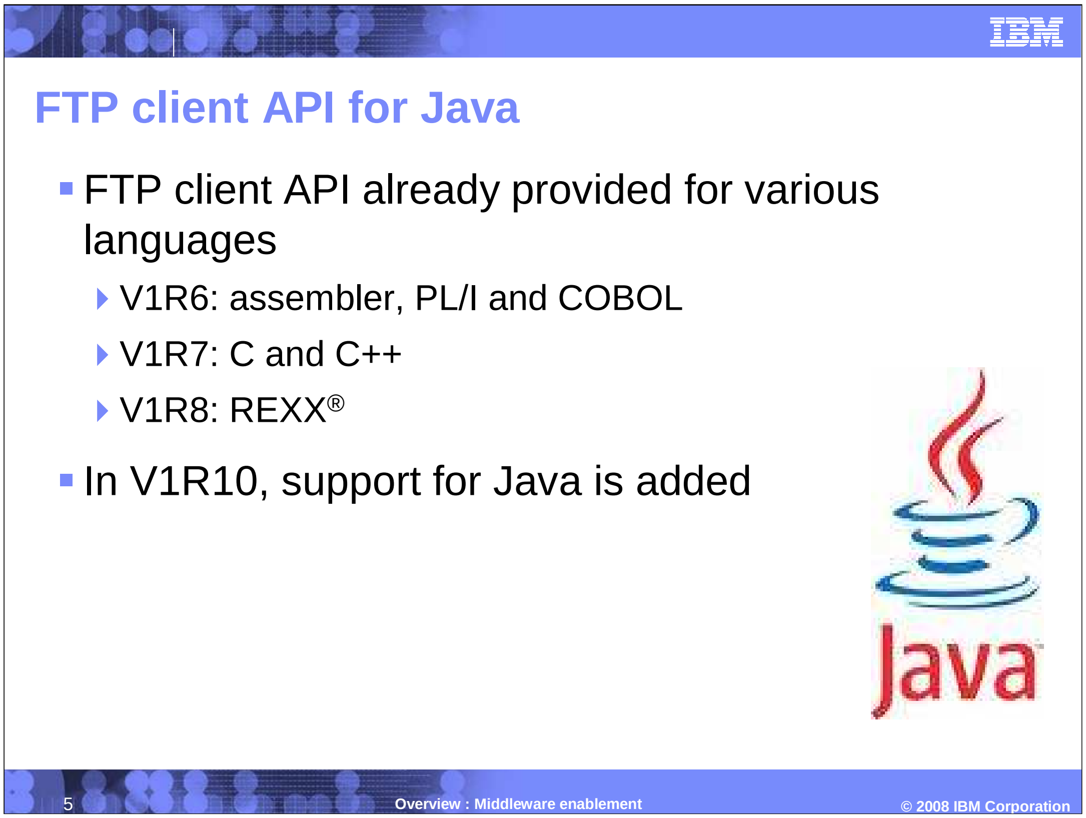
4 Overview : Middleware enablement © 2008 IBM Corporation

The APPLDATA field of the socket contains FTP connection data. It can be set or updated by a TCP application using an IOCTL sockets call and can be included in NETSTAT ALL, ALLCONN, and CONN reports and used as a filter. This data also can be included in the NMI, Network Monitor Interface. The suggested syntax for the field is to use an eight-character application identifier in the first eight characters of the 40-character APPLDATA field.

In V1R9, this support is used by CICS Sockets to associate CICS-specific information with CICS sockets endpoints such as, “EZACICSO SRV1 0000123 USER1234 CICA”.

Also in V1R9, it is used by the TN3270 server to associate TN3270-specific information with TN3270 sockets endpoints such as, “EZBTNSRV TCPABC80 TSO10001 ET B”.

Now in V1R10, both the FTP client and the FTP server will associate FTP-specific information with the FTP sockets endpoints. This information includes FTP component (Client, Server, Daemon), type of connection (Control or Data), User ID, Security characteristics (SSL/TLS, GSSAPI, Ciphers) and information about file being transferred (direction, type, location).



FTP client API for Java

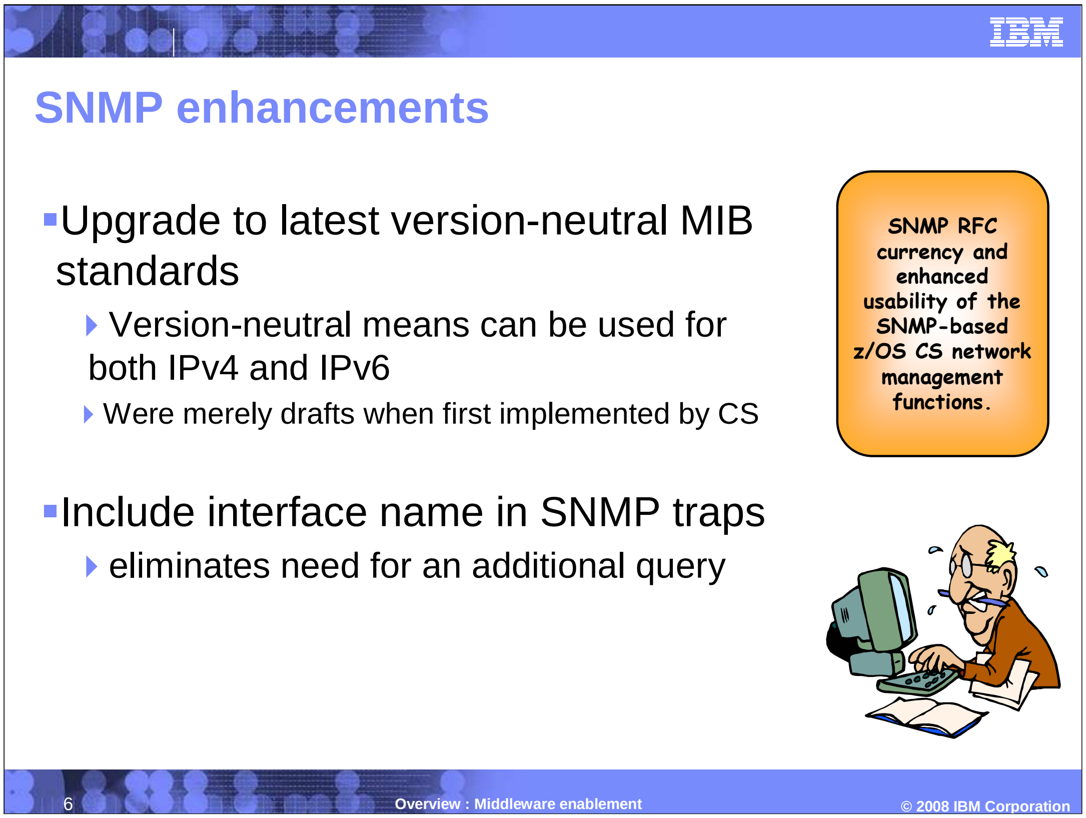
- FTP client API already provided for various languages
  - V1R6: assembler, PL/I and COBOL
  - V1R7: C and C++
  - V1R8: REXX®
- In V1R10, support for Java is added

Before V1R10, controlling the z/OS FTP client from within a Java application is difficult. z/OS V1R7 added an FTP client API for C and C++, but a Java application must use the Java Native Interface (JNI) to use this API. JNI is a complicated, low-level API and is not intended for use by the typical Java programmer.

There are other options for using FTP from within a Java application without interfacing with the z/OS FTP client. The core Java programming language has limited support for FTP using URL syntax. Using this, an application can retrieve a file from a remote FTP server and receive the results into the application program's memory. There are also vendor Java FTP libraries available which allow applications to use FTP to retrieve files from or store files on a remote FTP server. None of these options support z/OS features, such as MVS datasets.

To provide access to the z/OS FTP client from within a Java application, support for the Java programming language is added to the set of FTP client APIs that ship with z/OS Communications Server. The Java FTP client API consists of two parts. First, there is a package of Java classes that is used by a Java programmer to access the FTP client API. Second, there are native routines written in C++ that are invoked by the Java package using the Java Native Interface (JNI). Third, there is a sample program, ftpcapij.java, that uses the FTP client API for Java is provided.

The FTP client API for Java is documented in Javadoc. The Javadoc files consist of a set of HTML pages that describe the Java classes and the class methods. The Javadoc can be downloaded to a local workstation, the Javadoc files extracted, and viewed using any available Web browser.



SNMP enhancements

- Upgrade to latest version-neutral MIB standards
  - Version-neutral means can be used for both IPv4 and IPv6
  - Were merely drafts when first implemented by CS
- Include interface name in SNMP traps
  - eliminates need for an additional query

SNMP RFC currency and enhanced usability of the SNMP-based z/OS CS network management functions.

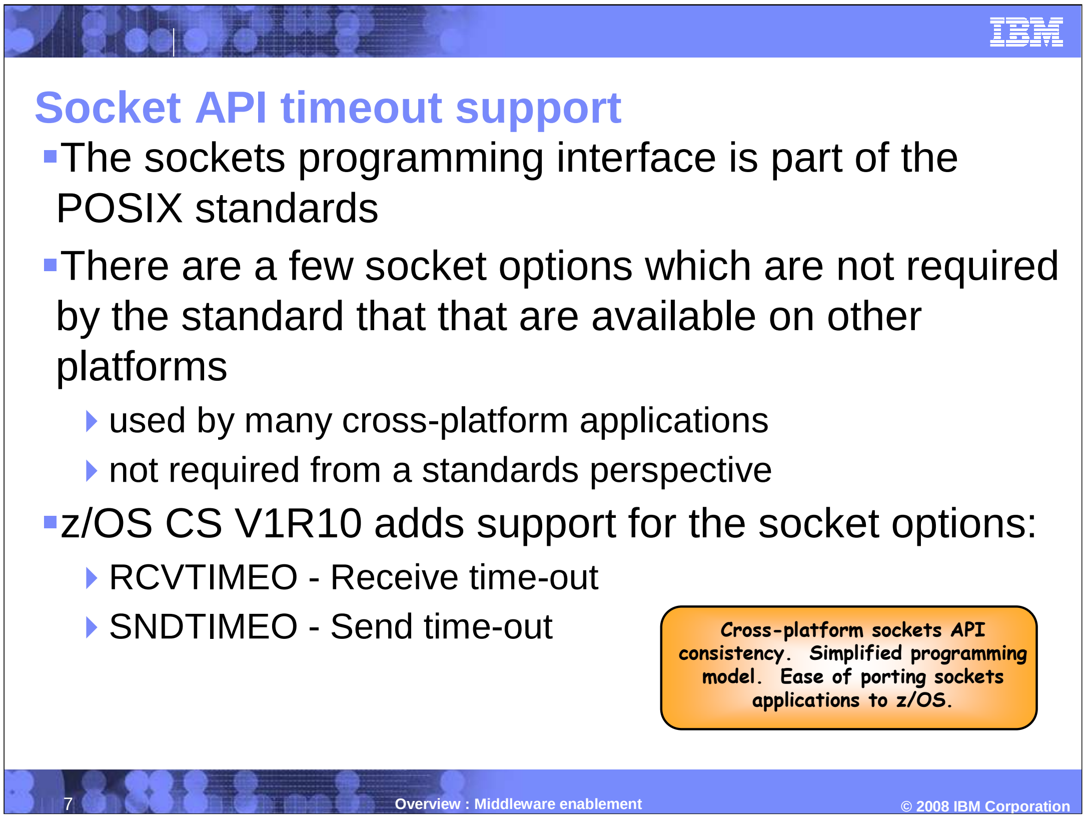
6 Overview : Middleware enablement © 2008 IBM Corporation

When z/OS CS implemented IPv6 support, it also implemented SNMP support for the so-called version-neutral MIBs (MIB variables that can be used for both IPv4 and IPv6 objects). At the point in time such support was implemented, the version-neutral MIBs were all defined in draft RFC status. These drafts have now moved into RFC status and a few updates have been made to them.

z/OS CS V1R10 updates its support for the version-neutral MIBs for several RFC levels. These include IP MIB - RFC4293, IP Forward MIB - RFC4292, TCP MIB - RFC4022, and UDP MIB - RFC4113.

The z/OS Communication Server includes support for generic LinkUp and LinkDown traps as described by RFC 2863. The generic LinkUp and LinkDown traps provide only an index to identify the affected link. Subsequent SNMP GET queries are required to determine the interface name of the affected link.

Some vendors (Cisco for example) have implemented enterprise-specific LinkUp and LinkDown traps to provide more information and eliminate the need for a subsequent GET request to fetch more information from the ifTable. z/OS will in V1R10 implement enterprise-specific linkUp and linkDown traps which include the interface name in the trap data.



Socket API timeout support

- The sockets programming interface is part of the POSIX standards
- There are a few socket options which are not required by the standard that that are available on other platforms
  - used by many cross-platform applications
  - not required from a standards perspective
- z/OS CS V1R10 adds support for the socket options:
  - RCVTIMEO - Receive time-out
  - SNDTIMEO - Send time-out

Cross-platform sockets API consistency. Simplified programming model. Ease of porting sockets applications to z/OS.

The name of the API standard is the Single UNIX Specification (SUS) Version 3 is sponsored by the Open Group (of which IBM is a member). The address is http://www.unix.org/single_unix_specification

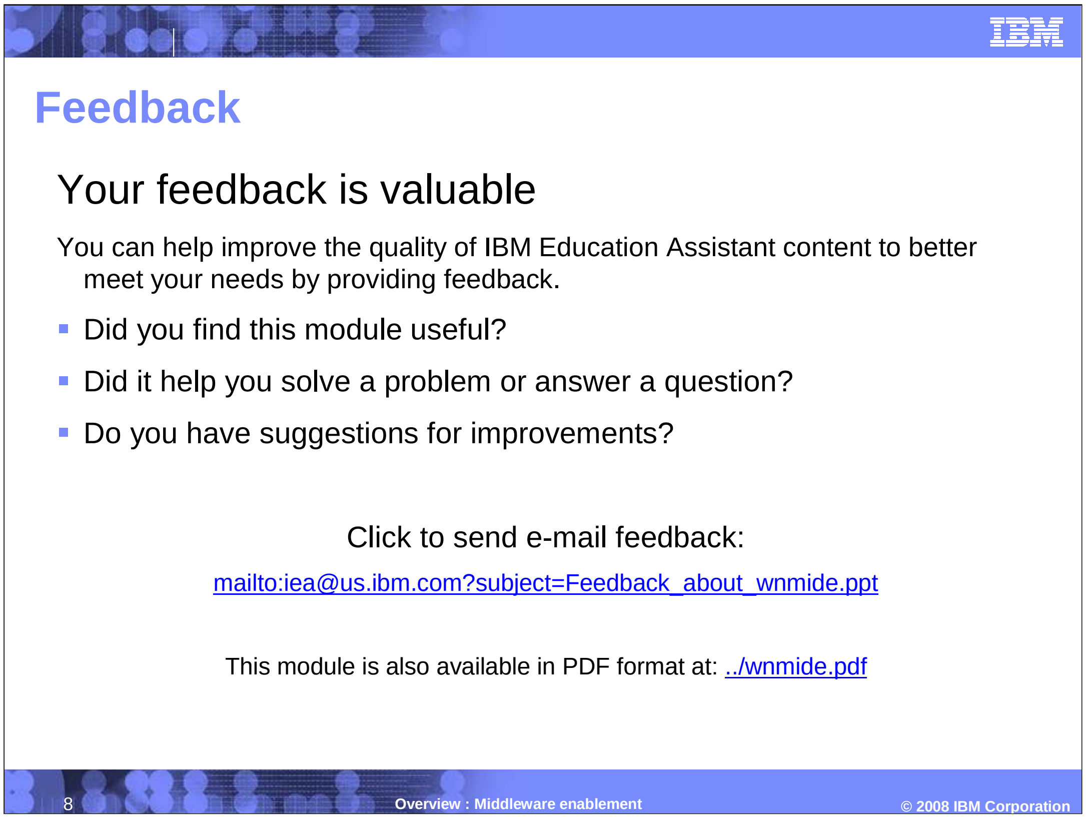
The RCVTIMEO option sets a maximum amount of time a sockets input operation will wait for data to arrive.

The SNDTIMEO option sets a maximum amount of time an output operation blocks (because TCP flow control prevents data from being sent).

Both sockets options are added to the setsockopt() call to set the options and the getsockopt() call to query the options.

These options provide for a simple mechanism by which an application issuing receive and send calls can protect itself against indefinite blocking.

Other, more complex programming mechanisms do exist to accomplish the same objective. These are Non-blocking sockets call with wait/retry logic, Select() call processing, and Asynchronous socket calls.



IBM

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?

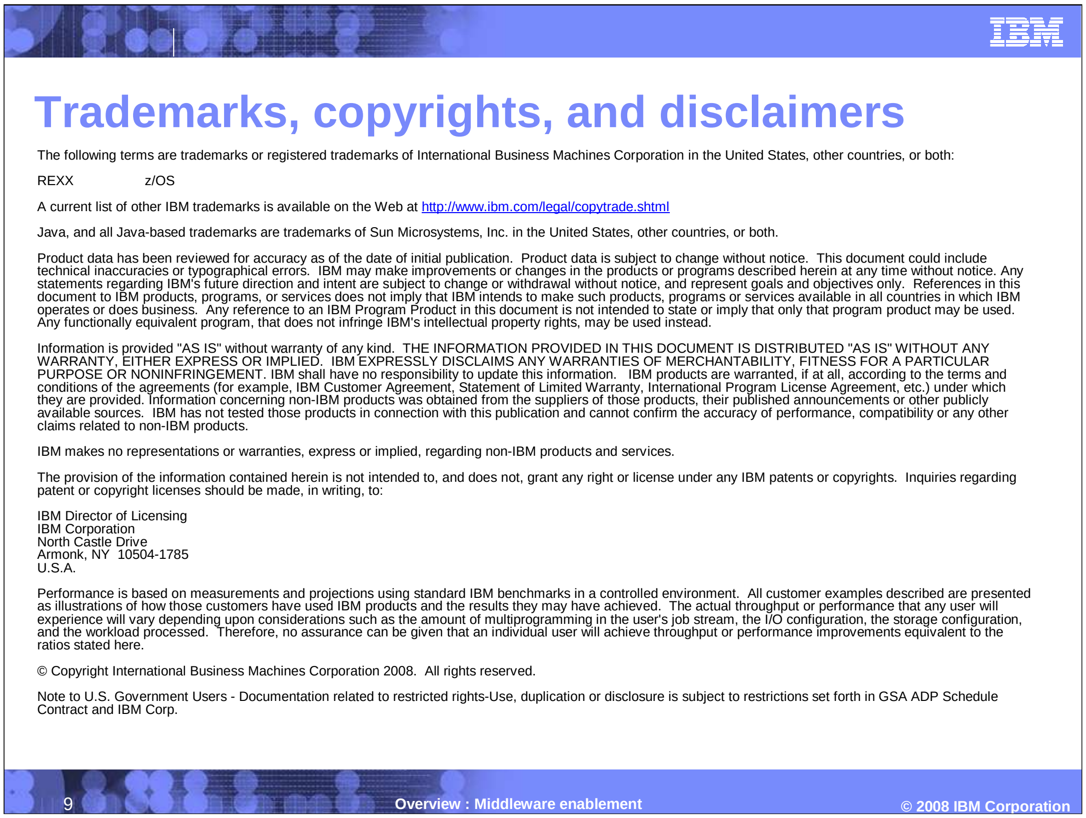
Click to send e-mail feedback:

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

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

8 Overview : Middleware enablement © 2008 IBM Corporation

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



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:

REXX	z/OS

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

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.

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.