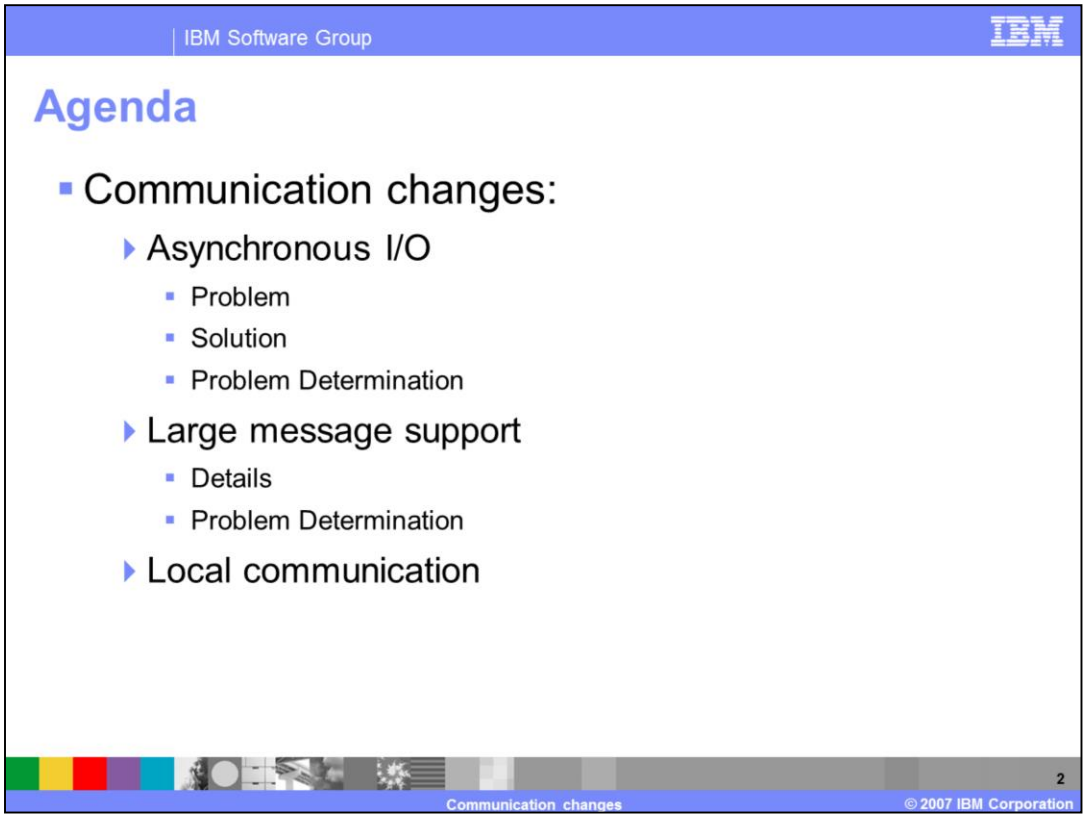
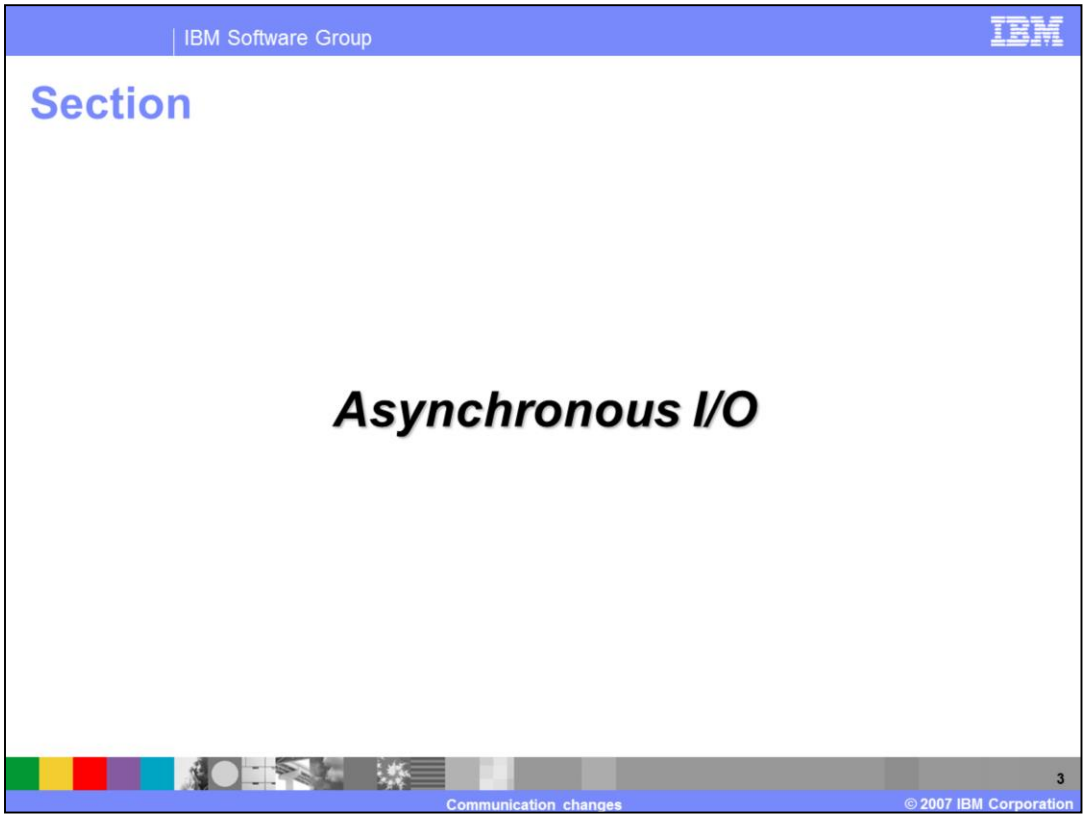
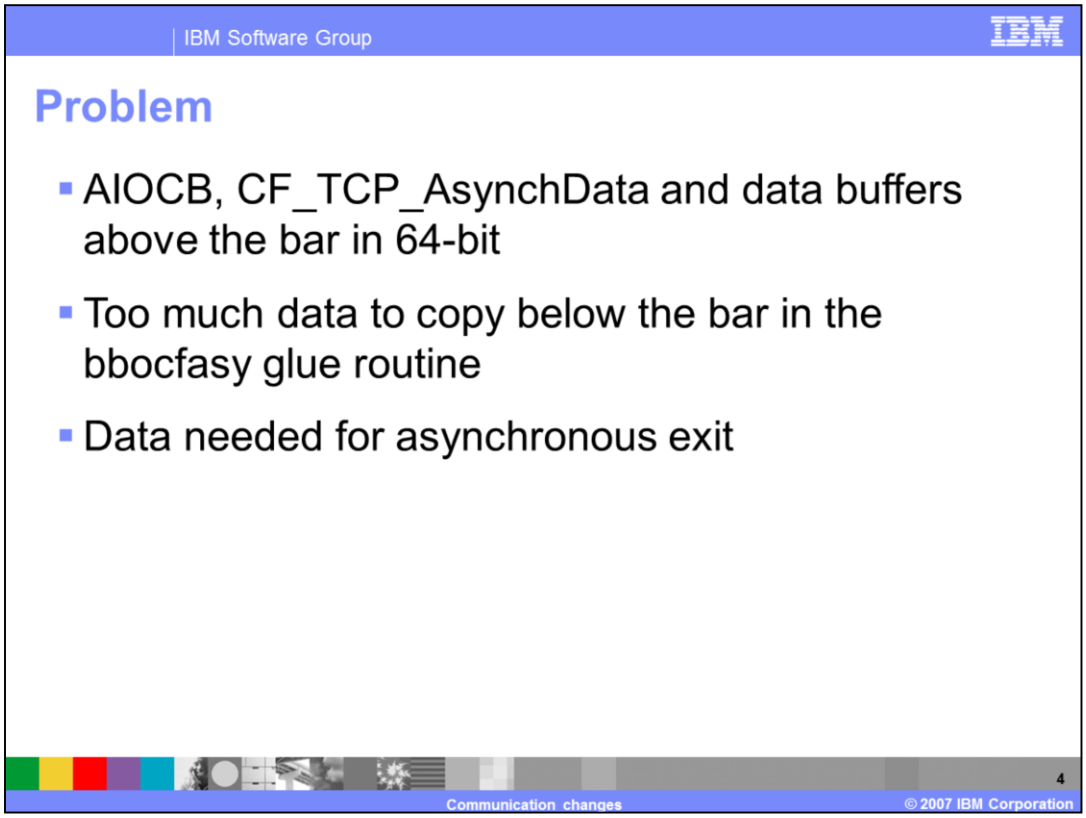
This presentation will discuss communication changes in 64 bit mode in a WebSphere Base Application Server V6.1 on z/OS.



This presentation will discuss communication changes related to asynchronous I/O and large message support. Local communication will also be discussed.

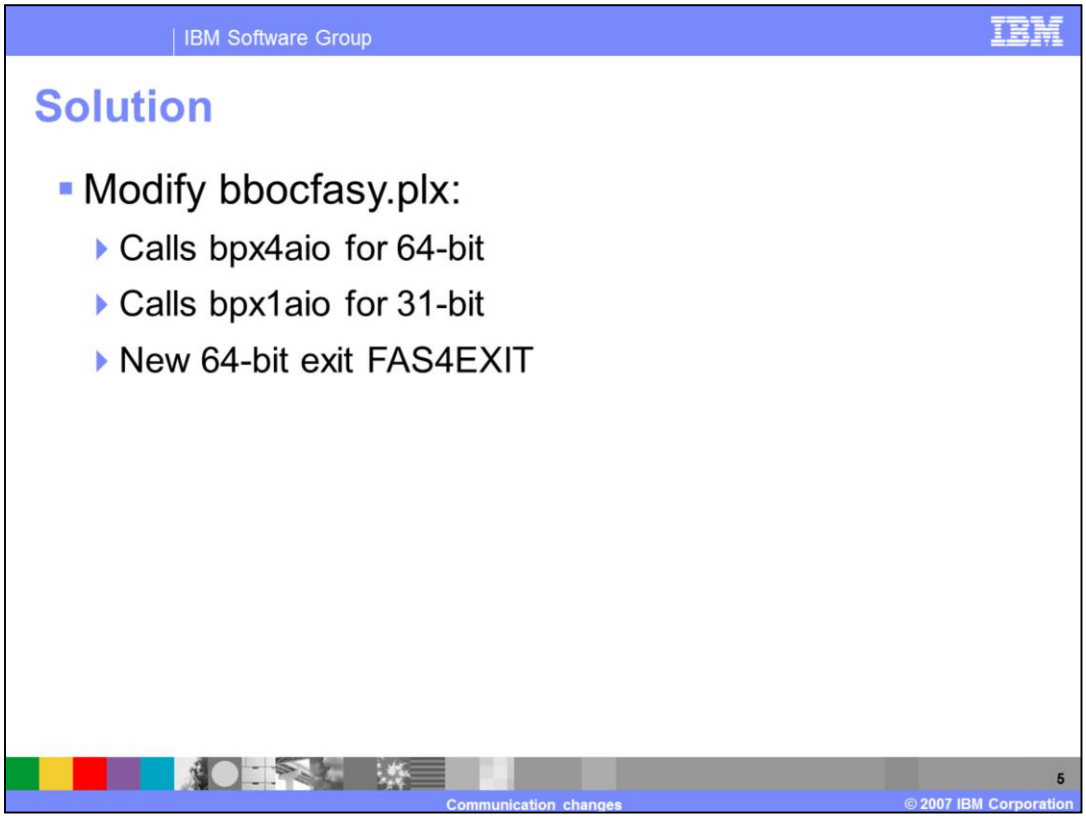


This section will discuss asynchronous I/O.



In 64-bit mode, both the AIOCB and the data buffers are above the bar. The data buffers in 64-bit mode can be very large, therefore, copying them below the bar is not practical due to storage issues

And these data areas need to be available when the asynchronous exit gets control. With glue code, any storage obtained is released after the PLX routine is called.

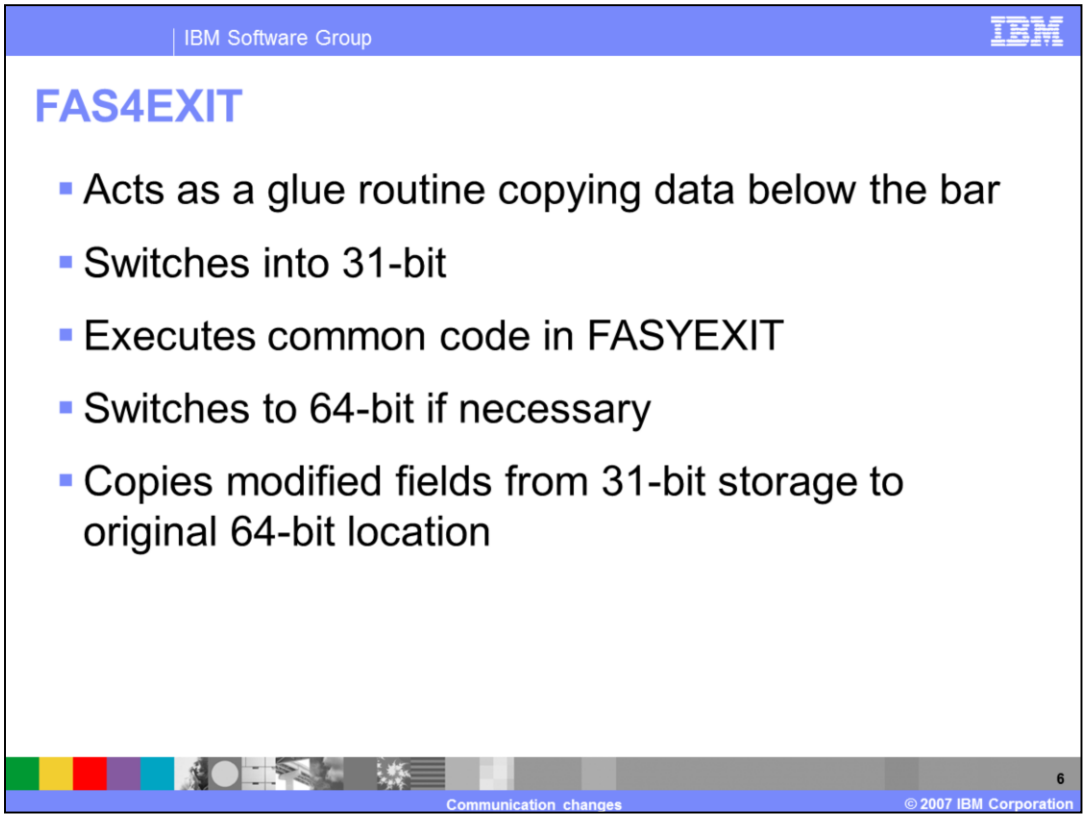


The solution to this problem is as follows:

BBOCFASY.PLX was modified to handle both 31-bit and 64-bit requests. In the main routine, which starts the asynchronous I/O routine, it determines whether the server is running in 31-bit or 64-bit mode.

If running in 64-bit, the 64-bit z/OS asynchronous I/O service, BPX4AIO, is called. Otherwise, the 31-bit asynchronous I/O service, BPX1AIO, is called.

The address of the exit to call when the asynchronous I/O completes is stored in the AIOCB before calling the asynchronous I/O service. For 64-bit, a new exit, FAS4EXIT, is specified; otherwise, FASYEXIT.



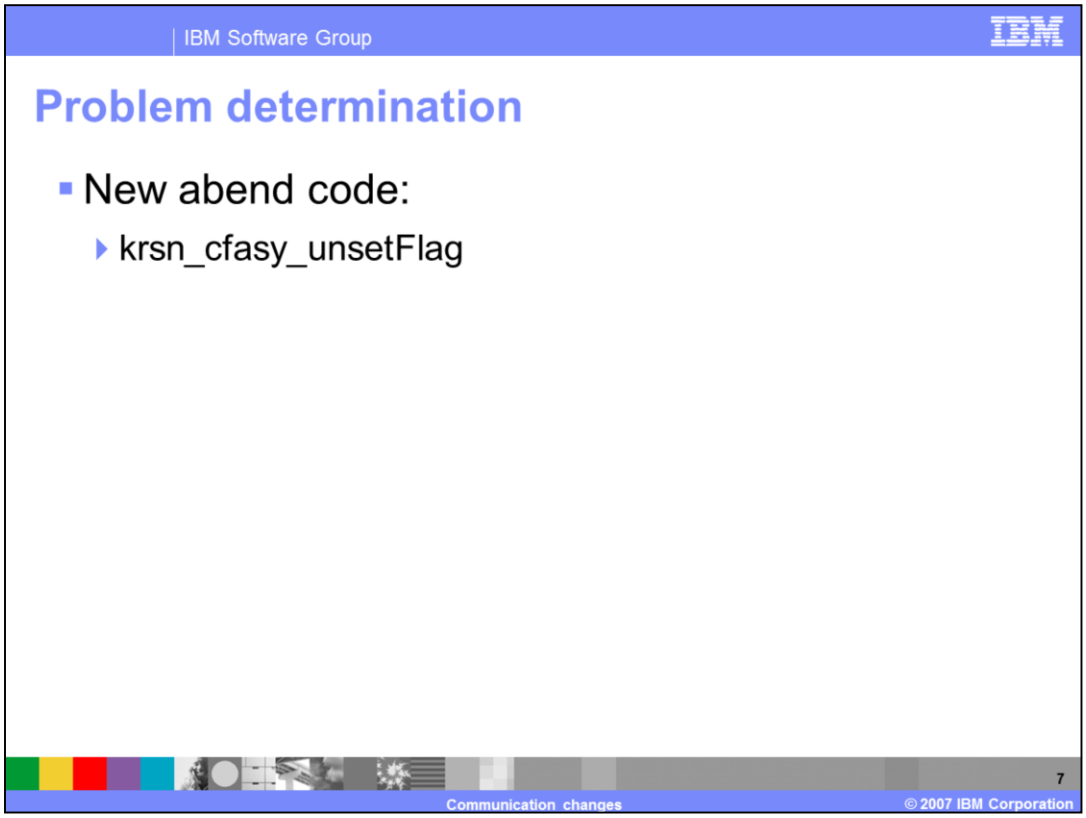
FAS4EXIT gets control when the asynchronous I/O routine completes.

The intent was to make use of as much of the existing exit code, FASYEXIT, as possible. So in FAS4EXIT, data that will be referenced during the exit processing is copied into 31-bit storage and a switch is made into 31-bit mode. It then joins a common code path with FASYEXIT.

Essentially FAS4EXIT acts as a glue routine for FASYEXIT.

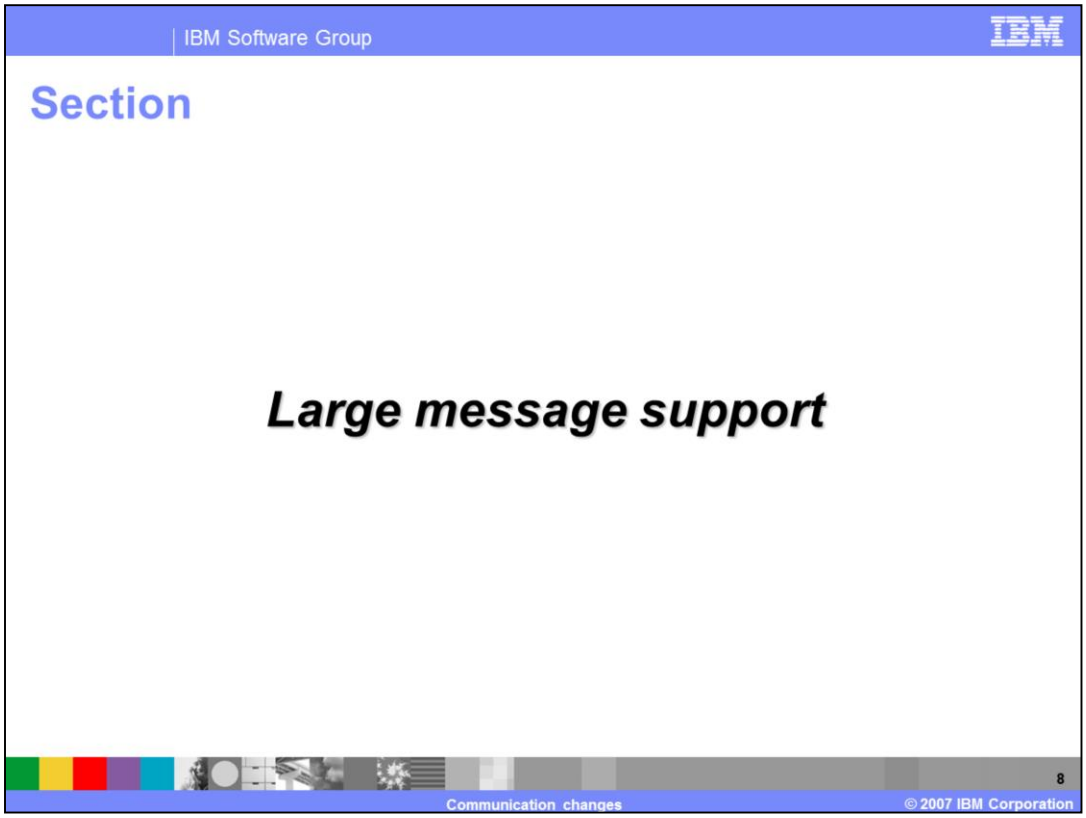
Where necessary, a switch is made to 64-bit mode. For example, if another asynchronous READ needs to be issued during exit processing and the server is running 64-bit mode, a switch is made to 64-bit mode and BPX4AIO is called. Then it switches back to 31-bit to continue the common FASYEXIT processing.

At the end of the processing, a switch is made back to 64-bit mode and the fields that had been copied below the bar are copied back to their 64-bit storage locations. This ensures all changes are reflected back to the handler of the completed I/O. Again, FAS4EXIT is acting as a glue routine.

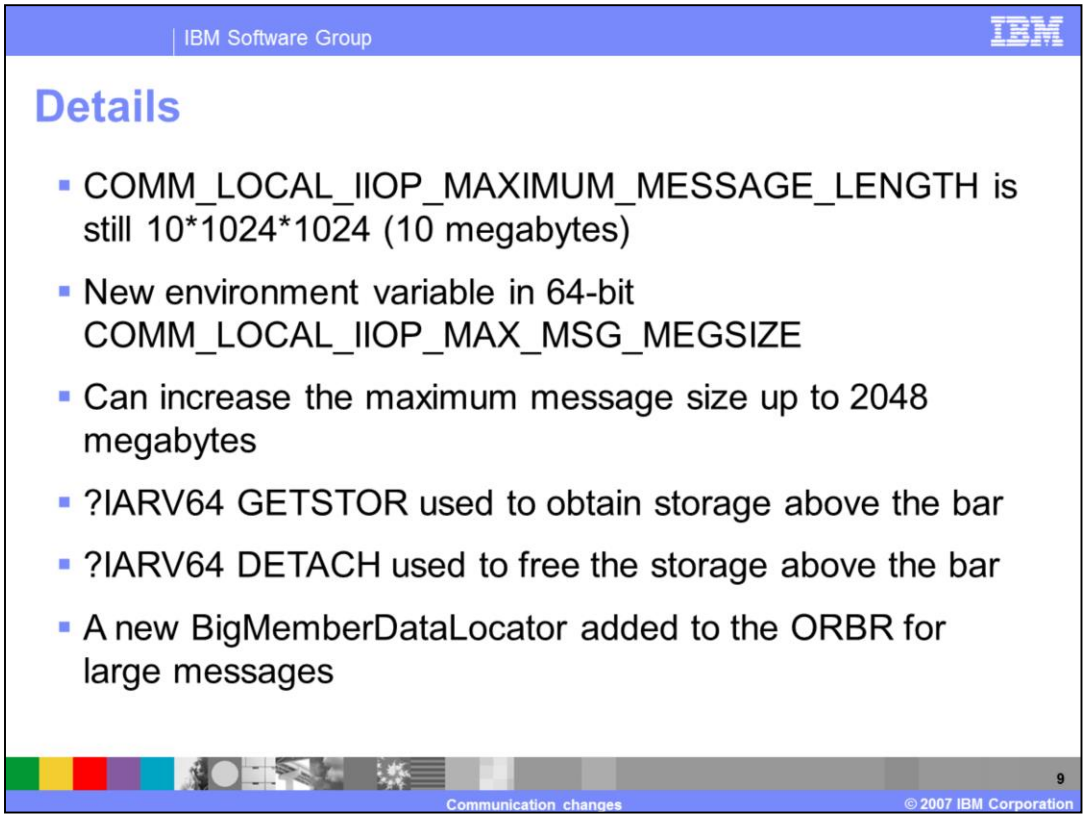


BBOABEND is issued with the reason code krsn_cfasy_unsetFlag in the main procedure of BBOCFASY if bacb_addr_mode_set is not set. This flag indicates whether the mode flag for 31-bit or 64-bit has been set. This is unexpected and thus an abend is issued.

There are no other changes to problem determination.



This section will discuss large message support.

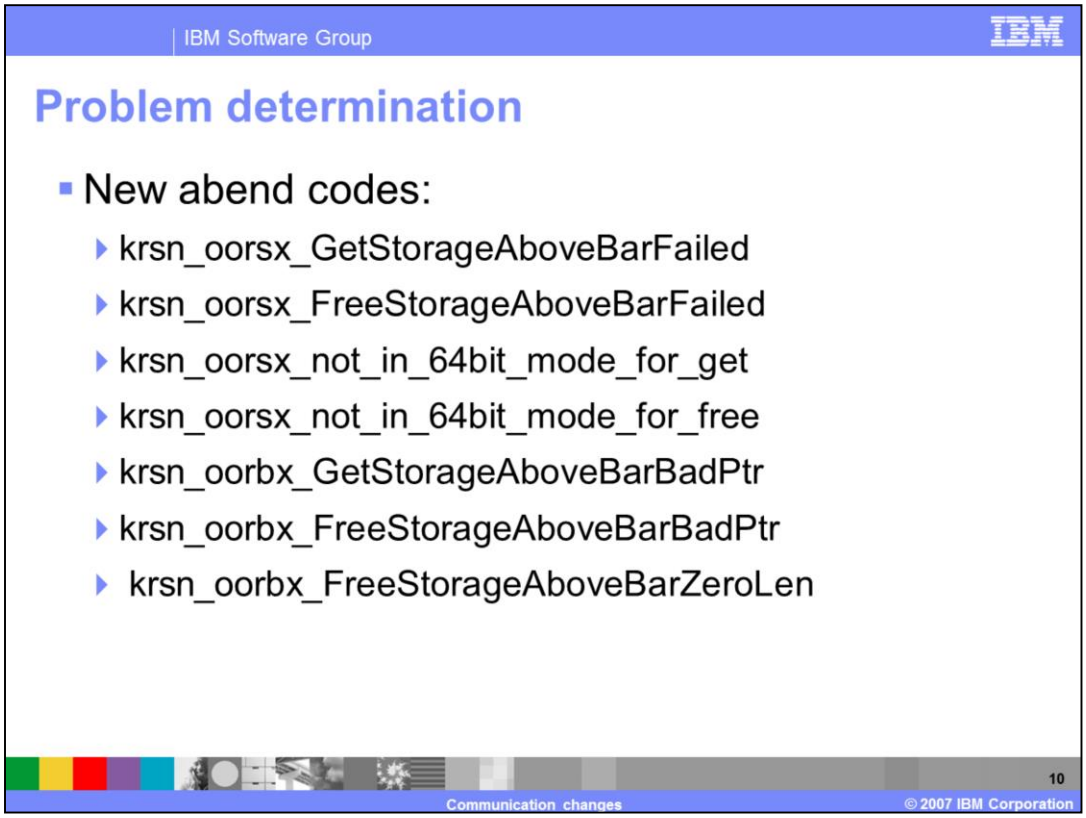


The 10 megabyte message limit is still in effect for 31-bit as an absolute maximum. In 64-bit mode, it is the default but it can be increased by the use of a new environment variable called COMM_LOCAL_IIOP_MAX_MSG_MEGSIZE which affects the 'meg' part of COMM_LOCAL_IIOP_MAXIMUM_MESSAGE_LENGTH.

The maximum value for COMM_LOCAL_IIOP_MAX_MSG_MEGSIZE is 2048 making the maximum message size 2048 megabytes.

When storage is needed for a large message in 64-bit mode, the storage is obtained above the bar using ?IARV64 GETSTOR and is freed using ?IARV64 DETACH.

New fields were added to the ORB_Request_BigMemberDataLocator for large messages: vLargeAttribute_Ptr, vLargeAllocatedTotalLen, and vLargeAttributeLength.



The following are some problem determination descriptions:

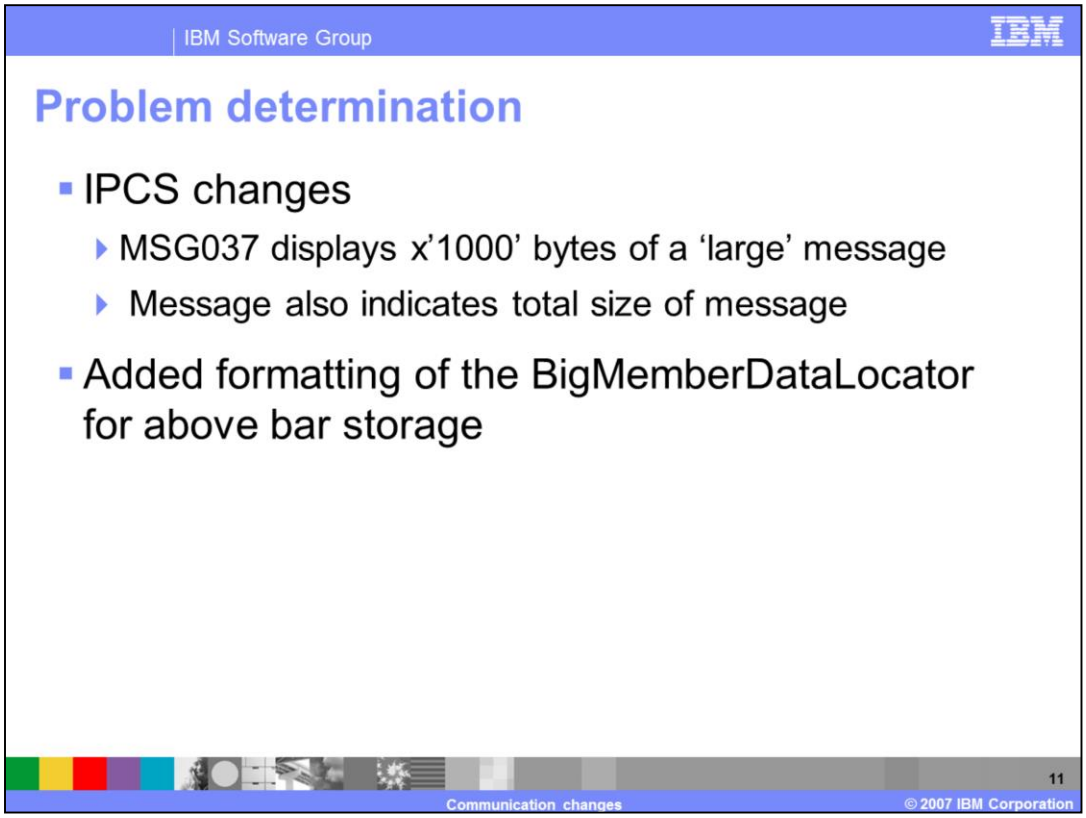
Krsn_oorsx_GetStorageAboveBarFailed and Krsn_oorsx_FreeStorageAboveBarFailed indicate a bad return code from the ?IARV64 service.

krsn_oorsx_not_in_64bit_mode_for_get and krsn_oorsx_not_in_64bit_mode_for_free indicate that the caller was not running in 64-bit mode when the getStorageAboveBar or freeStorageAboveBar routines were called.

krsn_oorbx_GetStorageAboveBarBadPtr and krsn_oorbx_GetFreeStorageAboveBarBadPtr indicate that a NULL pointer was supplied to either getStorageAboveBar or freeStorageAboveBar.

krsn_oorbx_FreeStorageAboveBarZeroLen indicates that a length of 0 was supplied to freeStorageAboveBar.

Since the routines, getStorageAboveBar and freeStorageAboveBar, are internal routines, it is not expected that these errors would occur and that is why an abend is issued if they do.



Since messages above the bar can be very large, MSG037 displays just x1000 bytes of the message. It gives both the address of the storage being displayed and the total length of the message.

The BigMemberDataLocator for above the storage is formatted in the dump along with the other BigMemberDataLocators. It displays the pointer to the above the bar location, the total number of bytes associated with this location, and the number of bytes in use.

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

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.