IBM Education Assistance for z/OS V2R1

Item: RTM Miscellany
Element/Component: BCP Recovery Termination Manager

Material is current as of March 2013

Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

- SdwaReleaseCode
- 64-bit FRR retry address in the System Trace
- Cross-memory failing instruction text for Estaex
- SDWAMABD added to SDWANMFS
- ASID added to the VRA for ABEND30D

Overview

- Problem Statement / Need Addressed
  - The Pause release code may not be available to a program which has been abended while Paused
  - Users of the limited 64-bit support need to see an entire FRR retry PSW
  - The failing instruction stream in the SDWA (SDWAFAIN) is not correct in some cross-memory situations
  - The 'Not My Fault Summary bit' in the SDWA (SDWANMFS) does not include RTM-initiated Detaches
  - For an ABEND30D, it is not obvious which ASID was terminating
- Solution
  - Provide the Pause release code in SdwaReleaseCode
  - Provide a 64-bit retry address in FRR retry entries in System Trace
  - Provide the failing instruction stream for RTM2 cross-memory cases
  - Include SDWAMABD in SDWANMFS
  - Include the terminating ASID in the VRA of ABEND30D logrec records
- Benefit / Value
  - More information available for debugging problems

Usage & Invocation – SdwaReleaseCode

- After a Release has been issued for a Paused task, there is a window of time where an asynchronous abend (CALLRTM TYPE=ABTERM) can interrupt the task in such a way that the release code is 'lost' because it is not visible to the task's recovery and after the abend the task is not Paused any longer
- RTM now provides this release code in new field SdwaReleaseCode so that it is visible to recovery
- New bit SdwaReleaseCodeValid indicates when SdwaReleaseCode contains a value of interest

Usage & Invocation – 64-bit FRR retry address in the System Trace

- FRR processing does not support 64-bit addresses
- As of z/OS V1R13, users of the limited support for executable code above the 2G bar have been able to arrange for FRR retry to that code by placing the actual retry address into 64-bit GPR 15 and requesting a retry to the address of CVTBSM0F. When a program does that, it would be valuable to see the actual retry address in the System Trace rather than the address of CVTBSM0F
- FRR retry processing now recognizes a request to retry to CVTBSM0F. In that situation, the contents of 64-bit GPR 15 are traced as the retry address
- For normal FRR retry, the System Trace entry contains a 64-bit address which was generated from the actual 32-bit retry address (the first word of this address will always be zero)

Usage & Invocation – Cross-memory failing instruction text for Estaex

- It has been a long-standing concern that the failing instruction stream (SDWAFAIN) may be incorrect for an abend that has occurred in other than the Home address space for ESTAEX, ARR, and IEARR recovery routines
- For cases where the failing instruction stream was not previously obtained before an Estae-type recovery routine is going to receive control, RTM now obtains it from the correct address space

Usage & Invocation – SDWAMABD added to SDWANMFS

- Bit SDWANMFS was added in z/OS V1R12 to provide a summary bit for a 'not my fault' list of SDWA indicators which show that a program has been abended for reasons outside of its direct control. Some recovery routines use this bit when determining whether to document (eg: take a dump or ask for a LOGREC record) that they were entered
- When a subtask is detached by RTM because its mother task has failed, SDWANMFS may not be on because the subtask is presented with the mother task's abend information (instead of the usual Detach ABEND13E which would cause SDWANMFS to be set on) and SDWANMFS was not on for the mother task's abend
- SDWAMABD indicates that RTM has detached this subtask, and has been added to the list of indicators included in SDWANMFS

Usage & Invocation – ASID added to the VRA for ABEND30D

- RTM memory termination processing in ASID 1 attaches a subtask to call End Of Memory resource managers for a terminating address space
- If the subtask remains dormant for 4 or minutes, RTM assumes that the subtask has 'hung' and takes an SVCDUMP for ABEND30D before issuing the abend to drive the EOM resource manager's recovery
- Before taking the SVCDUMP, RTM fills in the VRA of an SDWA with information about the failure so that it will be available in the dump
- This change adds the ASID of the terminating address space to that VRA to make debugging the problem a little easier

Appendix

- The system trace entry for FRR retry is documented in the RCVY trace entry section of the System Trace chapter in z/OS MVS Diagnosis: Tools and Service Aids GA22- 7589
- SdwaReleaseCode, SdwaFain, and SdwaNMFS are all documented in the SDWA mapping macro, IHASDWA