



Tivoli Workload Scheduler for z/OS

Data set triggering overview

Tivoli. software

© 2009 IBM Corporation
Updated August 10, 2009

Tivoli Workload Scheduler for z/OS® data set triggering overview.

This training module provides a high level overview of the data set triggering feature of Tivoli Workload Scheduler for z/OS. Data set triggering is part of Tivoli Workload Scheduler for z/OS 8.5 Event Driven Workload Automation (EDWA).

Data set triggering overview

- Changes the availability status of a special resource when a specific data set is closed
 - Can work with event-triggered tracking to dynamically add an application to the Tivoli Workload Scheduler for z/OS current plan
- Intercepts SMF records that are related to the closing of a data set
 - ▶ Creates a data set trigger special resource availability event
 - ▶ Is a type **S** event
- Requires the definition and implementation of data set triggering criteria definitions within Tivoli Workload Scheduler for z/OS
 - ▶ Used by tracker started task and subsystem
- Uses the same event tracking data flow that Tivoli Workload Scheduler for z/OS uses for other job-tracking events

Data set triggering overview.

Data set triggering changes the availability status of a special resource when a specific data set is closed. Data set triggering can also add an application to the Tivoli Workload Scheduler for z/OS current plan when a specific data set is closed. A data set that is defined as a candidate for data set triggering is represented by a special resource definition within Tivoli Workload Scheduler for z/OS. Data set triggering can cause a special resource availability event. The event-triggered tracking feature of Tivoli Workload Scheduler for z/OS processes the event to get an application dynamically added to the current plan.

Event recording begins by intercepting and examining System Management Facilities (SMF) data set close records using an SMF-provided exit point. Intercepting these SMF records can result in the creation of a Tivoli Workload Scheduler for z/OS type S data set triggered special resource availability event record. The tracker subsystem compares the name of the closed data set to the data set triggering criteria definitions that it has available before creating a type S event record. If the name and definitions match, the tracker subsystem creates a type S event record.

The event records creation and tracking process is the same general event tracking process that is used by Tivoli Workload Scheduler for z/OS for other job-tracking events. An explanation of the data flow for the events creation and tracking process is available on a subsequent slide in this module.

A different process is available for data set triggering on Unix System Services (USS), Hierarchical File System (HFS), and z/OS File System (ZFS) files. You can find information about the process in the *Filewatch implementation for USS* training module.

Data set triggering Type S event recording

1. The Tivoli Workload Scheduler for z/OS tracker subsystem records the initial data set close event as a type **S** event in memory
 - The SMF exit routine named IEFU83 intercepts the data set close SMF record and invokes the tracker subsystem
 - The tracker subsystem records the event into a system memory area called extended common service area (ECSA) if the event meets the data set triggering selection criteria
2. A type **S** event record is written to the tracker started task address space event data set
 - The event record is removed from the ECSA writer queue by the Tivoli Workload Scheduler for z/OS tracker started task event writer task
 - The event record is written to the tracker started task event data set by an event writer task
 - The tracker started task event data set DD name is EQQEVDS

Data set triggering type S event recording.

The first step of event recording is the recording of a data set close event into the Extended Common Service Area (ECSA) memory writer queue (WTRQ). This initial type S event record is based on the discovery of a data set close activity through the use of a SMF provided exit point and several specific SMF enabled records. The SMF exit routine IEFU83 invokes the tracker subsystem after the routine intercepts a data set close SMF record. The tracker subsystem compares the data set name provided by the SMF record to the data set triggering selection criteria that the tracker subsystem has available. If the name matches the criteria, the entry is written into ECSA.

In the second step the tracker started task address space event writer removes the type S event from the ECSA writer queue. The event writer writes a type S event record into the tracker started task event data set. The event data set has a DD name of EQQEVDS in the tracker started task JCL.

Data set triggering setup overview

- SMF requires SYS1.PARMLIB setup
 - ▶ SMF exit routine IEFU83
 - ▶ SMF records 14,15, 64
 - ▶ See the *SMF setup* training module for details
- Data set trigger selection criteria definitions must be supplied to the tracker
 - ▶ For information about the assembled macros, see the *EQQDSLST Implementation* training module
 - ▶ For information about the XML definitions and event driven workload automation (EDWA), see the *XML Implementation* training module
 - ▶ You can use the assembled macros and XML concurrently

Data set triggering setup overview.

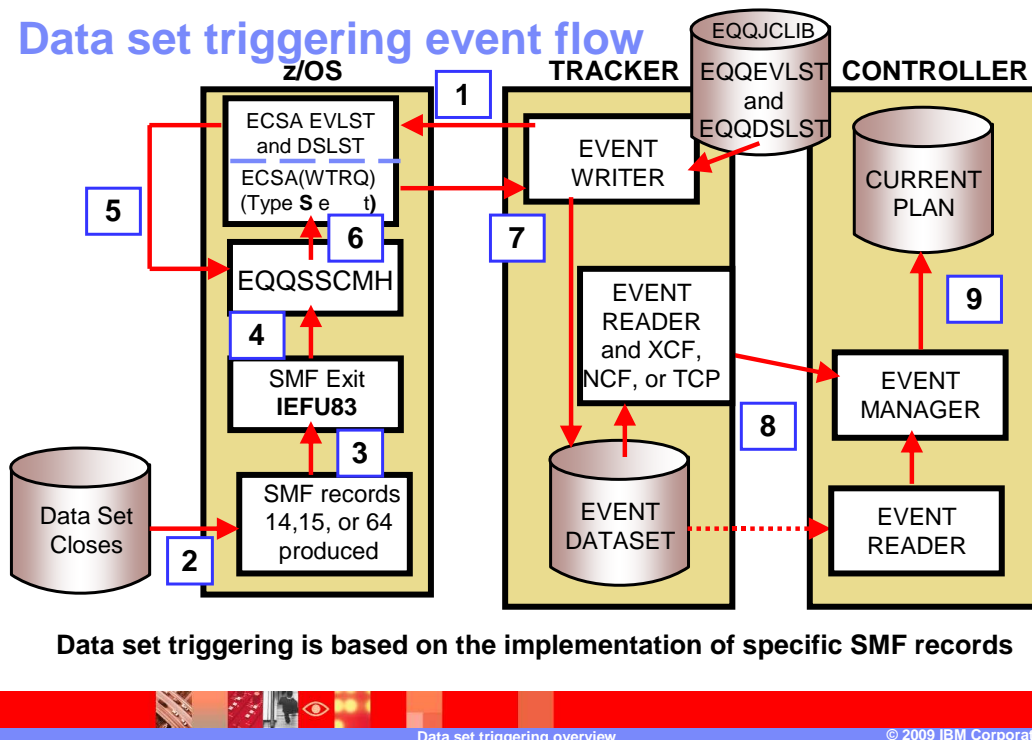
The event recording described on the previous slide has two areas of setup support. The first area of support is the SMF setup. You set up SMF in SYS1.PARMLIB to intercept the SMF data set close records using the SMF provided exit point. This setup includes the IEFU83 exit routine and the three SMF data set close records numbered 14,15, and 64. The second area of setup support is the creation and implementation of data set trigger selection criteria for the tracker started task and tracker subsystem.

Two methods are available for defining the selection criteria. The first method uses one or more assembled EQQLSENT macros, which contain the data set names and other selection criteria. These macros, provided by Tivoli Workload Scheduler for z/OS, are assembled into the EQQJCLIB data set in member EQQDSLST. The tracker started task reads the member EQQDSLST at the first startup only or after the operator issues an MVS console modify command. See the training module *EQQDSLST Implementation* for details.

The second and newer method provided with Tivoli Workload Scheduler for z/OS 8.5 is to define the data set triggering criteria in XML definitions. A job processes the definitions and distributes them to the tracker started task as member EQQEVLSST of the data set EQQJCLIB. This newer method is part of Event Driven Workload Automation (EDWA). See the training module *XML Implementation* for details.

A tracker started task and subsystem can use the EQQDSLST and EQQEVLSST methods at the same time. If both members are present, they are merged.

Data set triggering event flow



Data set triggering event flow.

This slide depicts an overview of the data flow for a data set close event that results in a data set trigger event. Assume that the SMF setup has been done. Also assume that data set trigger definitions are in either EQQDSLST or EQQEVLSST and are available to the tracker started task JCL using the DD name EQQJCLIB. The tracker and controller started tasks have also been started.

The tracker started task event writer task loads the merged EQQEVLSST and EQQDSLST members from EQQJCLIB into ECSA when the tracker started task address space is started for the first time after an IPL. A data set closes on the z/OS system and SMF produces an SMF record type 14, 15, or 64. SMF invokes the IEFU83 exit routine passing the SMF record to the exit routine. The IEFU83 exit routine invokes the tracker subsystem EQQSSCMH event-creation module and passes the SMF record information to it. EQQSSCMH compares the data set name in the SMF record to the data set trigger selection criteria in ECSA.

If the name matches the criteria, the tracker subsystem event-creation module EQQSSCMH records the type S data set close event into the writer queue of ECSA. The tracker event writer task in the tracker started task address space removes the type S entry from ECSA and writes a type S event record into the tracker started task event data set.

The tracker started task notifies the controller started task that the event has occurred using an event reader task and whatever communications method has been configured: XCF, NCF, or TCP/IP. If shared DASD is used, the controller reads the tracker event data set using a controller-based event reader task, as indicated by the dashed arrow on the slide. The controller updates special resource availability status. Event-triggered tracking adds an application to the current plan.

Summary

You should now be able to describe the components and data flow associated with the implementation of Tivoli Workload Scheduler for z/OS data set triggering

Summary.

In Summary you should now be able to describe the components and data flow associated with the implementation of Tivoli Workload Scheduler for z/OS data set triggering.

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_data-set-triggering-overview.ppt

This module is also available in PDF format at: ../data-set-triggering-overview.pdf

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

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Tivoli z/OS

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or 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 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.

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 NON-INFRINGEMENT. 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 2009. 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.

