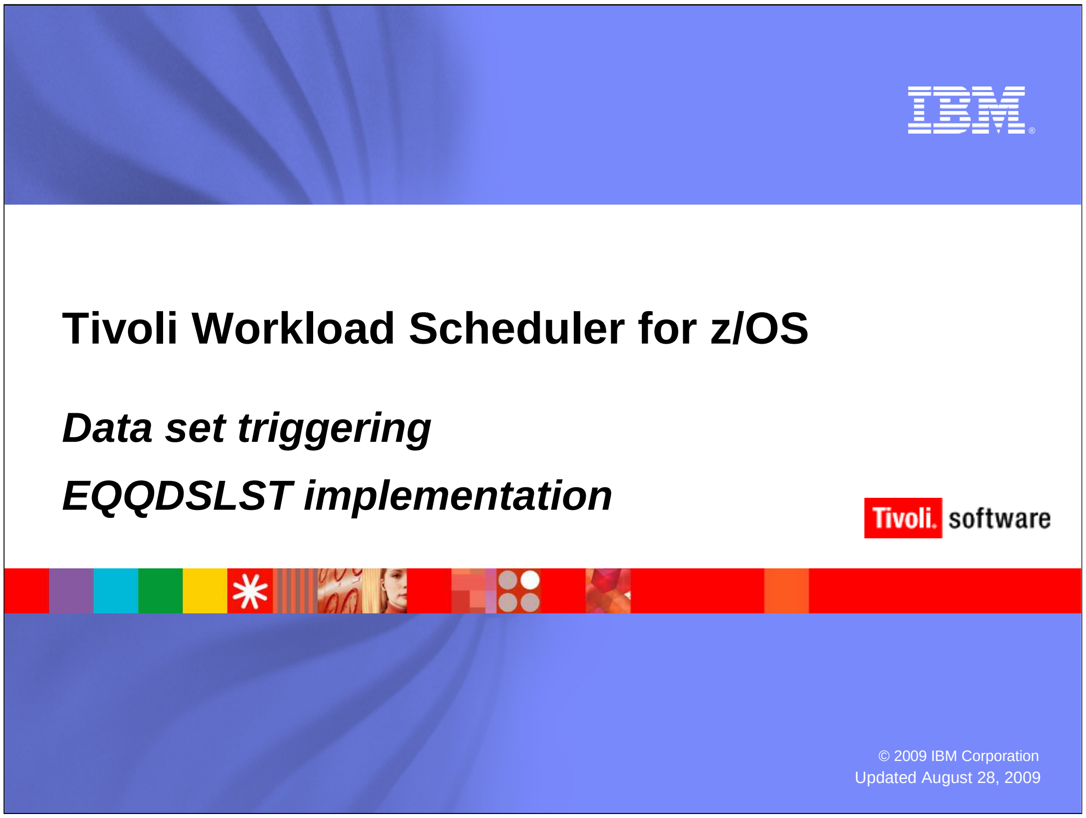


Tivoli Workload Scheduler for z/OS

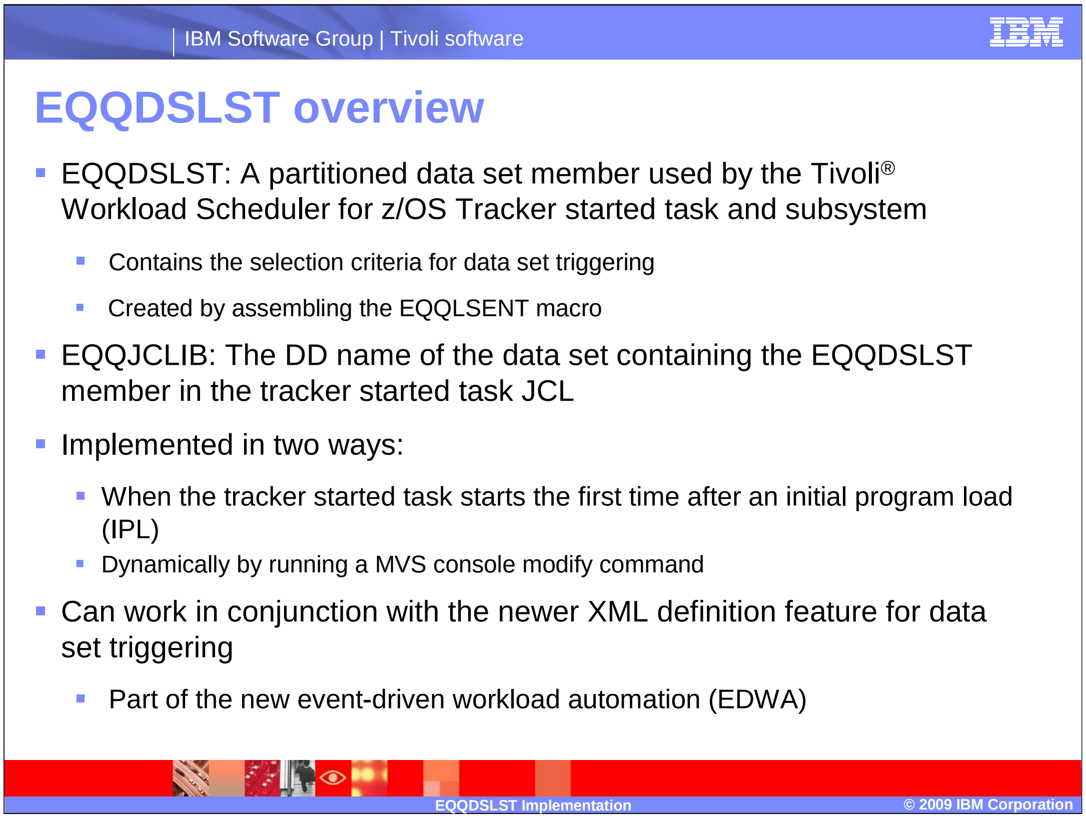
Data set triggering

EQQDSLST implementation

Updated August 28, 2009

EQQDSLST Implementation.

This training module provides the student with the knowledge required to implement the Tivoli Workload Scheduler for z/OS® provided EQQLSENT macro and EQQDSLST member for data set triggering. This training module assumes that a proper SMF setup has been performed. See the training modules *Data set triggering Overview* and *SMF Setup* for more details.



IBM Software Group | Tivoli software

EQQDSLST overview

- EQQDSLST: A partitioned data set member used by the Tivoli® Workload Scheduler for z/OS Tracker started task and subsystem
  - Contains the selection criteria for data set triggering
  - Created by assembling the EQQLSENT macro
- EQQJCLIB: The DD name of the data set containing the EQQDSLST member in the tracker started task JCL
- Implemented in two ways:
  - When the tracker started task starts the first time after an initial program load (IPL)
  - Dynamically by running a MVS console modify command
- Can work in conjunction with the newer XML definition feature for data set triggering
  - Part of the new event-driven workload automation (EDWA)

EQQDSLST Implementation

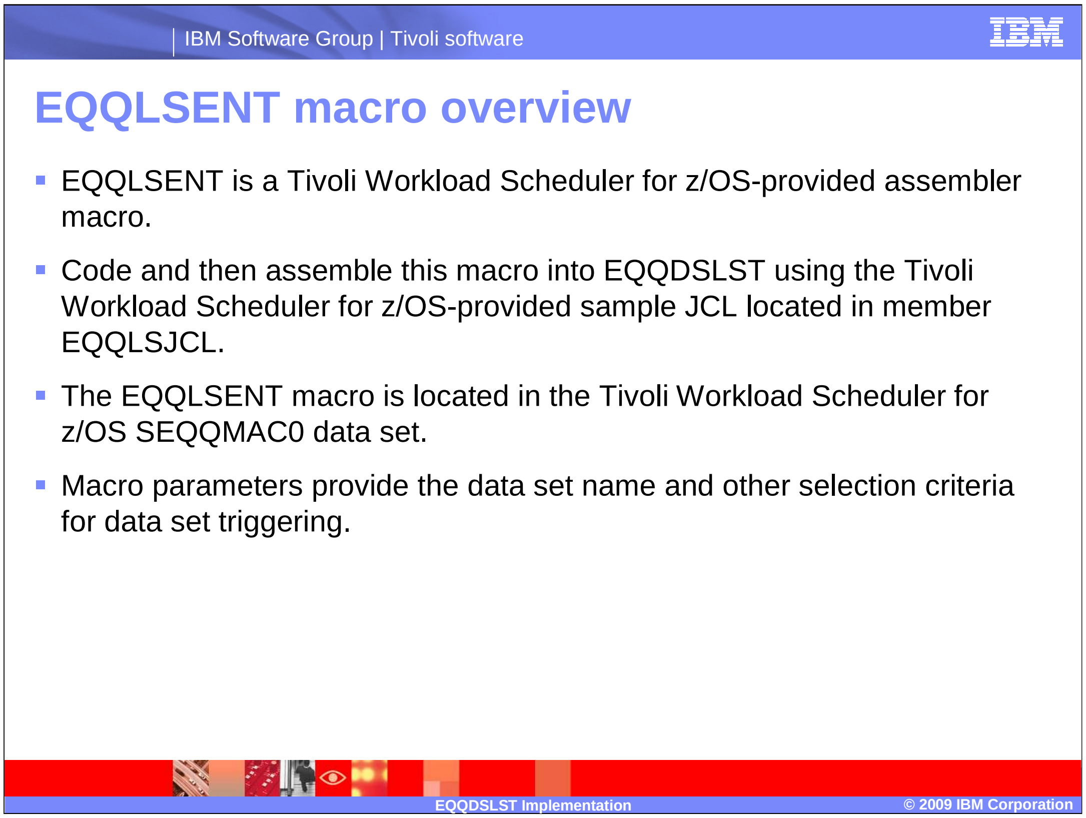
© 2009 IBM Corporation

EQQDSLST Overview.

EQQDSLST is the name of a partitioned data set member used by the Tivoli Workload Scheduler for z/OS tracker started task. The tracker subsystem will use the data in EQQDSLST after it has been loaded into ECSA. EQQDSLST contains the selection criteria for data set triggering, which you create by assembling the EQQLSENT macro. The selection criteria consists mainly of the data set name that is to be used as the trigger. The DD name of the data set containing the EQQDSLST member in the tracker started task JCL is EQQJCLIB.

EQQDSLST is implemented the first time the tracker started task is started after an IPL. Subsequent stops and re-starts of the tracker started task in between IPLs will not implement updated EQQDSLST members. However, updated EQQDSLST members can be implemented dynamically in between IPLs using a MVS console modify command.

EQQDSLST can also work in conjunction with the newer XML definition feature for data set triggering, which uses the member name EQQEVLST in the EQQJCLIB data set. If they are both present, the two members are merged, and EQQEVLST is used first. This newer method using XML is part of the new event-driven workload automation (EDWA) in Tivoli Workload Scheduler for z/OS 8.5.



IBM Software Group | Tivoli software

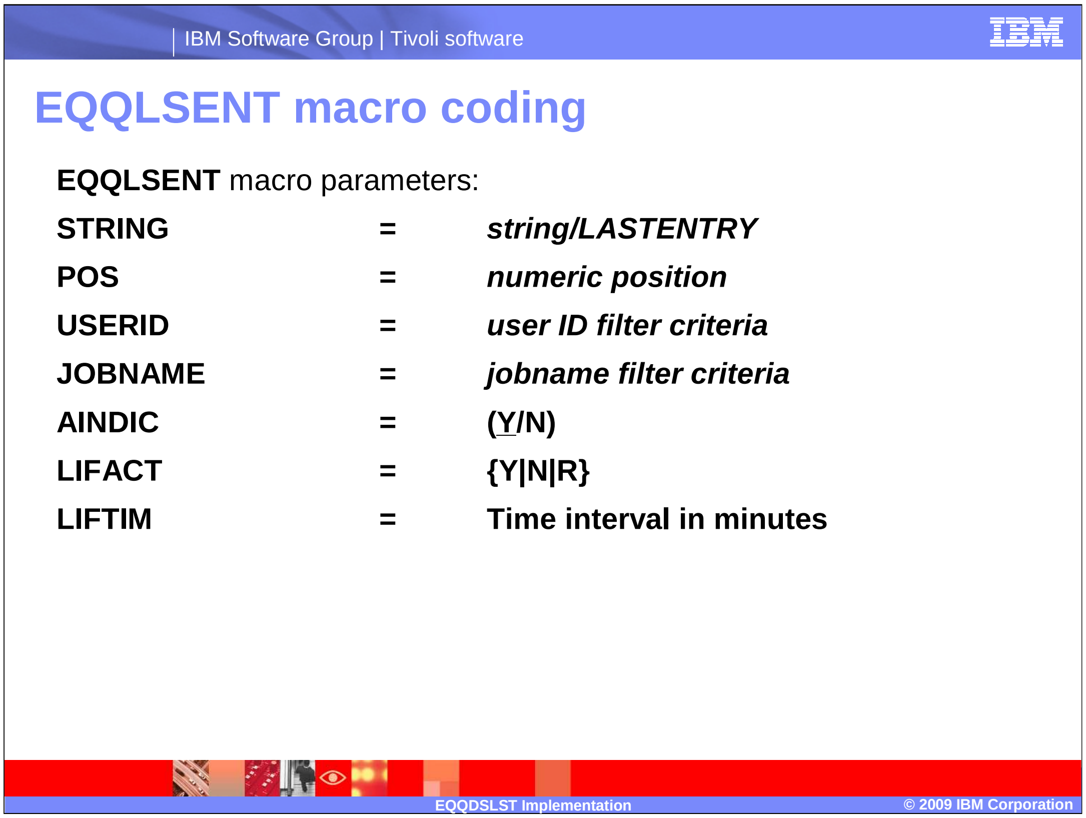
EQQLSENT macro overview

- EQQLSENT is a Tivoli Workload Scheduler for z/OS-provided assembler macro.
- Code and then assemble this macro into EQQDSLST using the Tivoli Workload Scheduler for z/OS-provided sample JCL located in member EQQLSJCL.
- The EQQLSENT macro is located in the Tivoli Workload Scheduler for z/OS SEQQMAC0 data set.
- Macro parameters provide the data set name and other selection criteria for data set triggering.

EQQDSLST Implementation

EQQLSENT Macro Overview.

EQQLSENT is a Tivoli Workload Scheduler for z/OS provided assembler macro. You must code and then assemble this macro into the EQQDSLST member of the tracker started task EQQJCLIB data set. The EQQLSENT macro is located in the Tivoli Workload Scheduler for z/OS SEQQMAC0 data set. The sample member EQQLSJCL in the Tivoli Workload Scheduler for z/OS SEQQSAMP data set can be used to assemble one or more EQQLSENT macros. The macro parameters provide the data set name and other selection criteria for data sets that will produce data set triggering special resource events when the data set is closed.



IBM Software Group | Tivoli software

EQQLSENT macro coding

EQQLSENT macro parameters:

STRING	**=**	***string/LASTENTRY***
POS	**=**	***numeric position***
USERID	**=**	***user ID filter criteria***
JOBNAME	**=**	***jobname filter criteria***
AINDIC	**=**	**(Y/N)**
LIFACT	**=**	**{Y|N|R}**
LIFTIM	**=**	**Time interval in minutes**

EQQDSLST Implementation

© 2009 IBM Corporation

EQQLSENT macro coding.

The EQQLSENT macro has several parameters that are shown on this slide. You can find a detailed description of the EQQLSENT macro and the parameters in Appendix E of the *IBM Tivoli Workload Scheduler for z/OS Installation* manual.

The STRING=string|LASTENTRY is a required keyword specifying the character string to be searched for. The string can be from 1 to 44 characters long and is used to represent all or part of the data set name. A single macro definition can identify several data sets as eligible for data set triggering. Any data set name that matches the string provided beginning at the starting position specified is considered a match. Also, to identify a fully qualified data set name including the complete last qualifier, add a space as the last character and enclose the string in single quotation marks. LASTENTRY should be coded for the last EQQLSENT macro only. It is the required value for last EQQLSENT macro specified in the SYSIN for the job that assembles the macros.

The POS=numeric position is a required keyword specifying the numeric position within the SMF record where the search string begins.

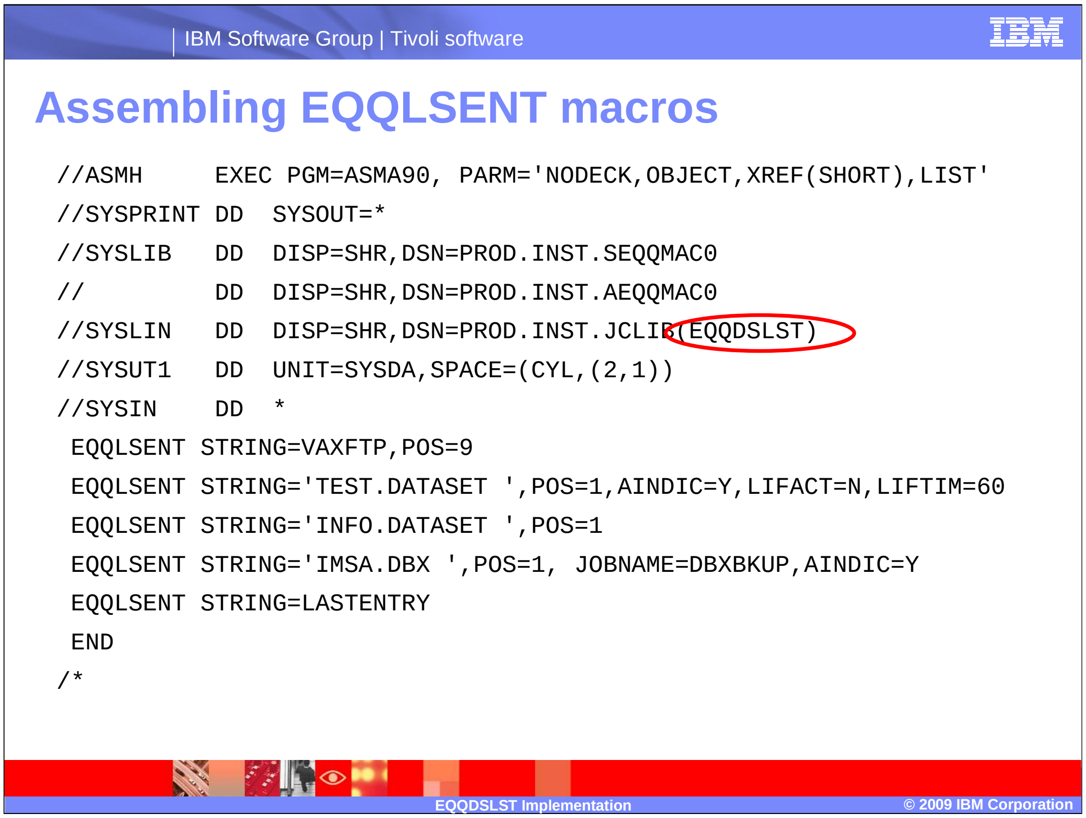
The USERID=string is an optional keyword specifying a generic character string to be compared with the SMFxxUID field, which contains the user identification associated with the job, started task, or time-sharing option (TSO) user that requested the activity against the data set that resulted in the data set closing. The string can be from one to eight characters long.

The JOBNAME=string is an optional keyword. It specifies a generic character string to be compared with the SMF14JBN, SMF15JBN, or SMF64JMN field, which contains the name of the job, started task, or TSO user that requested the activity against the data set that resulted in the data set closing. The string can be from one to eight characters long. If the data set is to be processed by FTP, JOBNAME corresponds to the user ID under which the dataset is received.

The AINDIC={Y|N} is an optional keyword specifying that the special resource is considered either available (Y) or unavailable (N) when the data set closing event occurs. The default is that the special resource is available once the data set close event trigger occurs.

The LIFACT={Y|N|R} is an optional keyword specifying the value to which the global availability of the special resource is reset, after the interval of time specified by the LIFTIM has expired. The allowed values are: *Y*, which sets the global availability to Yes; *N*, which sets the global availability to No; *R,* which sets the global availability to blank. This keyword is valid only if the LIFTIM parameter is specified. The default is R.

The LIFTIM=interval is an optional keyword specifying the interval of time, in minutes, after which the global availability of the special resource is reset to the value specified by LIFACT. The allowed range is from 1 to 999999. The time interval starts after the trigger has occurred when the data set closing event is created.



IBM Software Group | Tivoli software

Assembling EQQLSENT macros

```
//ASMH     EXEC PGM=ASMA90, PARM='NODECK,OBJECT,XREF(SHORT),LIST'
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DISP=SHR,DSN=PROD.INST.SEQQMAC0
//         DD  DISP=SHR,DSN=PROD.INST.AEQQMAC0
//SYSLIN   DD  DISP=SHR,DSN=PROD.INST.JCLIB(EQQDSLST)
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSIN    DD  *
 EQQLSENT STRING=VAXFTP,POS=9
 EQQLSENT STRING='TEST.DATASET ',POS=1,AINDIC=Y,LIFACT=N,LIFTIM=60
 EQQLSENT STRING='INFO.DATASET ',POS=1
 EQQLSENT STRING='IMSA.DBX ',POS=1, JOBNAME=DBXBKUP,AINDIC=Y
 EQQLSENT STRING=LASTENTRY
 END
/*
```

EQQDSLST Implementation

© 2009 IBM Corporation

Assembling EQQLSENT macros.

The slide shows an excerpt from the EQQLSJCL member of SEQQSAMP. You can use this job to assemble the data set triggering selection table EQQDSLST. It is a one-step job. You must add a valid jobcard and replace the data set names with names that are valid for your environment.

This sample selection table will create special resource events when:

For the first macro coding, a data set having VAXFPT in positions 9-14 of the data set name is closed.

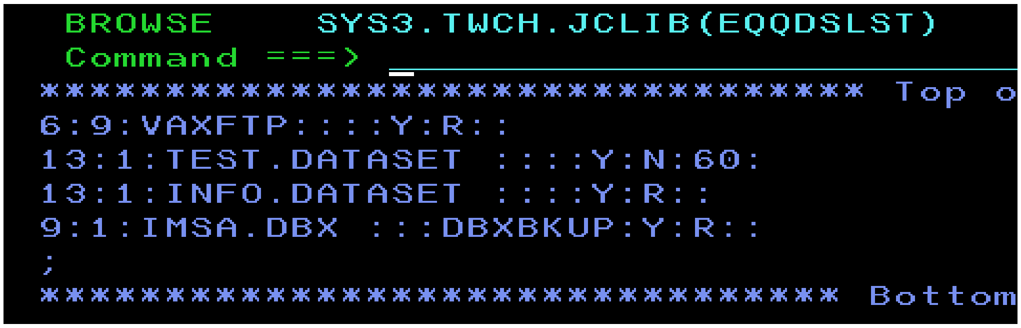
For the second macro coding, a data set named TEST.DATASET is closed. The special resource event created for the TEST.DATA data set has an expiration time of 60 minutes because LIFTIM=60. Because LIFACT=N was specified, the special resource availability for TEST.DATA is set to no after the 60-minute expiration.

For the third macro coding, a data set trigger occurs after a data set named INFO.DATASET is closed.

For the fourth macro coding, a data set named IMSA.DBX is closed and the name of the job or started task closing it is named DBXBKUP.

The LASTENTRY macro coding creates an end to the selection table. The last EQQLSENT macro in this example specifies LASTENTRY and is required. If the entry is not the last entry in the SYSIN, the member created is truncated when loaded into ECSA.

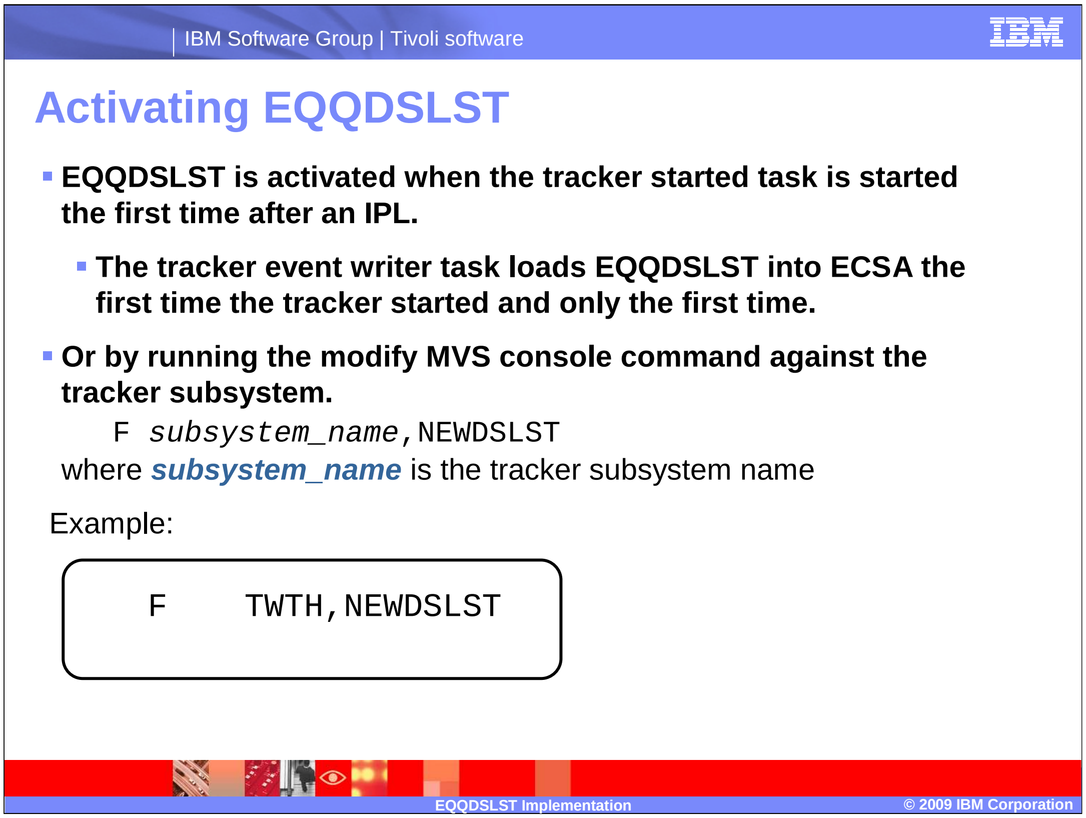
EQQDSLST contents



```
  BROWSE     SYS3.TWCH.JCLIB(EQQDSLST)
  Command ===>
********************************* Top o
6:9:VAXFTP::::Y:R::
13:1:TEST.DATASET ::::Y:N:60:
13:1:INFO.DATASET ::::Y:R::
9:1:IMSA.DBX :::DBXBKUP:Y:R::
;
******************************** Bottom
```

EQQDSLST contents.

The slide shows the contents of the EQQDSLST member after running the sample EQQLSJCL. Four lines correspond to the four EQQLSENT macros that were coded with data set triggering selection criteria. The last line, which contains a semicolon, corresponds to the EQQLSENT macro coded as LASTENTRY.



IBM Software Group | Tivoli software

Activating EQQDSLST

- **EQQDSLST is activated when the tracker started task is started the first time after an IPL.**
  - **The tracker event writer task loads EQQDSLST into ECSA the first time the tracker started and only the first time.**
- **Or by running the modify MVS console command against the tracker subsystem.**

```
   F subsystem_name,NEWDSLST
```

where ***subsystem_name*** is the tracker subsystem name

Example:

```
   F    TWTH,NEWDSLST
```

EQQDSLST Implementation

Activating EQQDSLST.

Consider that you want to trigger a data set close special resource event within a specific Tivoli Workload Scheduler for z/OS tracker subsystem using the EQQDSLST member. In that case, the tracker started task for that tracker subsystem must have access to the EQQDSLST member in the EQQJCLIB data set. This is because the tracker started task event writer task must load the contents of the current EQQDSLST member into ECSA.

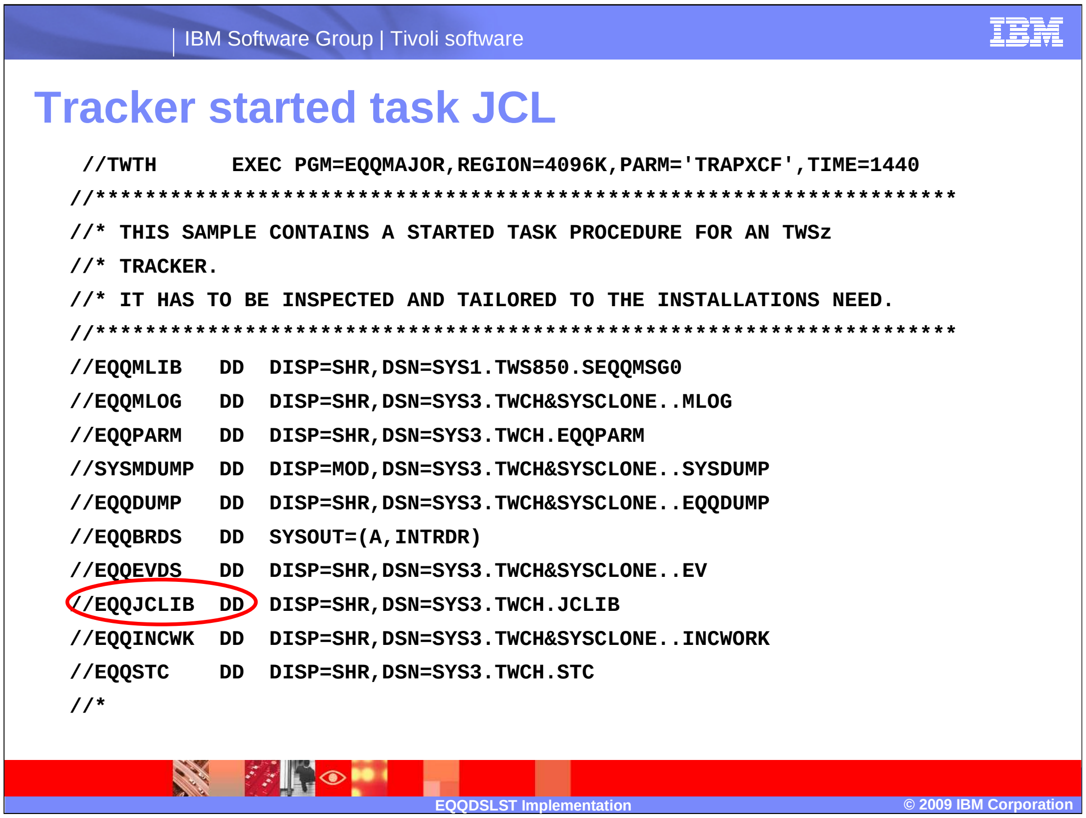
The tracker started task event writer task loads the contents of the current EQQDSLST member into ECSA at one of two times:

When the tracker started task is started for the first time after an IPL or

When an MVS console modify command that specifies NEWDSLST is issued for a specific tracker subsystem.

This slide shows an example of a MVS **modify** command that requests the tracker susbsystem TWTH load and use the current contents of the EQQDSLST member.

It is important to note that automatic loading of EQQDSLST into ECSA only occurs the first time a tracker started task is started after an IPL. Subsequent stops and starts of the tracker started task in between IPLs will not load EQQDSLST into ECSA.

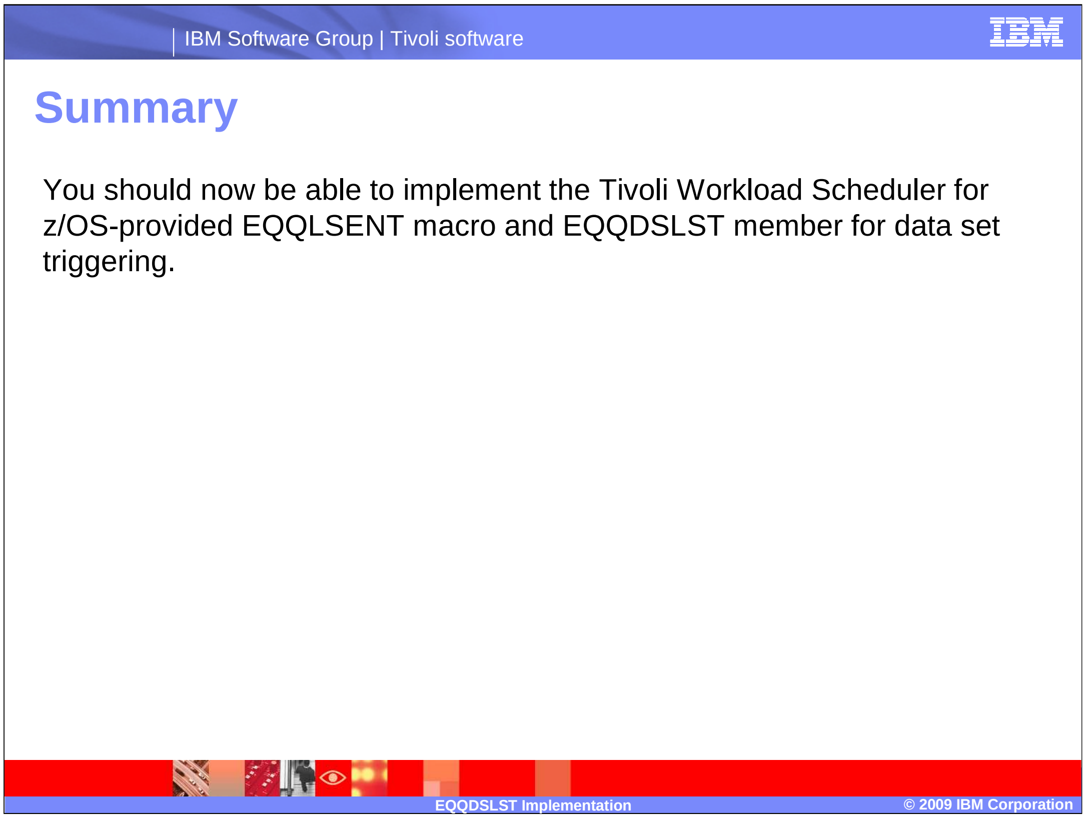


Tracker started task JCL

```
 //TWTH      EXEC PGM=EQQMAJOR,REGION=4096K,PARM='TRAPXCF',TIME=1440
//*********************************************************************
//* THIS SAMPLE CONTAINS A STARTED TASK PROCEDURE FOR AN TWSz
//* TRACKER.
//* IT HAS TO BE INSPECTED AND TAILORED TO THE INSTALLATIONS NEED.
//*********************************************************************
//EQQMLIB   DD  DISP=SHR,DSN=SYS1.TWS850.SEQQMSG0
//EQQMLOG   DD  DISP=SHR,DSN=SYS3.TWCH&SYSCLONE..MLOG
//EQQPARM   DD  DISP=SHR,DSN=SYS3.TWCH.EQQPARM
//SYSMDUMP  DD  DISP=MOD,DSN=SYS3.TWCH&SYSCLONE..SYSDUMP
//EQQDUMP   DD  DISP=SHR,DSN=SYS3.TWCH&SYSCLONE..EQQDUMP
//EQQBRDS   DD  SYSOUT=(A,INTRDR)
//EQQEVDS   DD  DISP=SHR,DSN=SYS3.TWCH&SYSCLONE..EV
//EQQJCLIB  DD  DISP=SHR,DSN=SYS3.TWCH.JCLIB
//EQQINCWK  DD  DISP=SHR,DSN=SYS3.TWCH&SYSCLONE..INCWORK
//EQQSTC    DD  DISP=SHR,DSN=SYS3.TWCH.STC
//*
```

Tracker started task JCL.

This slide shows an example of a tracker started task JCL and the EQQJCLIB DD statement that is required to support data set triggering. The EQQJCLIB data set is a partitioned data set that contains the EQQDSLST member. It will also contain the EQQEVLST member, if XML is being used to define data set selection criteria for data set triggering. See the *XML Implementation* training module for details.

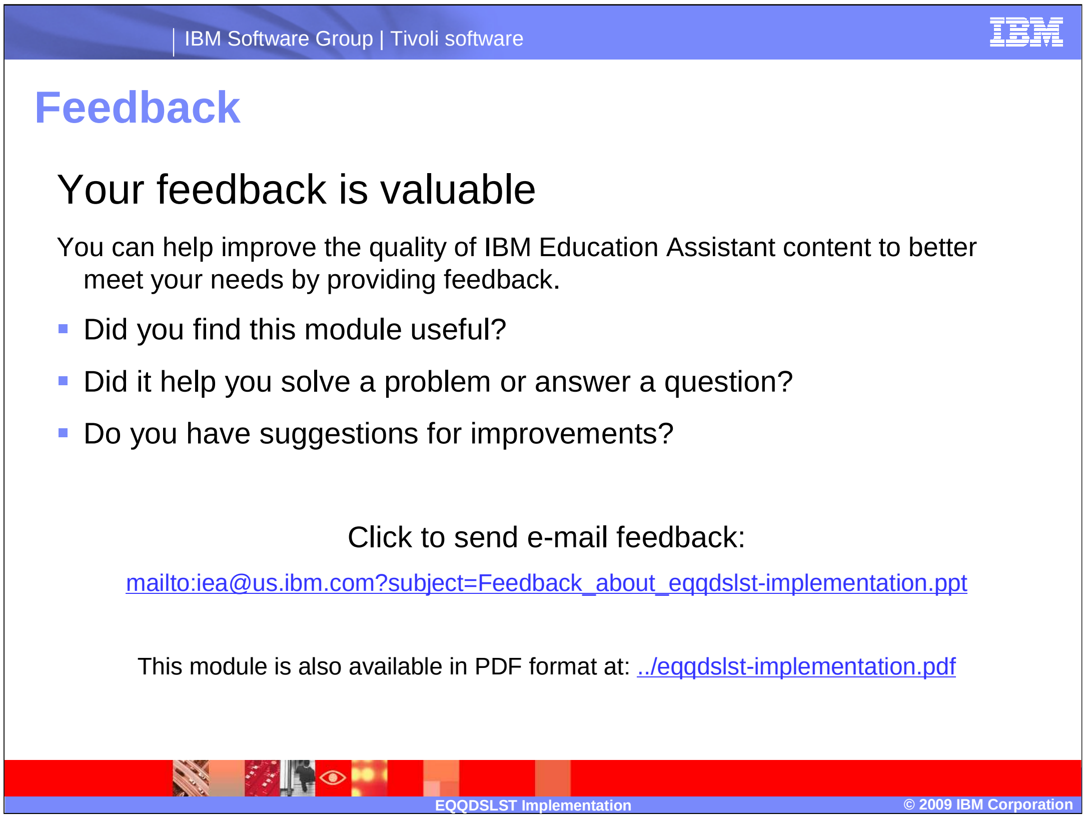


Summary

You should now be able to implement the Tivoli Workload Scheduler for z/OS-provided EQQLSENT macro and EQQDSLST member for data set triggering.

Summary.

In summary you should now be able to implement the Tivoli Workload Scheduler for z/OS provided EQQLSENT macro and EQQDSLST member for 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_eqqdslst-implementation.ppt

This module is also available in PDF format at: ../eqqdslst-implementation.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 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.