

IWS Administration Training

Odb215318
Global Knowledge

Objective

In this training, you will learn about the architecture, configuration and installation of IWS for z/OS.

After completing this training, you should be able to:

- Configure IWS for z/OS
- Manage IWS for z/OS
- Migrate one version to another
- Use the tools

Planning of the course

DAY 1

- IWS ARCHITECTURE
- INSTALLATION
- SYSTEM

DAY 2

- IWS CONFIGURATION
- BACKUP AND DRP
- SECURITY
- UPGRADE of IWS version

DAY 3

- IWS TOOLS
- WAPL

Global2153 Knowledge

Architecture

- Architecture
- Component
 - Controller
 - Tracker
 - Datastore
 - End to End
 - Output Collector
 - TCP/IP server
 - z/OS connector
- Z/OS system architecture
- Files
- Configuration sample
- Planning

Odb2155318
Global Knowledge

Installation and System

- **EQQJOBS**
- **z/OS parameter**

Odb2155318
Global Knowledge

CONFIGURATION

- ALL PARAMETERS
- PARM per ITEMS
 - Configuration
 - Security
 - Job success or failure
 - Restart and Cleanup
 - Job log retrieval
 - Automatic Job recovery
 - Workload restart
 - Performance
 - Reporting
 - Output Processing
 - End to End
 - WLM
 - External Monitoring

CONFIGURATION SAMPLE

- A controller and a tracker connected via XCF
- A controller and a tracker connected via TCPIP
- A controller and a datastore connected via XCF
- A z/OS SYSPLEX
- An Hybrid architecture between IWS for z/OS and IWS Distributed
- z/Centric - z/OS Agent
- DWC

DRP

- ALL description file
- How are used the file
- Current plan turnover process
- What to do after a 9.5

Odlb2155318
Global Knowledge

SECURITY

- Setup security in IWS
- Manage security in RACF
- Manage security on LDAP

Odlb2155318
Global Knowledge

MIGRATION

- Auditing actual IWS
- Setup the new JOBLIB
- Use the alias dataset
- Verify the File (SIZE, recordsize, DCB)
- Install STEPLIB tools for TSO/E
- Installing toleration PTF
- Generate all TWS job using the ISPF Panels
- Create Parameter member for all the IWS utilities
- Modify the STC and system PARAMETER for the new release
- Create all Migration JOB
- Prepare and test the backout on test environment

TOOLS

- PIF
- EQQYCAIN (Batch Command Interface Tool)
- OCL
- EQQEVPGM
- VARIABLES
- OPC RECOVER
- JCL DIRECTIVE
- WAPL (replace SOE)

Odb2155318
Global Knowledge

IWS Architecture

Odb2155318
Global Knowledge

Objective

In this chapter , you will learn about the architecture of IWS product

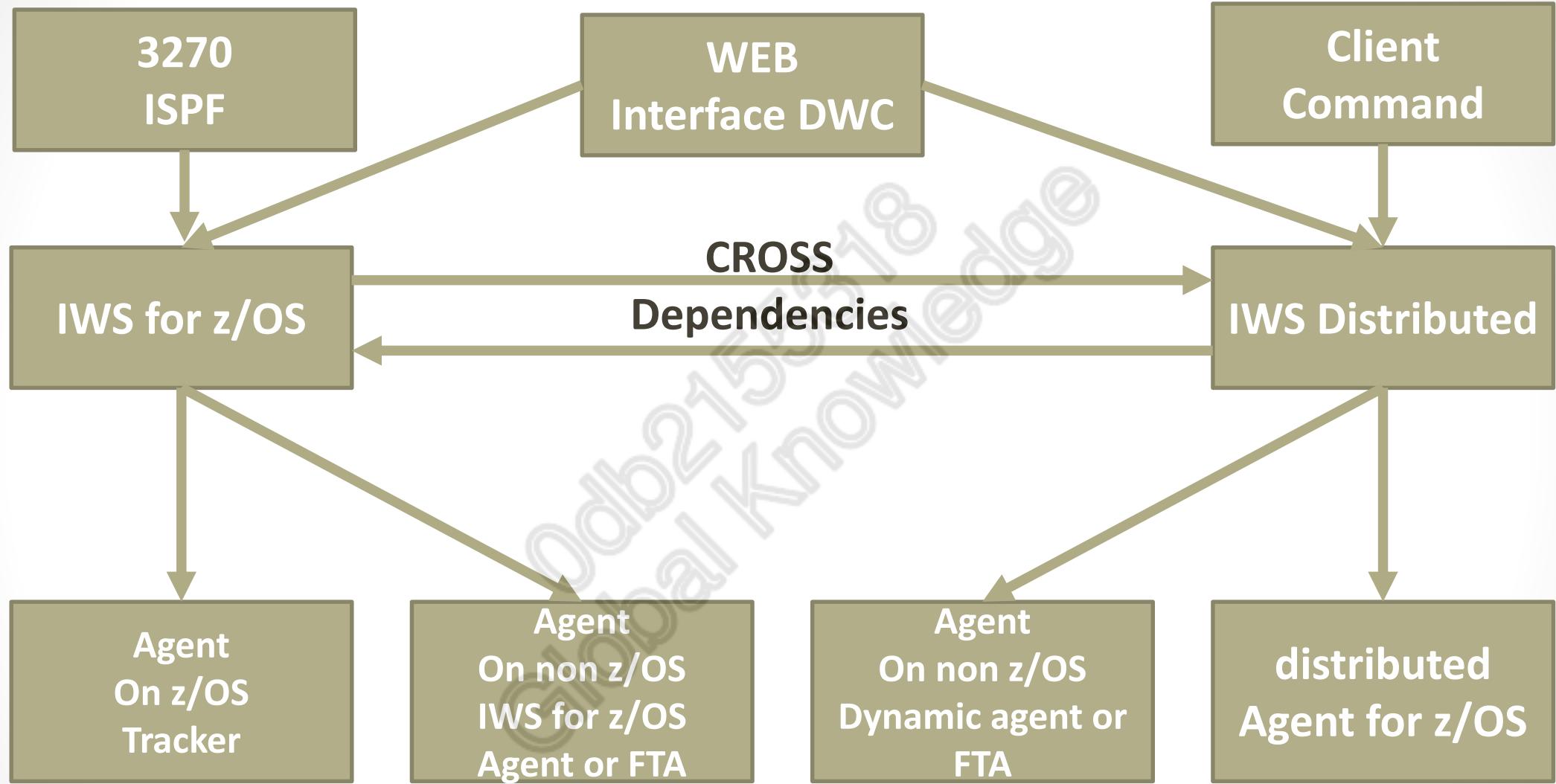
After completing this training, you should be able to:

Understand the IWS component

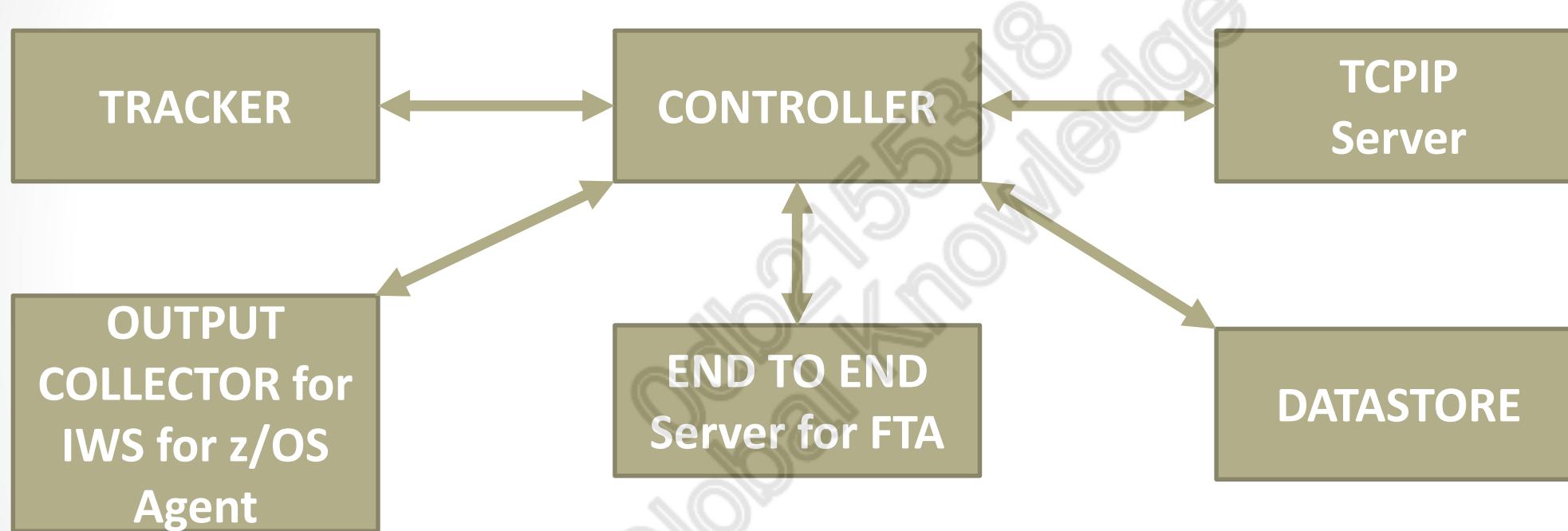
Planning process

File usage

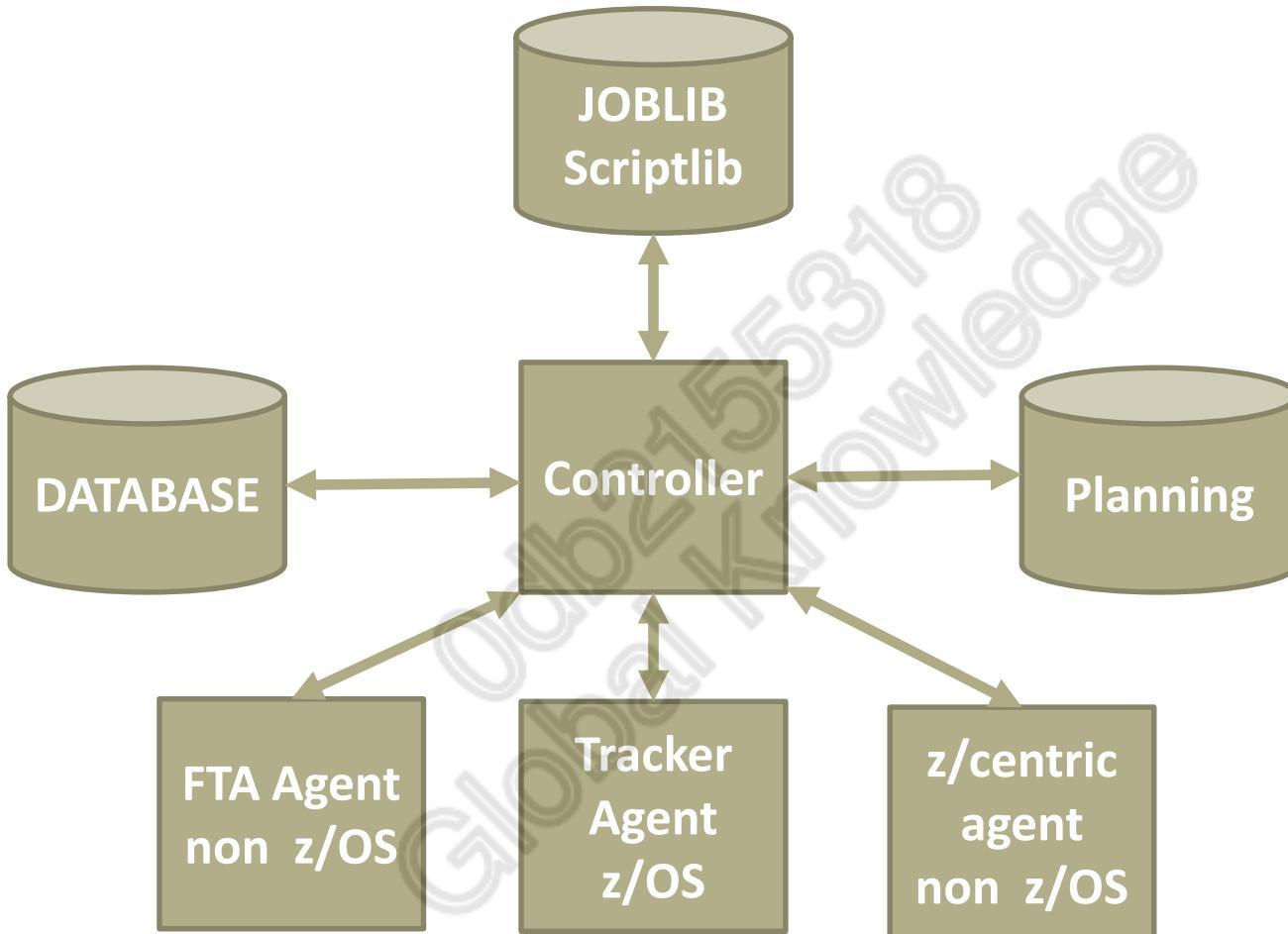
IWS Architecture



z/OS component



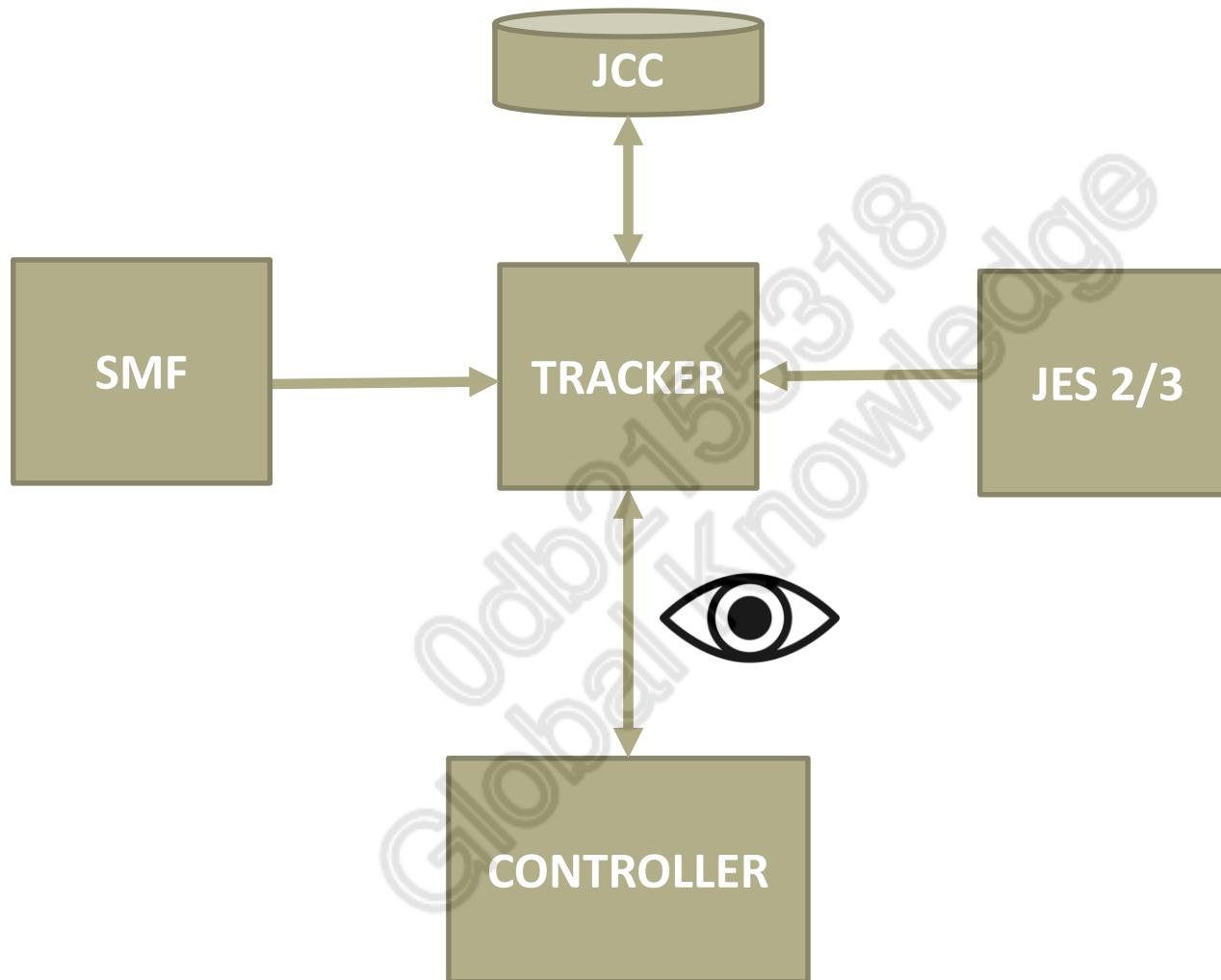
z/OS Architecture : Controller



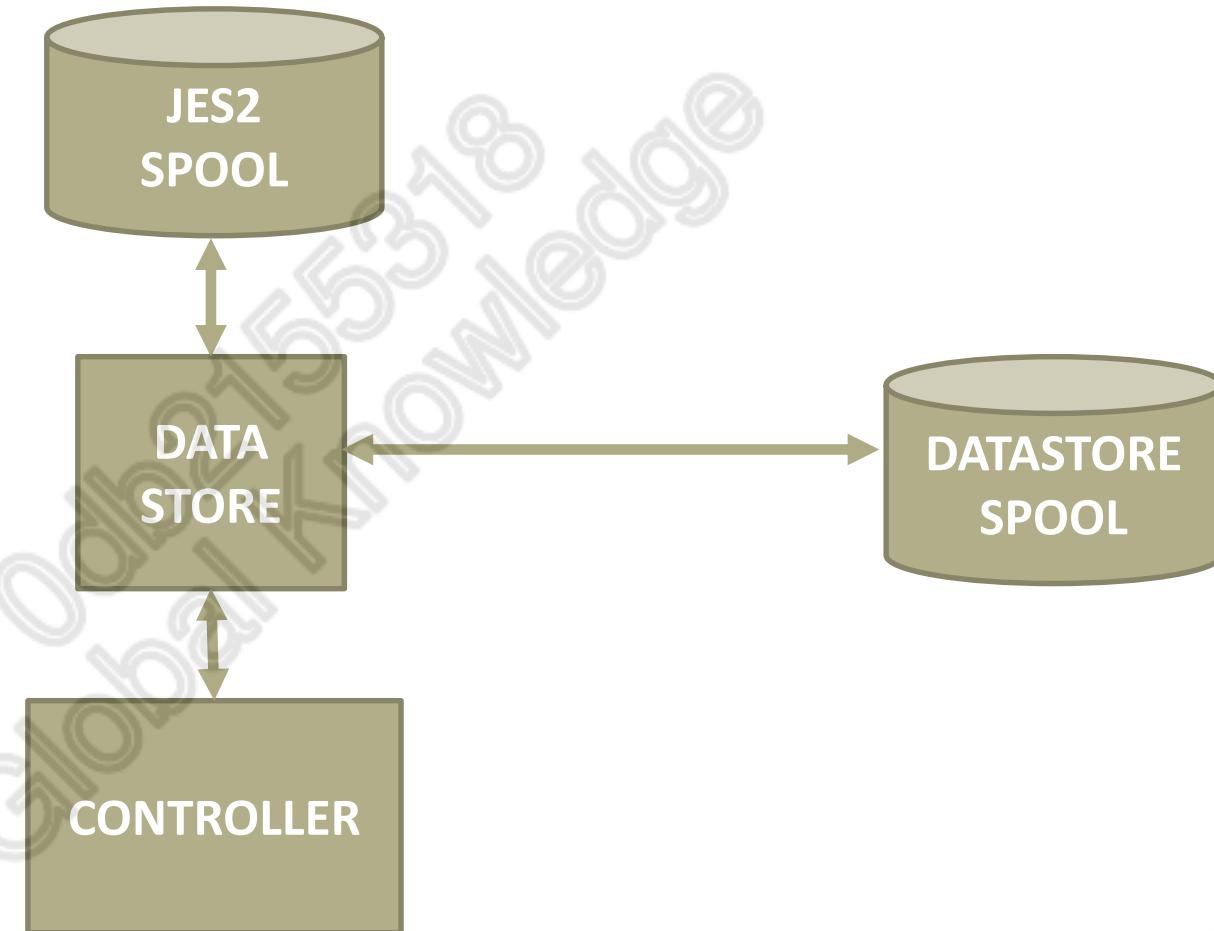
FTA / zcentric

- Linux RHEL or SUSE
- AIX
- Windows
- HP
- Solaris
- Ubuntu
- Centos
- AS/400 (IBM I)
- z/linux

z/OS Architecture : TRACKER

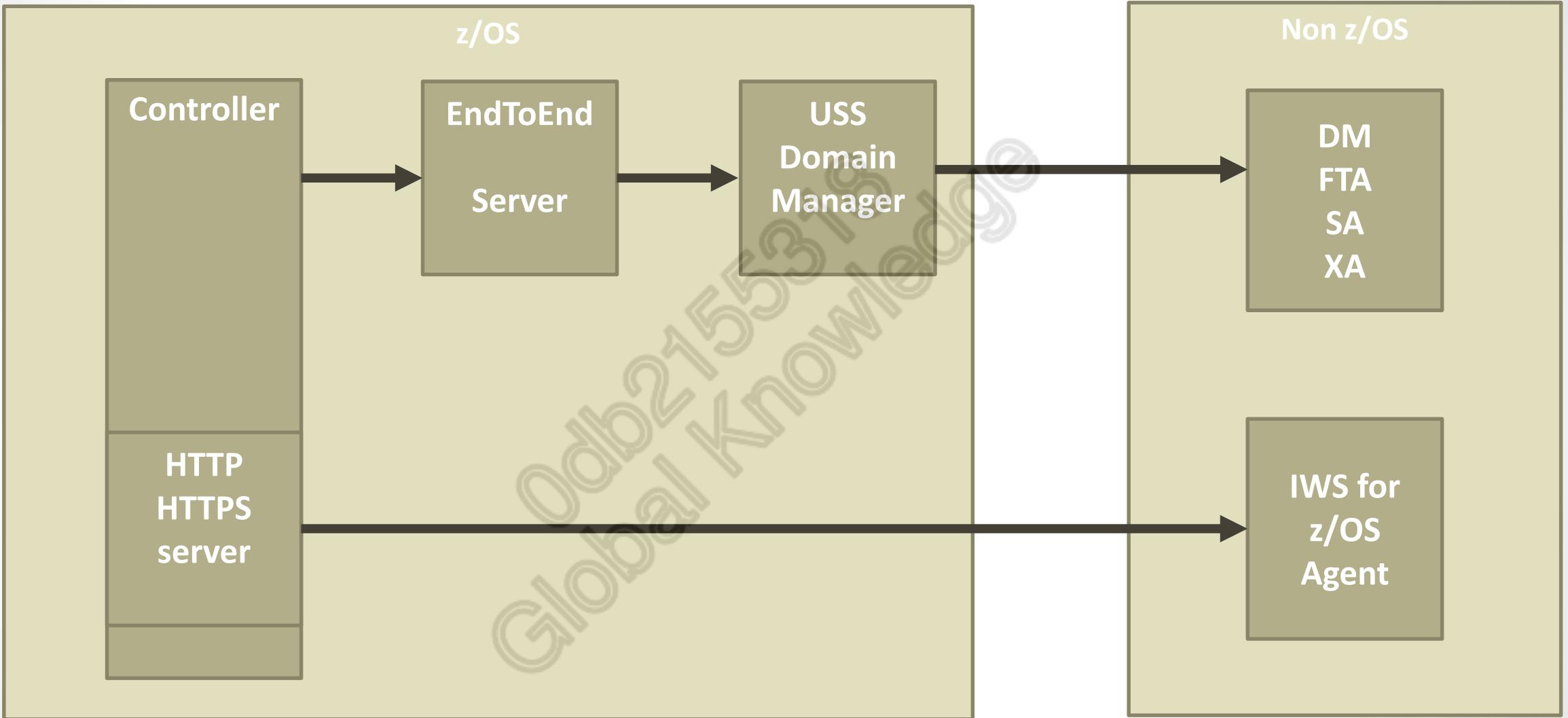


z/OS Architecture : DATASTORE

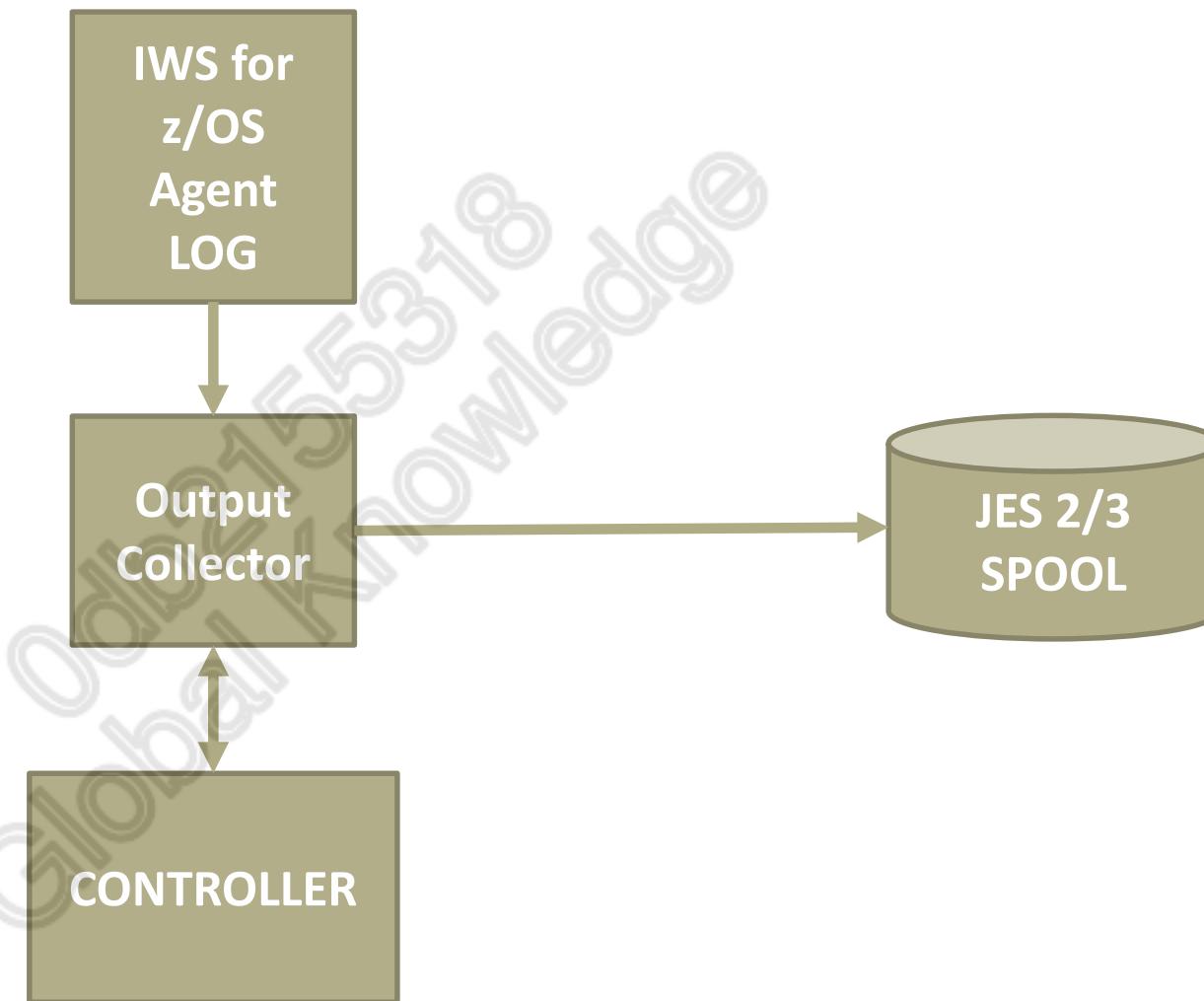


- Restart And Cleanup
- Dataset cleanup
- JOBLOG Retrieval

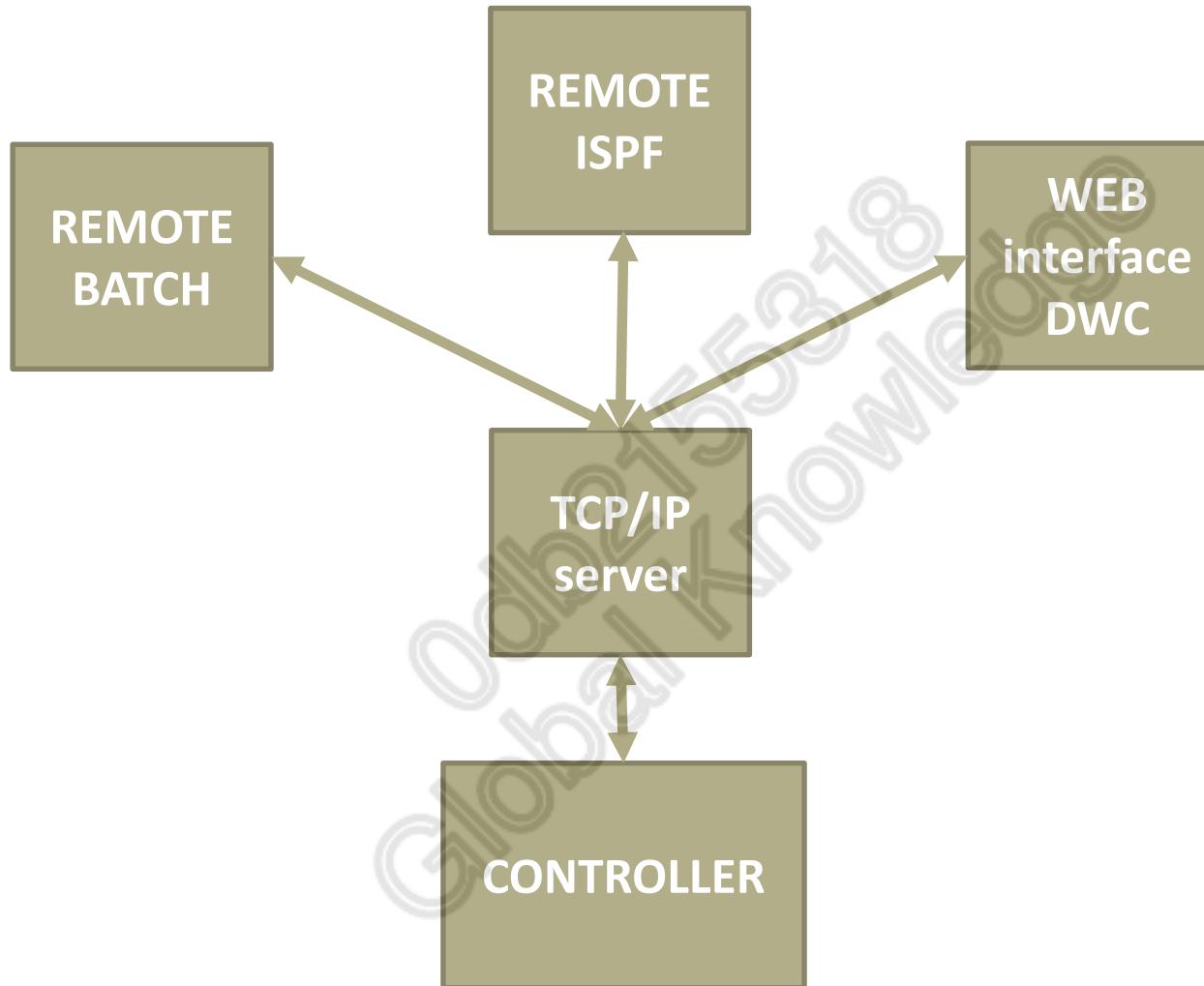
z/OS Architecture : End to End



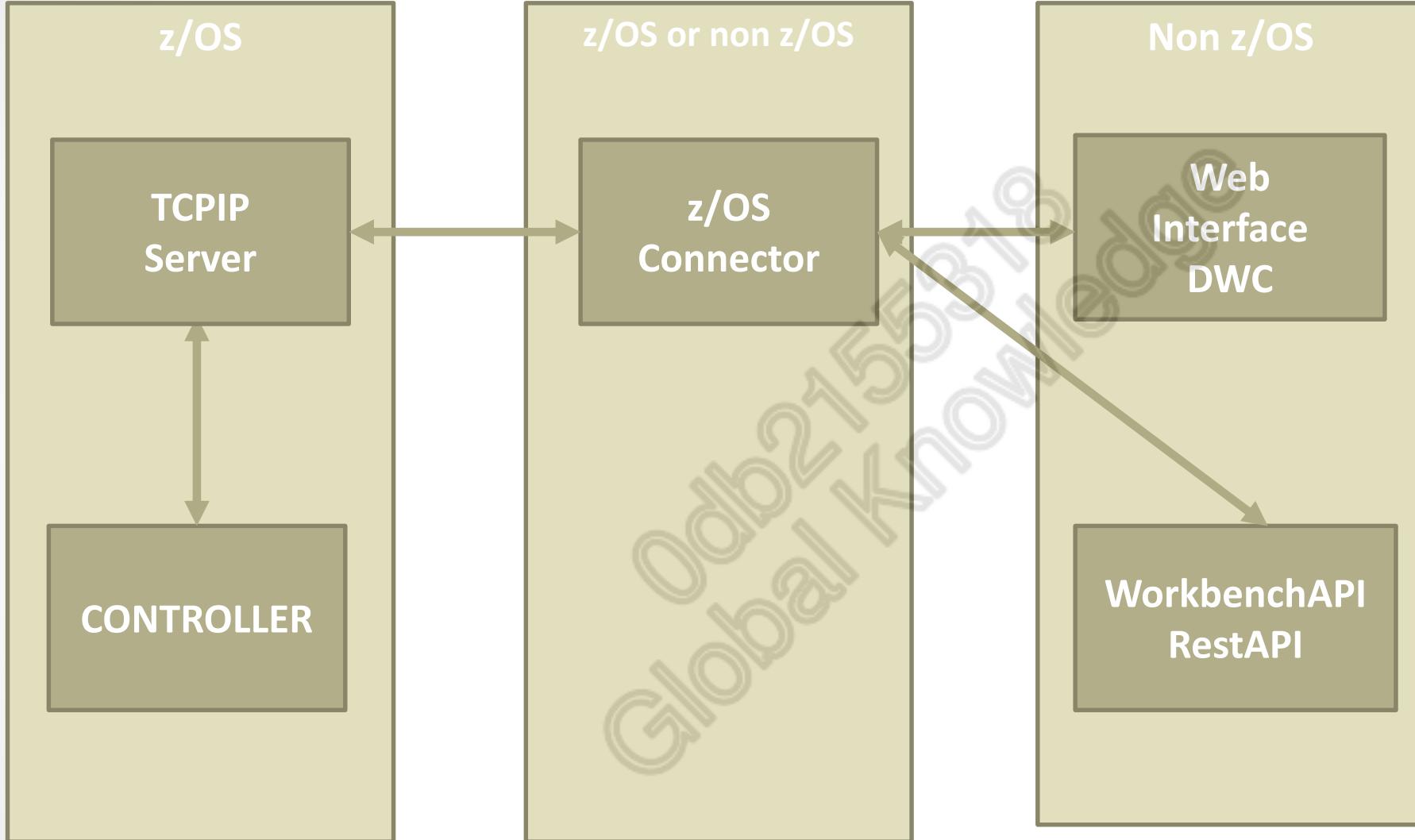
z/OS Architecture : Output Collector



z/OS Architecture : TCP/IP server

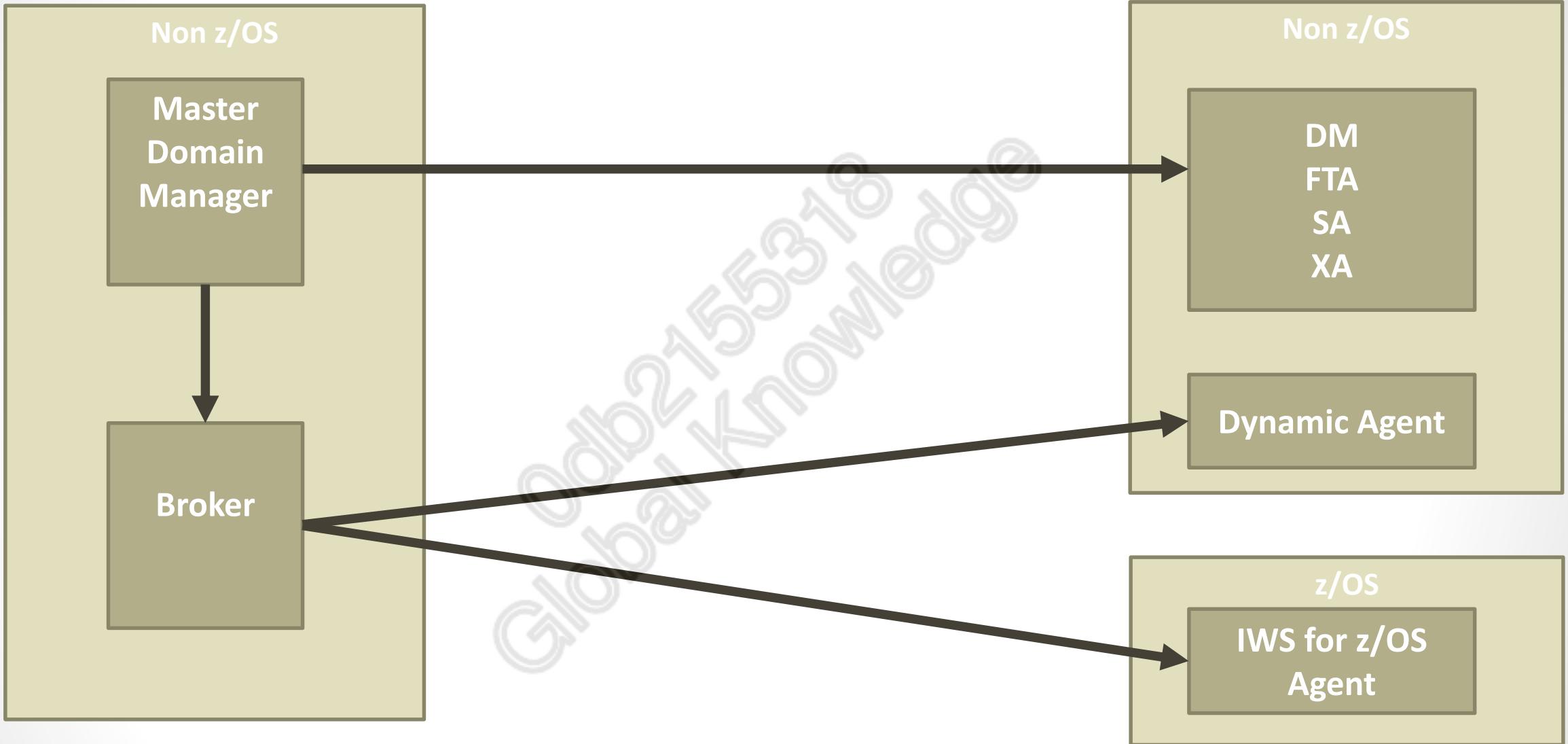


z/OS Connector Architecture



- Linux RHEL or SUSE
- AIX
- Windows
- z/Linux

Distributed Architecture with z/OS



z/OS Architecture : SYSTEM



- IEFSSNnn for subsystem Controller and Tracker
- IKJTSOnn for TSO/E authorization
- PROGnn for Library authorization
- SMFPRMnn for SMF parameter and Exit
- BPXPRMnn for Z/FS file
- JES2PRMnn for Exit parameter
- STC Controller, Tracker, Datastore, TCPIP server, EndToEnd server, Output collector
- PROC for different IWS service like EQQDELDS ...
- Update of Logon procedure to include IWS ISPF File



CONTROLLER

Odb2155318
Global Knowledge

z/OS Architecture : Controller File

- Database file
- Current Plan
- Long Term Plan
- End To End scheduling
- Event
- JCL Repository
- Job Tracking
- DataStore

Odb2155318
Global Knowledge

Controller File : Database

DATABASE	Physical VSAM Dataset	DD NAME
Application Description (1.4)	AD	EQQADD\$
Calendar (1.2)	WS	EQQWSDS
Job Descriptions (1.8)	AD	EQQADD\$
Operator Instruction (1.5)	OI	EQQOID\$
Period (1.3)	WS	EQQWSDS
Special resource (1.6)	RD	EQQRDDS
Side Information (ETT) (1.7)	SI	EQQSIDS
Variable Table (1.9)	AD	EQQADD\$
Workstation Description (1.1)	WS	EQQWSDS
Run Cycle Group (1.10)	WS	EQQWSDS

Controller File : Current Plan

Current Plan	Physical VSAM Dataset	DD NAME
Primary Current Plan	CP1	EQQCP1DS
Alternate Current Plan	CP2	EQQCP2DS
New Current Plan	NCP	EQQNCPDS
New Current Plan Extension	NCX	EQQNCXDS
Current plan Extension	CX	EQQCXDS
Primary extended Data	XD1	EQQXD1DS
Alternate extended data	XD2	EQQXD2DS
New Extended data	NXD	EQQNXDDS

Controller File : Long Term Plan

Long Term Plan	Physical VSAM Dataset	DD NAME
Long Term Plan	LT	EQQLTDS

Controller File : End To End scheduling

End To End scheduling	Physical Dataset	DD NAME
Centralized script dataset for end-to-end scheduling with fault tolerance capabilities	CS (PDS)	EQQTWSCS
Input Event data sets for end-to-end scheduling with fault tolerance capabilities	WSIN (SEQ)	EQQWSIN
Output Event data sets for end-to-end scheduling with fault tolerance capabilities	WSOUT (SEQ)	EQQWSOUT
Script library for end-to-end scheduling with fault tolerance capabilities	SCLIB (PDS)	EQQSCLIB
Current plan backup for the creation of Symphony	SCP (VSAM)	EQQSCPDS

Controller File : Event

Event	Physical VSAM Dataset	DD NAME
Event Log	EV	EQQEVD\$
Event for Event reader (01-16)	EVxx	EQQEVDxx

Controller File : JCL

JCL Repository	Physical VSAM Dataset	DD NAME
Primary JCL repository	JS1	EQQJS1DS
Secondary JCL repository	JS2	EQQJS2DS

- One JCL per record
- Default recordsize 180000 then 3120 lines
- Maximum recordsize 735480 then 9193 lines

Controller File : Job Tracking

JCL Repository	Physical VSAM Dataset	DD NAME
Job Tracking Log	JTxx	EQQJTxx
Dual Job Tracking Log	DLxx	EQQDLxx
JT archive data set	JTARC	EQQJTARC

Controller File : DataStore

JCL Repository	Physical VSAM Dataset	DD NAME
Structured data files	SDFxx	EQQSDFxx
Primary index	PKIxx	EQQPKIxx
Unstructured data files	UDFxx	EQQUDFxx
Secondary index	SKIxx	EQQSKIxx

- Structured data file is for JESJCL,JESYSMSG, JESMSGLG
- Unstructured data file is for SYSOUT

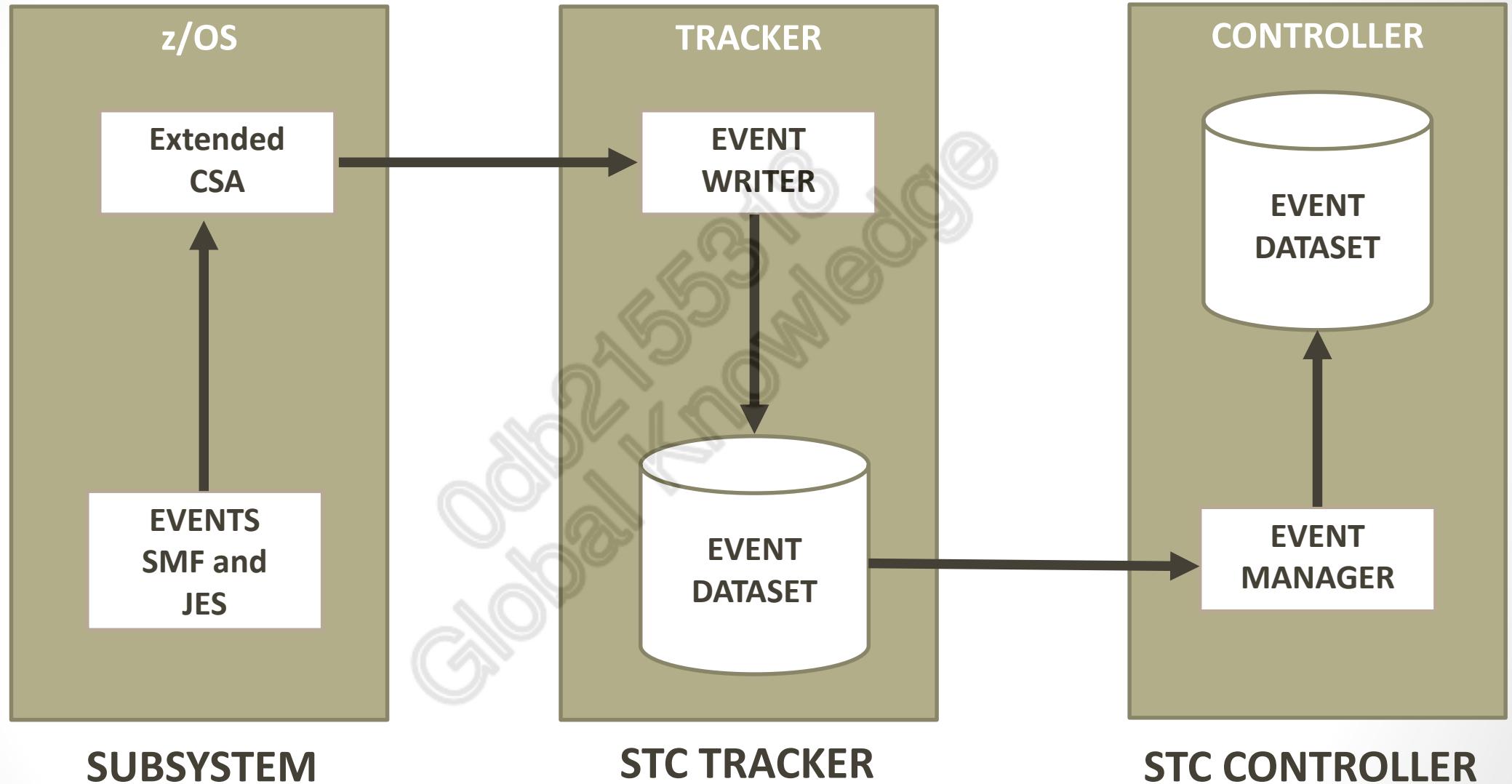
TRACKER

Odb2155318
Global Knowledge

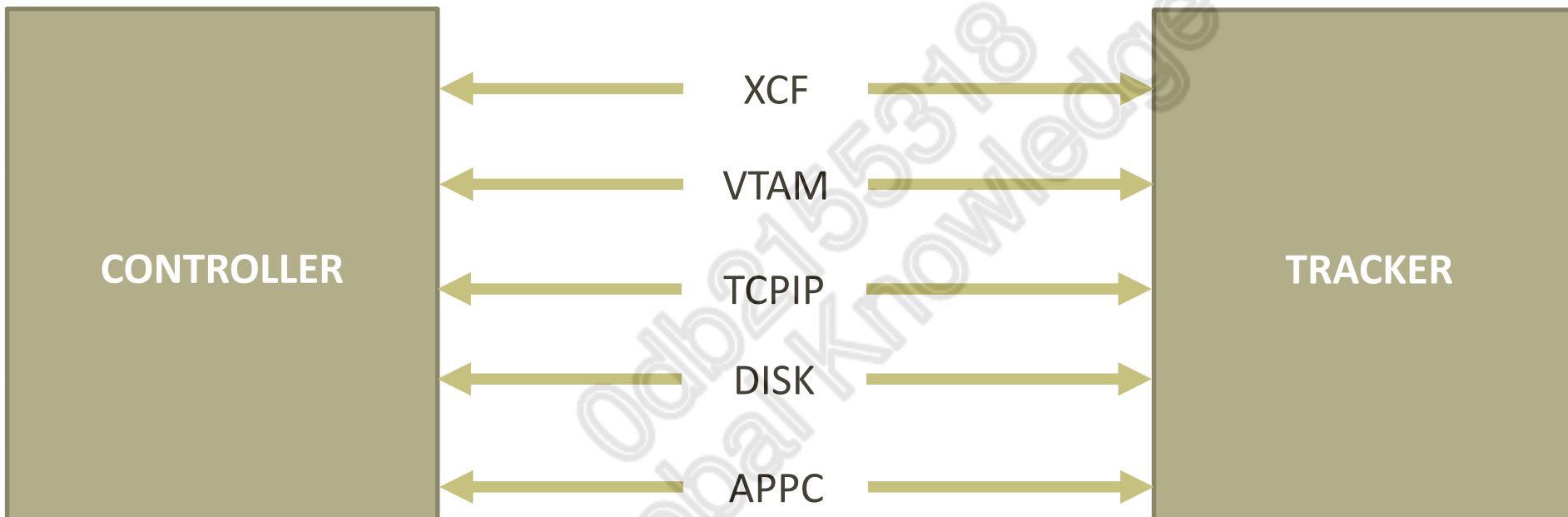
z/OS Architecture : Tracker File

JCL Repository	Physical VSAM Dataset	DD NAME
Internal Reader	BR	EQQBRDS
Event File	EV	EQQEVD\$
Submit Release	SU	EQQSUD\$
JCLIB for JCC definition	JC	EQQJCLIB
Incident work for JCC	INCWK	EQQINCWK
STC Submission	STC	EQQSTC

Tracker Process



Controller / Tracker Connection



DATASTORE

Odb2155318
Global Knowledge

DATASTORE FILE

JCL Repository	Physical VSAM Dataset	DD NAME
Structured data files	SDFxx	EQQSDFxx
Primary index	PKIxx	EQQPKIxx
Unstructured data files	UDFxx	EQQUDFxx
Secondary index	SKIxx	EQQSKIxx

- Structured data file is for JESJCL, JESYSMSG, JESMSGLG
- Unstructured data file is for SYSOUT

END TO END

Odbl2155318
Global Knowledge

END TO END FILE

END TO END	Physical Dataset	DD NAME
QUEUE server to Controller	IN	EQQTWSIN
QUEUE Controller to server	OU	EQQTWSOU
Temporary store job from joblib	CS	EQQTWSCS

OUTPUT COLLECTOR

Odb2155318
Global Knowledge

OUTPUT COLLECTOR FILE

Output Collector	Physical Dataset	DD NAME
Stores event	OUCEV	EQQOUCHEV
Request checkpoint	OUCKP	EQQOUCKP

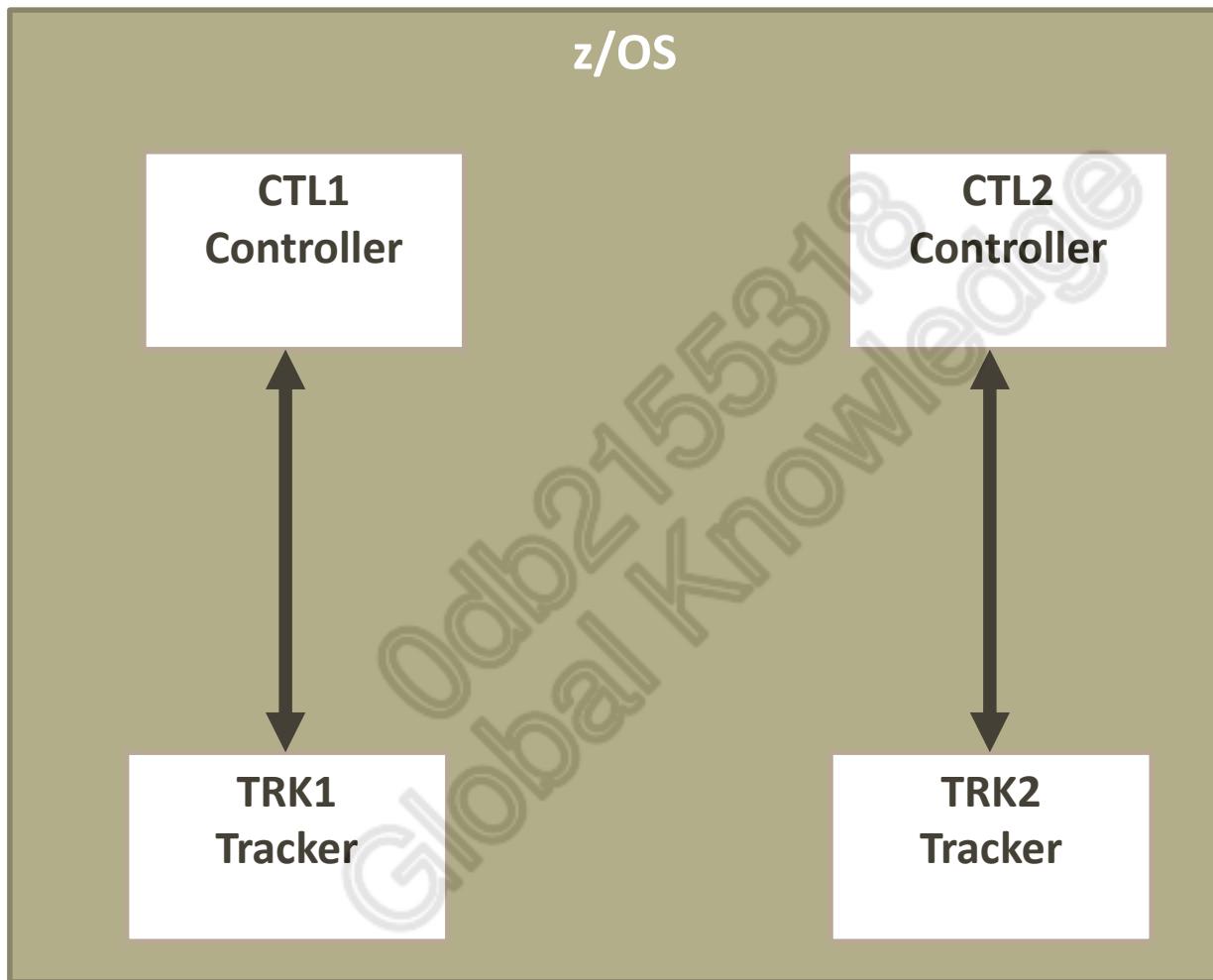
The communication process is as follows:

1. The Event Manager task of the controller writes an event in EQQOUCHEV every time a job completes or terminates and updates the next-to-write counter.
2. The Output collector started task reads an event from this data set, checkpoints it in EQQOUCKP, dispatches it to the proper thread, and marks the event as processed moving to the next-to-read index in the data set header

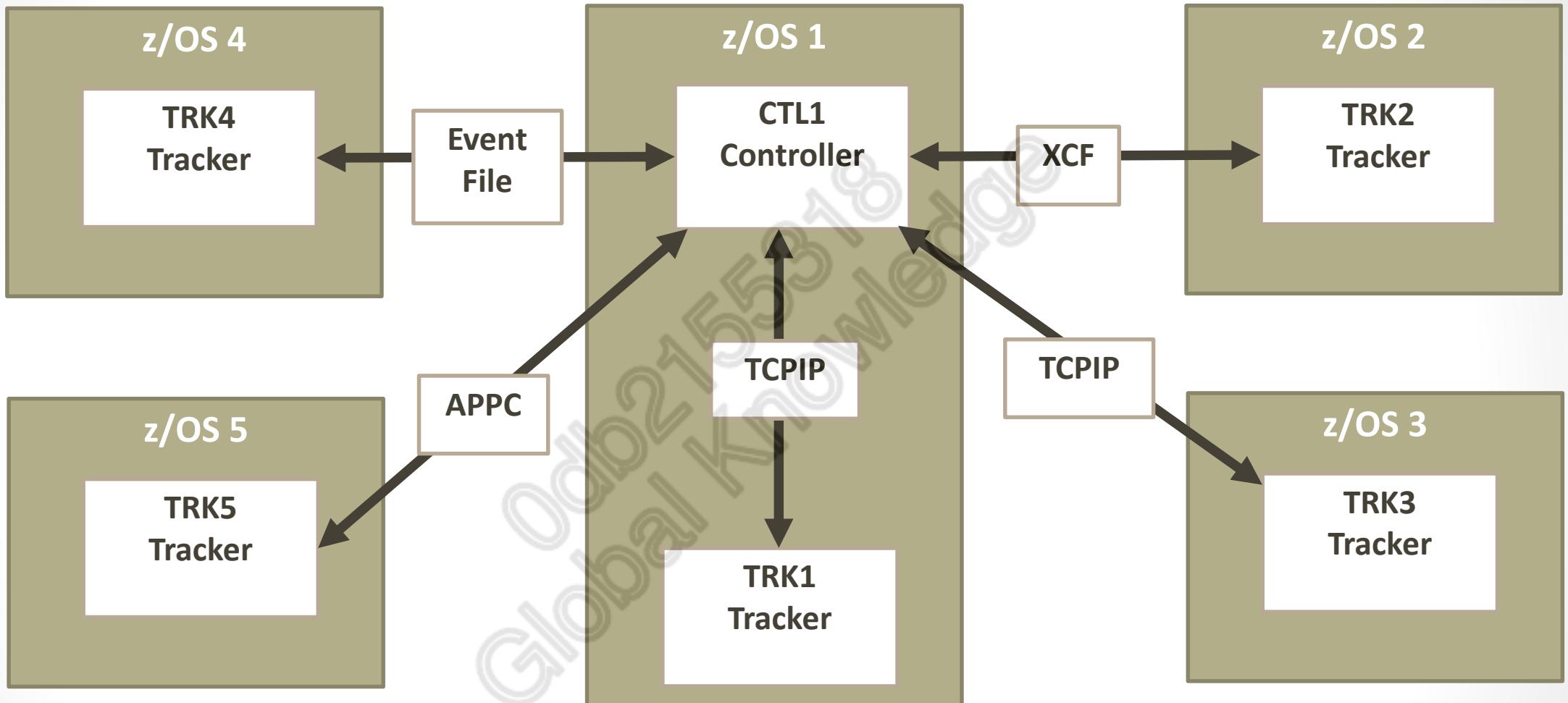
Configuration Sample

Odb2155318
Global Knowledge

Tracker and Controller on same z/OS

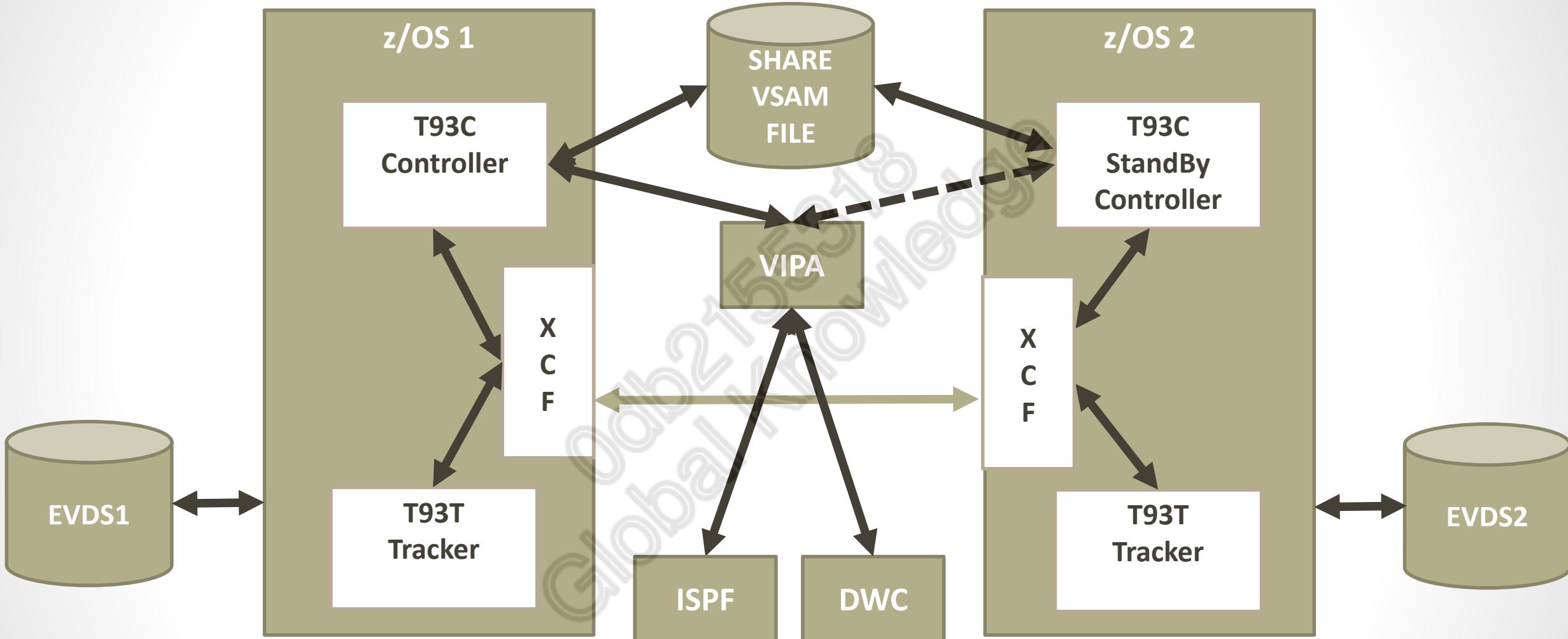


Tracker on other z/OS

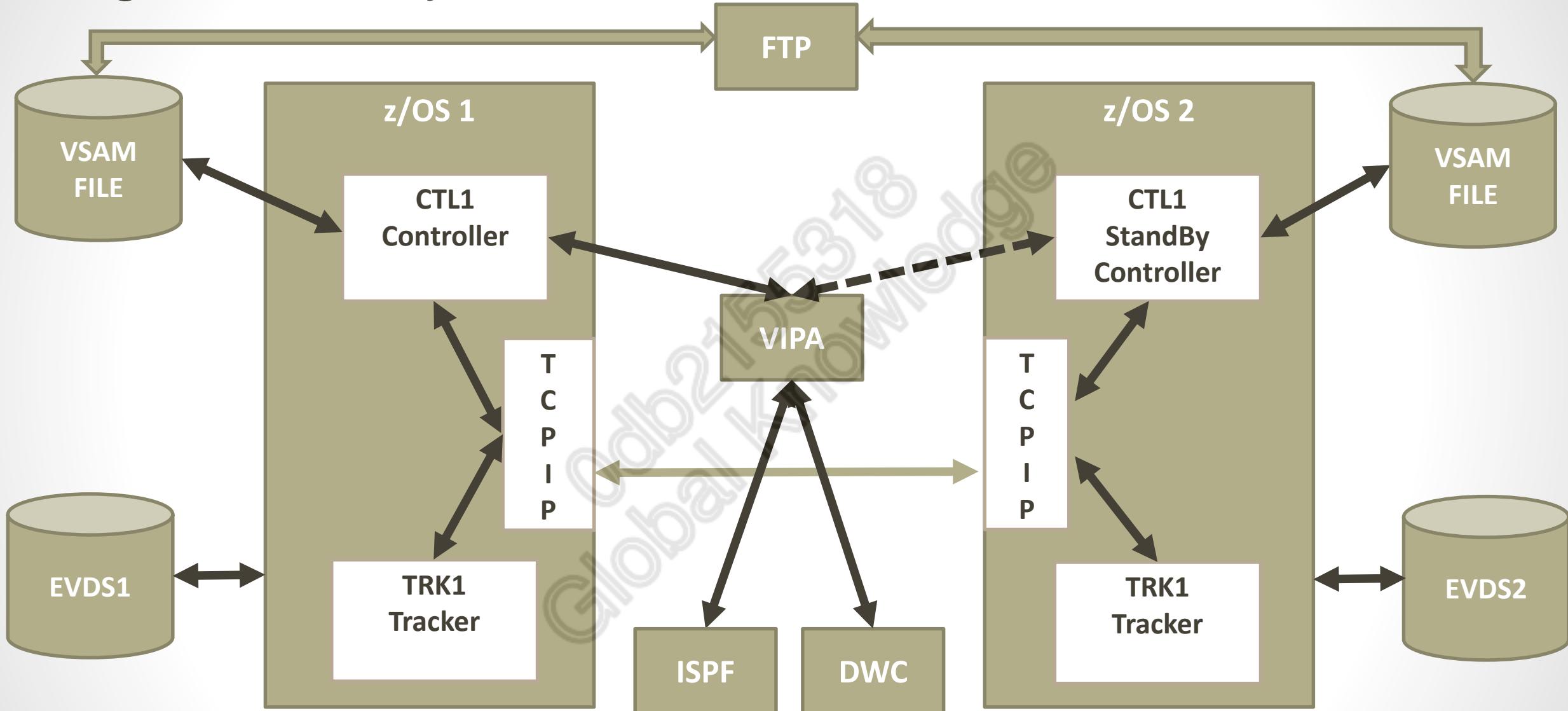


You define at least one workstation per Tracker

High Availability with SYSPLEX



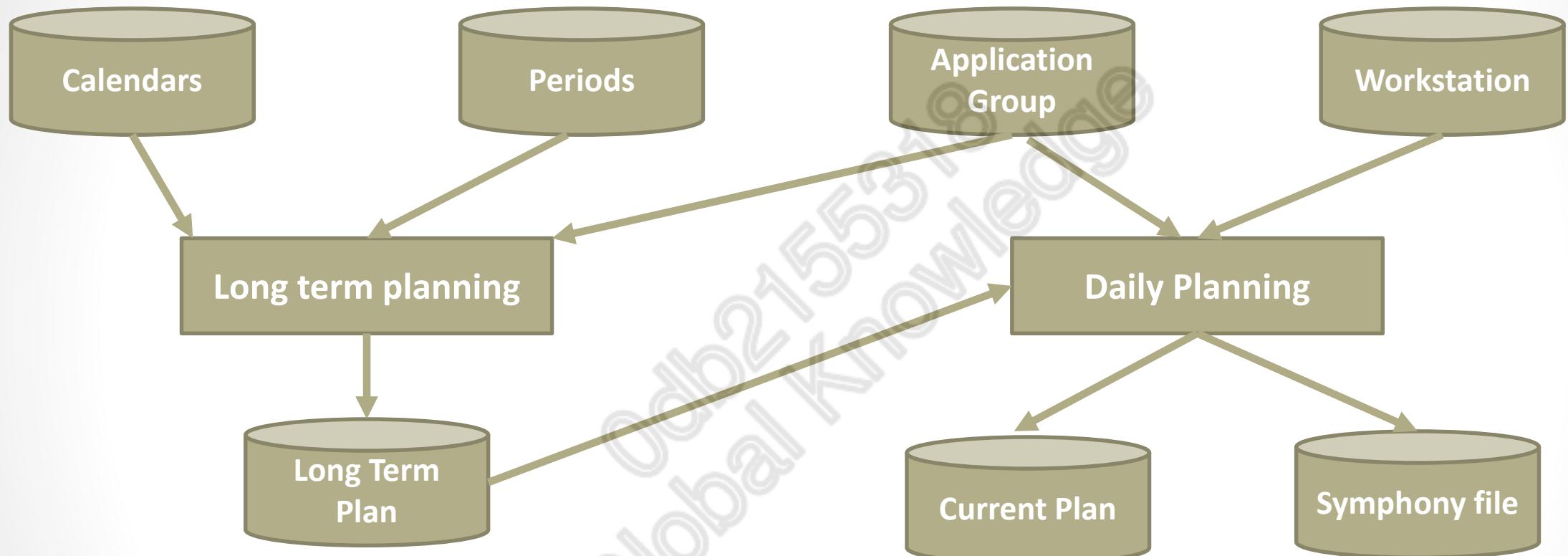
High Availability with TCPIP



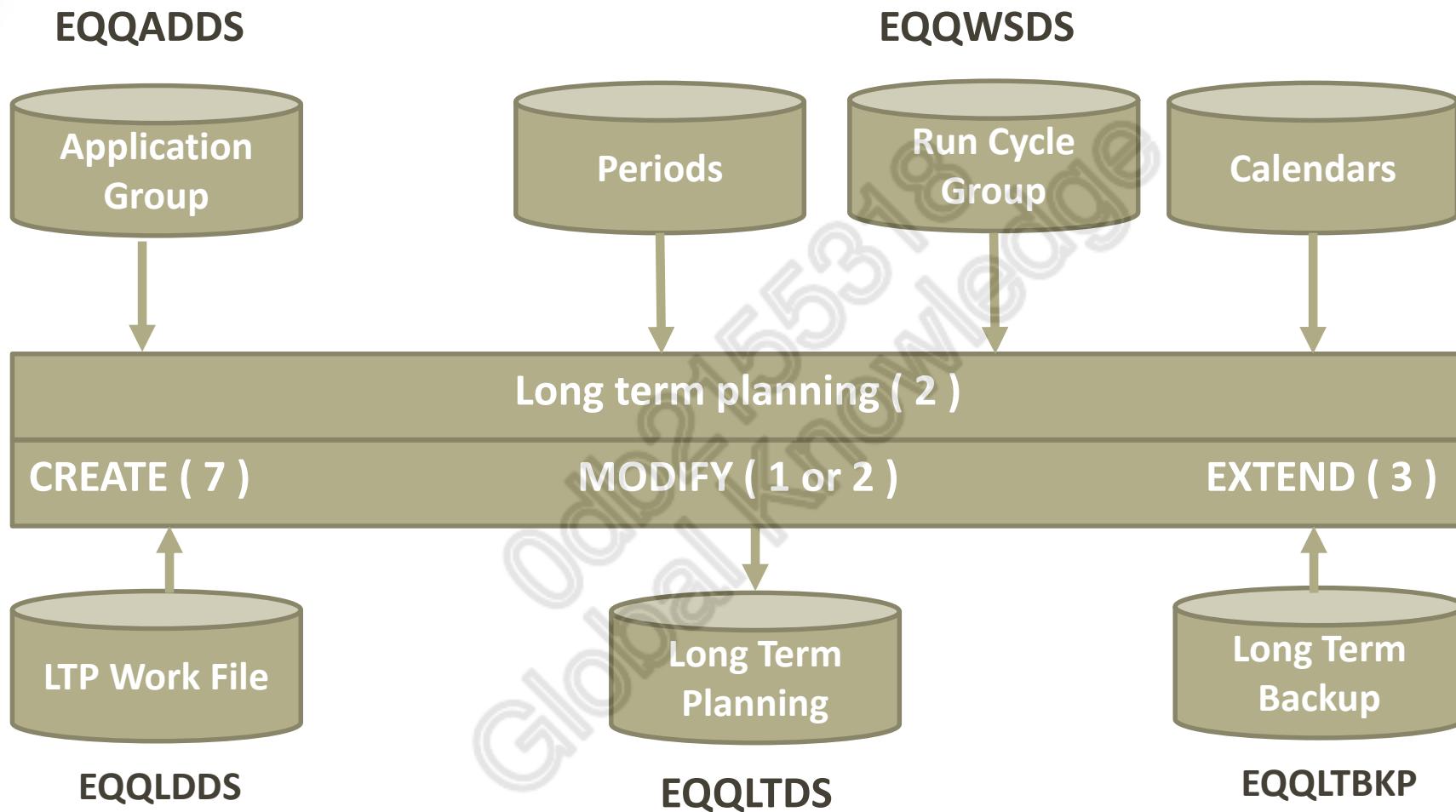
PLANNING

Odb2155318
Global Knowledge

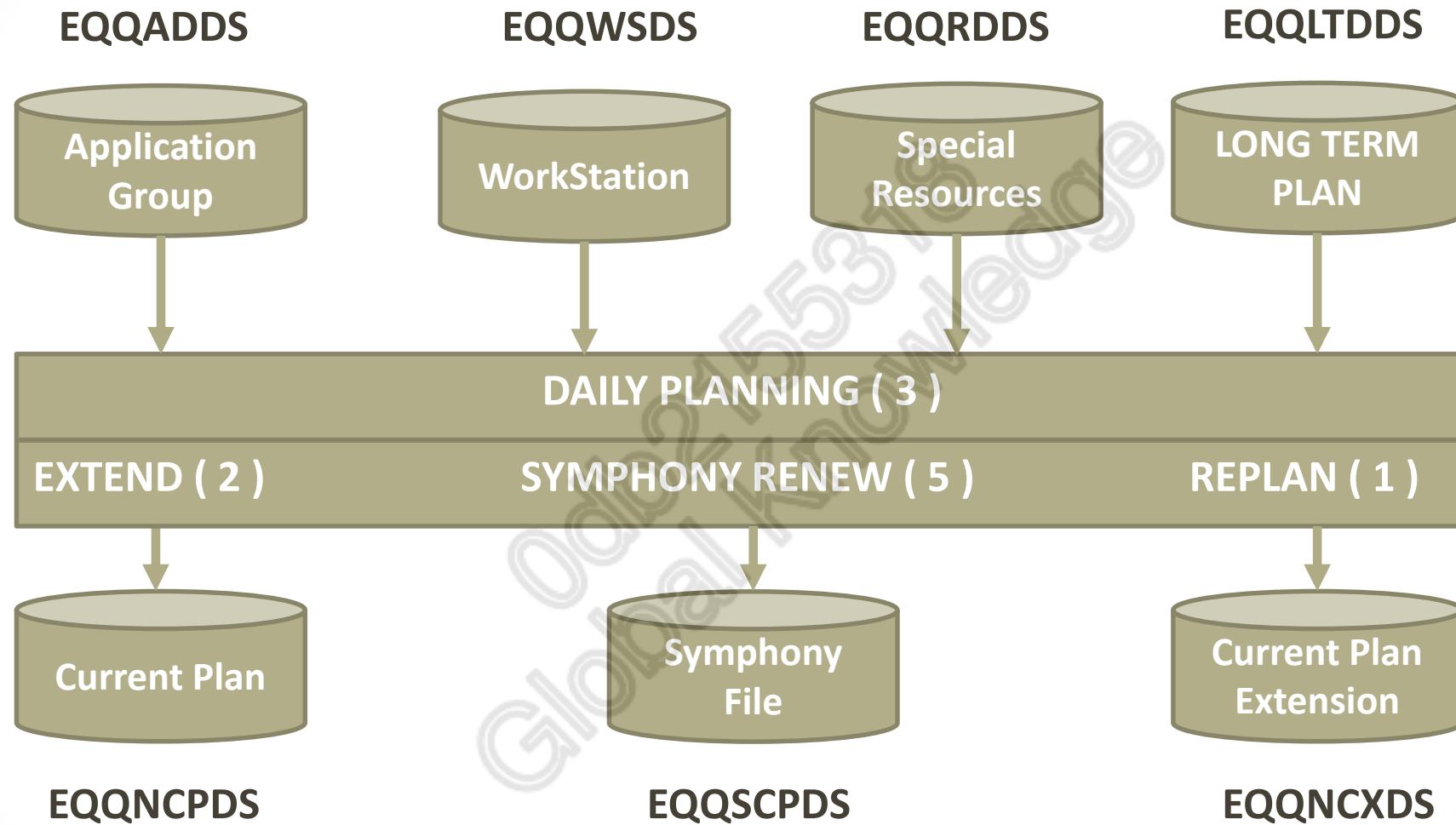
Policy Management



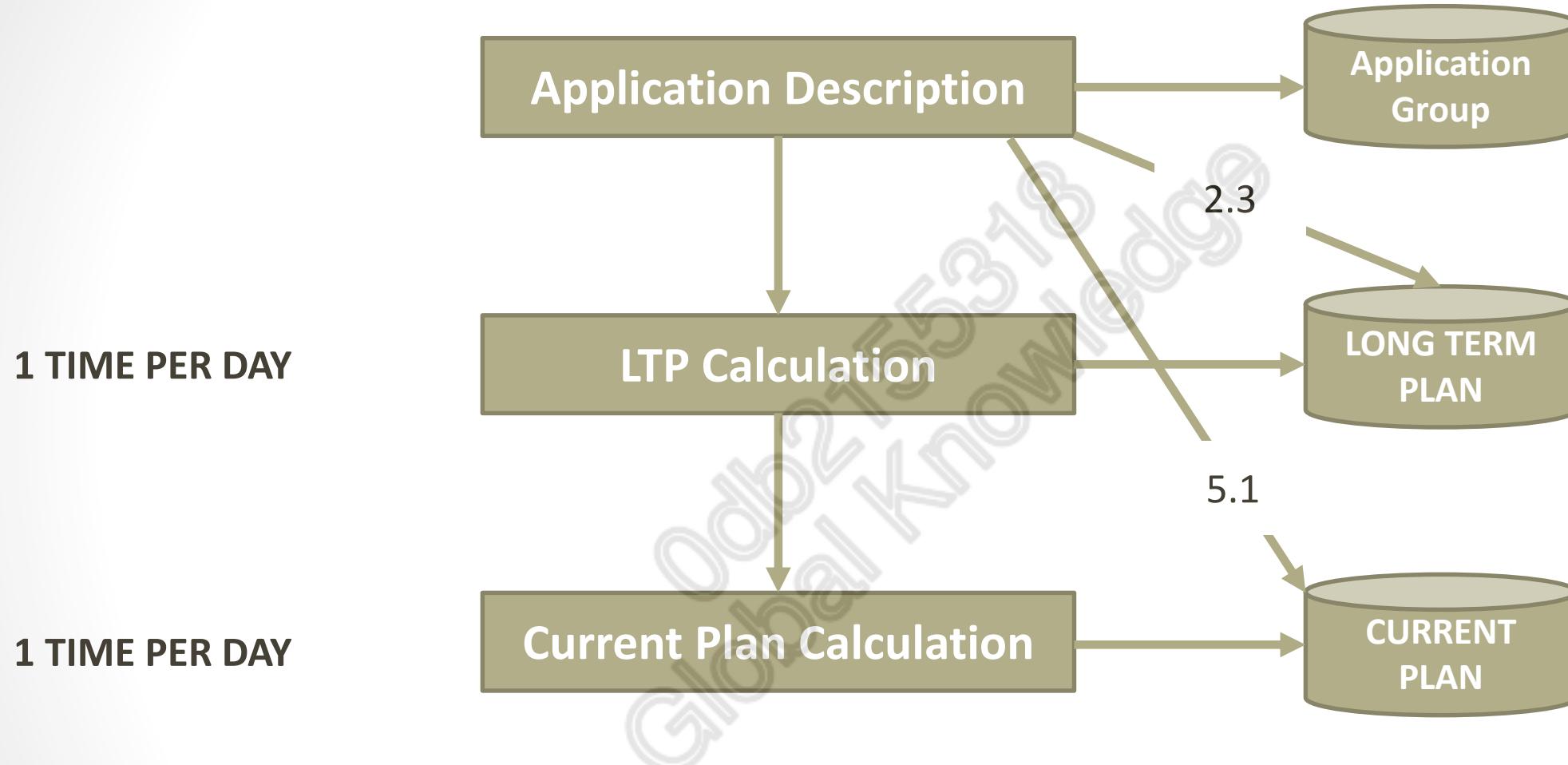
LONG TERM PLANNING



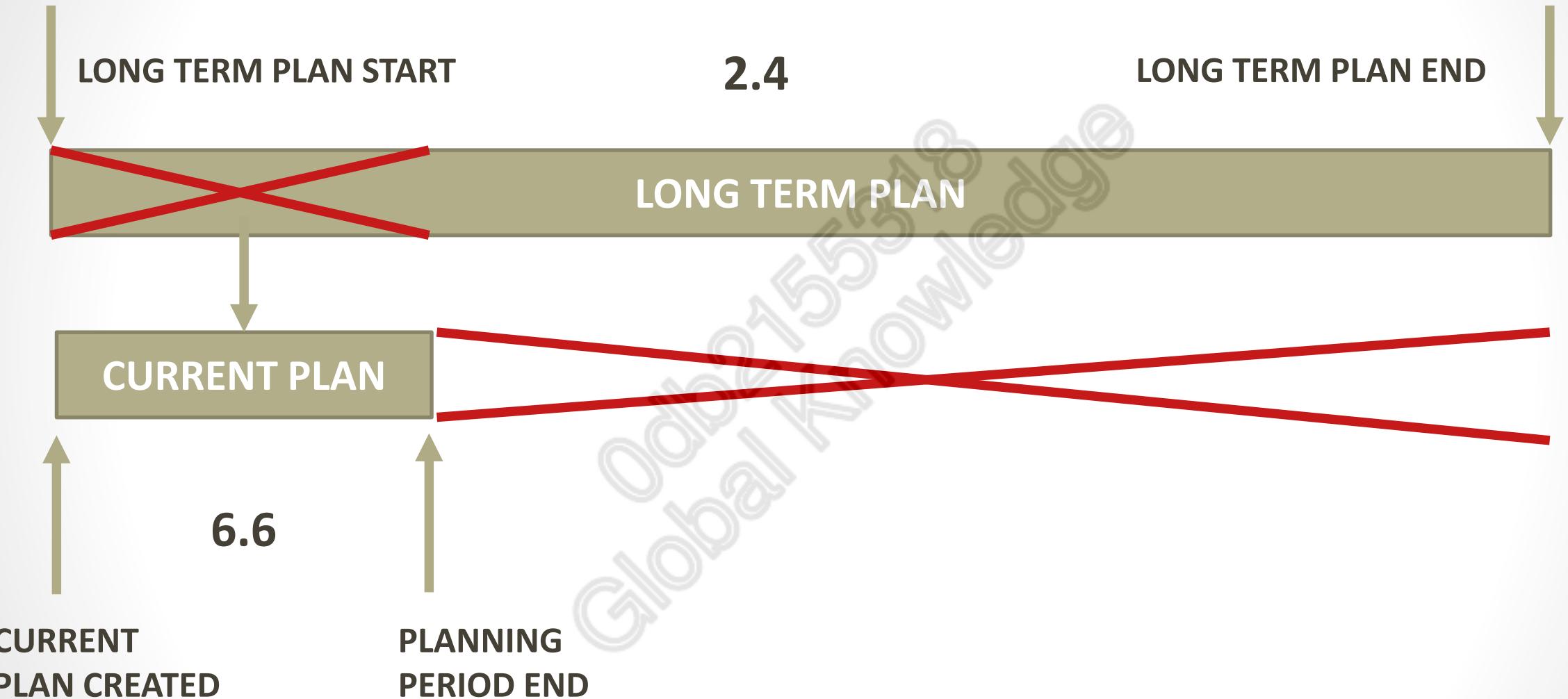
DAILY PLANNING



Application life cycle



Planning



Planning Activity

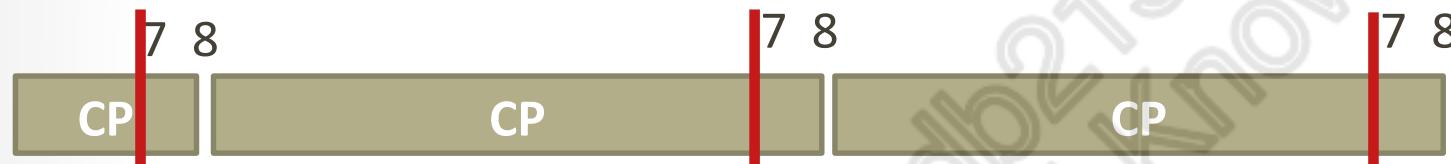
MONDAY

TUESDAY

WEDNESDAY

LTP

RUN of LTP and CP at 7.00 AM and End of CP at 8 AM



RUN of LTP and CP at 4.00 PM and End of CP at 5 PM



IWS Installation

Odb2155318
Global Knowledge

Objective

In this chapter , you will learn about the installation of IWS for z/OS product

After completing this training, you should be able to:

- Use EQQJOBS
- Decide option
- Select into the sample library

EQQJOBS setup

ISPF Allocations for EQQJOBS

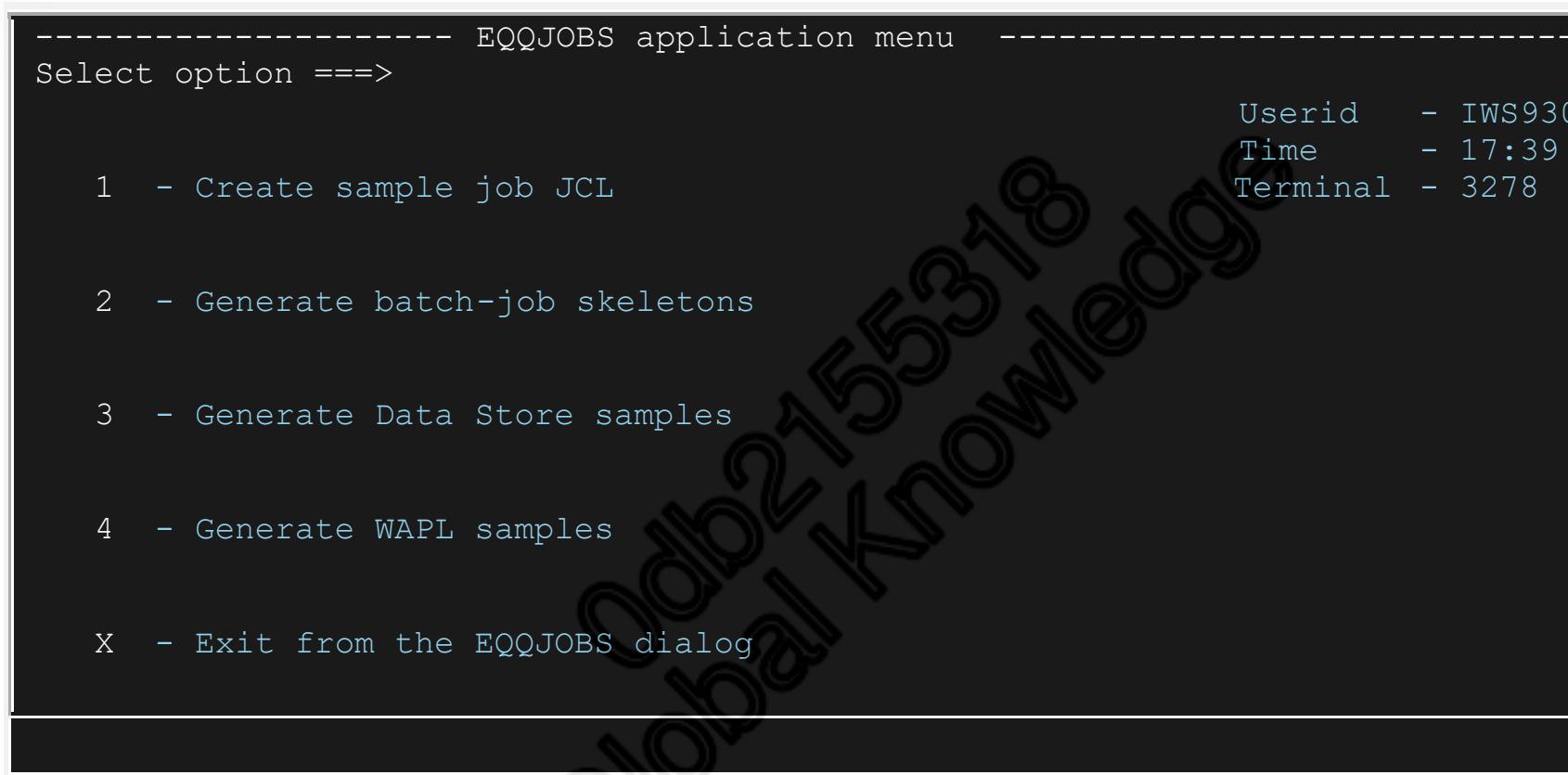
DDNAME

- SYSPROC
- ISPPLIB
- ISPSLIB

Libraries

- SEQQCLIB
- SEQQPNL0
- SEQQSAMP
- SEQQSKL0

EQQJOBS : Main screen



EQQJOBS : Option 1 Create sample JCL

```
----- Create sample job JCL -----
Command ===>

The dataset names specified on this panel should be fully qualified
names without any enclosing apostrophes.

Enter the name of the output library:

Sample job JCL      ==> IWS930.V9R3M0.INSTALL

Job statement information:

==> //IWS930XX JOB (1),TRAINING,CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID
==> //*
==> //*
==> //*

The following dataset names are used by one or more of the generated jobs.
Message library name ==> IWS930.V93RM0.SEQQMSG0
Data library name    ==> IWS930.V9R3M0.SEQQDATA
Parameter library    ==> IWS930.V9R3M0.SEQQPARM
Checkpoint dataset   ==> IWS930.V9R3M0.SEQQCKPT

Press ENTER to continue
```

EQQJOBS : Option 1 Screen 2

```
----- Create sample job JCL -----
Command ===>

Enter the following required job stream parameters:
Non-VSAM dsn prefix    ===> IWS930.V9R3M0
VSAM dsn prefix        ===> IWS930.V9R3M0
Unit name               ===> 3390          Default unit name
Primary volume serial  ===> IWS930          Primary volume serial for VSAM
Backup volume serial   ===> IWS930          Secondary volume serial for VSAM
SYSOUT class            ===> *              SYSOUT class for reports

The following information is optional:
STEPLIB dsname          ===> IWS930.V9R3M0.SEQQLDM0
VSAMCAT dsname           ===>
VSAM password            ===>
Dsn prefix of old
  VSAM files             ===>
  non-VSAM files         ===>

Samples with cloning support generated: Y (Y/N)
Static symbol used      ===> SYSCLONE      Without enclosing '&' and period

Press ENTER to continue
```

EQQJOBS : Option 1 Screen 3

```
----- Create sample job JCL -----
Command ===>

END TO END WITH FAULT TOLERANCE: Y          (Y= Yes ,N= No)
Installation Directory      ===> /usr/lpp/TWS/V9R3M0
                                ===>
                                ===>
Work Directory           ===> /var/TWS/inst
                                ===>
                                ===>
User for IWSz Address Space ===> 0
Refresh CP group          ===> 0
RESTART AND CLEANUP (DATA STORE) Y          (Y= Yes ,N= No)
Reserved destination       ===> DSTDEST
Connection type            ===> TCP          (SNA/XCF/TCP)
SNA Data Store luname     ===>             (only for SNA connection )
SNA FN task luname        ===>             (only for SNA connection )
Xcf Group                 ===>             (only for XCF connection )
Xcf Data store member      ===>             (only for XCF connection )
Xcf FL task member         ===>             (only for XCF connection )
TCP Data store host name: ===> ZOS220      (only for TCP connection )
TCP Data store port number ===> 5555        (only for TCP connection )
Press ENTER to continue
```

EQQJOBS : Option 1 Screen 4

```
----- Create sample job JCL -----
Command ===>

JAVA UTILITIES ENABLEMENT:          Y      (Y= Yes ,N= No)
Installation Directory      ===> /usr/lpp/TWS/V9R3M0
                                ===>
                                ===>
                                ===> /usr/lpp/java/J6.0_64
                                ===>
                                ===>
                                ===> /var/TWS/inst
                                ===>
                                ===>
User ID                      ===> 0
Group ID                     ===> 0

JZOS Batch Launcher
  PDSE Library      ===> JVA710.SIEALNKE
  Load Module Name  ===> JVMLDM71

Press ENTER to continue
```

EQQJOBS : Option 1 Screen 5

```
----- Create sample job JCL -----
Command ==>

SSL FOR TCP/IP CONNECTION      Y          (Y= Yes ,N= No)
SSL Work Directory      ===> /var/TWS/inst/ssl
                         ===>
                         ===>

SSL User ID      ===> 0
SSL Group ID      ===> 0
OUTPUT COLLECTOR      N          (Y= Yes ,N= No)
Class      ===> A
Writer      ===> ZCENT
MLOG SWITCH      Y          (Y= Yes ,N= No)
Switchlimit      ===> 5
                         (0 means function is inactive,
                          other values represent the
                          number of records that must
                          be reached in the current MLOG
                          file to switch to the alterna
                          MLOG)
STEP AWARENESS      Y          (Y= Yes ,N= No)
Store on primary controller  ===> Y
                                         (Y= Yes ,N= No) forced to N if
                                         Step Awareness is N

Press ENTER to continue
```

EQQJOBS : Option 1 Screen 6

```
----- Create sample job JCL -----
Command ===>

BKPT configuration: Y          (Y= Yes ,N= No)
Backup data set name ==> IWS930.V9R3M0.BKPT

BKPTOPTS statement procedure names:

DUMP Procedures for:
NCP,NCX,NXD Files    ==> DMPNCP
CX,CP1,XD1 Files     ==> DMPCX1
CX,CP2,XD2 Files     ==> DMPCX2
LTP File              ==> DMPLTP
RESTORE Procedures:
NCP,NCX,NXD Files    ==> RSTNCP
CX,CP1,XD1 Files     ==> RSTCX1
CX,CP2,XD2 Files     ==> RSTCX2
LTP File              ==> RSTLTP

Press ENTER to create sample job JCL
```

EQQJOBS : JCL

MEMBER	DESCRIPTION
EQQAUDIB	Sample to invoke EQQAUDIT in batch
EQQBENCO	SAMPLE JCL TO ENCRYPT THE PASSWORDS IN THE OSLCOPTS
EQQBENCR	SAMPLE JCL TO ENCRYPT THE PASSWORDS IN THE USRREC
EQQBKP	SAMPLE STARTED TASK PROCEDURE FOR A IWSz for Remote Host Standby
EQQBKPO	SAMPLE STARTED TASK PROCEDURE FOR A IWSz for Remote Host Standby
EQQBKPOP	Backup Controller configuration parameters
EQQBKPP	Backup Controller configuration parameters
EQQBSCAN	Uses the batch loader to scan an Application Description
EQQBSUBS	Uses the batch loader to create the Application Descriptions and Operator Instructions.
EQQBVSAM	Deletes and defines an Application Description data set and creates an Application Description and Operator Instructions, by using the batch loader.
EQQCHKEV	Utility that checks if all events in EQQTWSIN and EQQTWSOU were correctly processed.
EQQCON	Sample started task procedure for a controller and tracker in the same address space
EQQCONO	Sample started task procedure for controller only
EQQCONOP	Sample parameters used by EQQCONO
EQQCONP	Sample initial parameters for a controller and tracker in the same address space
EQQDBENC	Contains the JCL to encrypt the password in the DBOPT statement
EQQDBOPT	Sample DBOPT statement
EQQDP COP	JCL and usage notes for copy VSAM function
EQQE2EP	Sample initial parameters for server and batch to define if the end-to-end scheduling with fault tolerance capabilities is active
EQQFLWAT	FILEWATCH UTILITY USED TO CHECK HFS OR ZFS FILES
EQQICNVH	Sample job to migrate history DB2 tables
EQQICNVS	Sample job to migrate VSAM files
EQQJER2U	Sample to restore the EXIT7 as a JES2 usermod

EQQJOBS : JCL

MEMBER	DESCRIPTION
EQQJER2V	Sample to restore the EXIT5 as a JES2 usermod
EQQJER3U	Sample to restore the EQQUX191 AND EQQUX291 as a JES3 usermod
EQQJES2	ASSEMBLE AND LINKEDIT THE JES2 EXIT7
EQQJES2U	INSTALL THE EXIT7 EXIT AS A JES2 USERMOD
EQQJES2V	INSTALL THE EXIT51 EXIT AS A JES2 USERMOD
EQQJES21	ASSEMBLE AND LINKEDIT THE JES2 EXIT51
EQQJES3	ASSEMBLE AND LINKEDIT A JES3 EXIT
EQQJES3U	INSTALL THE EQQUX291 EXIT AS A JES3 USERMOD
EQQMFTWSO	MIGRATE THE TWSOU EVENTS DATASET FROM 120 BYTES TO 160 BYTE
EQQORST	A RESET ACTION IN USS ENVIRONMENT FOR END-to-END FEATURE
EQQOUC	STARTED TASK PROCEDURE FOR A IWSz OUTPUT COLLECTOR
EQQOUCH	Output Collector JOBLOG HEADER
EQQOUCP	parameters for the startup procedure EQQOUC
EQQPCS01	ALLOCATE IWSz PRODUCTION VSAM DATASETS
EQQPCS02	ALLOCATE EVENT DATA SET
EQQPCS03	ALLOCATE IWSz COPIES OF VSAM DATASETS
EQQPCS05	ALLOCATES TWS for z/OS USS FILES for END to END
EQQPCS06	ALLOCATE E2E FEATURE SEQUENTIAL
EQQPCS07	ALLOCATE IWSz RESTART AND CLEAN UP
EQQPCS08	JOB ALLOCATES TWS for z/OS USS FILES for HISTORICAL DATA ARCHIVING FOR DWC and EDWA FEATURE FOR DATA SET TRIGGERING
EQQPCS09	ALLOCATES TWS for z/OS DATA SETS (GDG ROOT AND VSAM BACKUP DATA SET)
EQQPCS10	CREATE TWS for z/OS USS WORK DIRECTORY USED FOR SSL
EQQPCS11	ALLOCATES ALL THE DATA SETS USED BY THE OUTPUT COLLECTOR FEATURE
EQQPCS12	ALLOCATES TWS for z/OS GDG DATA SET ROOT NEEDED BY THE MLOG SWITCH FEATURE
EQQPCS13	ALLOCATES TWS for z/OS GDG DATA SET ROOT NEEDED BY THE MLOG SWITCH FEATURE

EQQJOBS : JCL

MEMBER	DESCRIPTION
EQQPMCKP	Merges the checkpoint data sets of the old and new systems in the production system migration process
EQQRAD	Restore procedure for the Application Description
EQQRCERA	COPY INTO RACF DATABASE A NOT EXPIRING SECURITY CERTIFICATE FOR HTTPS COMMUNICATION TO BE USED AS CERTIFICATION AUTHORITY
EQQRCERD	COPY INTO RACF DATABASE A NOT EXPIRING SECURITY CERTIFICATE FOR HTTPS COMMUNICATION TO BE USED AS CERTIFICATION AUTHORITY
EQQRCERT	COPY INTO RACF DATABASE A NOT EXPIRING SECURITY CERTIFICATE FOR HTTPS COMMUNICATION TO BE USED AS CERTIFICATION AUTHORITY
EQQRECP1	Restore procedure for CX, CP1, XD1, ST files.
EQQRECP2	Restore procedure for CX, CP2, XD2, ST files.
EQQRENCP	Restore procedure for NCP, NCX, NXD files.
EQQRESDB	Restore procedure for AD, RD, WS, SI, OI files
EQQRESLT	Restore procedure for LTP files.
EQQRJS1	Restore procedure for the primary JCL repository.
EQQRJS2	Restore procedure for the secondary JCL repository.
EQQROI	Restore procedure for the Operator Instructions.
EQQRRD	Restore procedure for the Resource Description.
EQQRSI	Restore procedure for the Side Information file.
EQQRWS	Restore procedure for the Workstation Description.
EQQSAD	Send procedure for the Application Description.
EQQSAMPI	Copies sample databases from the sample library to VSAM data sets.
EQQSECP1	Dump procedure for CX, CP1, XD1, ST files.
EQQSECP2	Dump procedure for CX, CP2, XD2, ST files.
EQQSENCP	Dump procedure for NCP, NCX, NXD files.
EQQSENDB	MAKES DUMP, TERSE OF AD, RD, WS, SI, OI AND SENDS THEM TO THE BACKUP CONTROLLER VIA FTP
EQQSENLT	Dump procedure for LTP files.
EQQSER	Sample started task procedure for a Server.

EQQJOBS : JCL

MEMBER	DESCRIPTION
EQQSERP	Sample initial parameters for a Server.
EQQSJS1	Send procedure for the primary JCL repository.
EQQSJS2	Send procedure for the secondary JCL repository.
EQQSLCHK	JCL to perform a syntactic check on SCRIPT library members
EQQSMF	Updates SMF exits for IBM Workload Scheduler for z/OS
EQQSMLOG	Sample procedure that creates the GDG data set where the outgoing MLOG
EQQSOI	Send procedure for the Operator Instructions.
EQQSRD	Send procedure for the Resource Description.
EQQSSI	Send procedure for the Side Information file.
EQQSW5	Send procedure for the Workstation Description.
EQQTRA	Sample started task procedure for a tracker.
EQQTRAP	Sample initial parameters for a tracker.
EQQTROPT	Sample TRGOPT statement.
EQQ9SMDE	Updates the RACF class-descriptor table (ICHRCDE).
EQQ9SM01	Updates the RACF router table (ICHRFR01).

EQQJOBS : Option 2 Batch jobs skeleton

```
----- Generate IWSz batch-job skeletons -----
Command ===>

Enter the name of the output library. This should be a fully qualified
dataset name without any enclosing apostrophes.

Batch-job skeletons    ===> IWS930.V9R3M0.INSTALL

The following dataset names are used by one or more of the generated jobs.
You can specify an asterisk (*) to indicate the name of the subsystem.

Message library name  ===> IWS930.V93RM0.SEQQMSG0
Parameter library      ===> IWS930.V9R3M0.SEQQPARM
Member in parameter
  library             ===> IWSCBTCH
Checkpoint dataset     ===> IWS930.V9R3M0.SEQQCKPT

Press ENTER to continue
```

EQQJOBS : Option 2 screen 2

```
----- Generate IWSz batch-job skeletons -----
Command ===>

Enter the following required job stream parameters:

Non-VSAM dsn prefix ===> IWS930.V9R3M0
VSAM dsn prefix      ===> IWS930.V9R3M0
Unit name            ===> 3390          Default unit name
Unit name (temp ds) ===> SYSDA        Unit name for temporary datasets
Unit name (sort ds) ===> SYSDA        Unit name for sort work datasets
SYSOUT class         ===> *           SYSOUT class for reports

The following information is optional:

STEPLIB dsname       ===> IWS930.V9R3M0.SEQQLDM0
STEPCAT dsname       ===>
EQQMLOG dsname       ===>

The following information is REQUIRED WITH DBCS support:

KJSRTBL dsname      ===>

Press ENTER to continue
```

EQQJOBS : Option 2 screen 3

```
----- Generate IWSz batch-job skeletons -----
Command ===>
Specify if you want to use the following optional features:

END TO END WITH FAULT TOLERANCE:      Y      (Y= Yes ,N= No)

RESTART AND CLEAN UP (DATA STORE):      Y      (Y= Yes ,N= No)

FORMATTED REPORT OF TRACKLOG EVENTS:   N      (Y= Yes ,N= No)
  EQQTROUT dsname      ===> IWS930.V9R3M0.&SYSCLONE.TRACKLOG
  EQQAUDIT output dsn  ===> IWS930.V9R3M0.&SYSCLONE.EQQAUDIT.REPORT

JAVA UTILITIES ENABLEMENT:              Y      (Y= Yes ,N= No)
  Work Directory        ===> /var/TWS/inst
                                ===>
                                ===>
  JZOS PDSE Library    ===> JVA710.SIEALNKE
  JZOS Load Module Name ===> JVMLDM71
  REXX SYSEXEC dsname  ===> IWS930.V9R3M0.SEQQMISC
  Input XML dsname for
  data set triggering   ===> IWS930.V9R3M0.EVLIB.XML($$$$$$)

Press ENTER to generate IWSz batch-job skeletons
```

EQQJOBS : Batch Skeleton

MEMBER	DESCRIPTION	ISPF
EQQADCD5	APPLICATION CROSS REFERENCE OF CONDITIONAL DEPENDENCIES	1.4.4.8
EQQADCO5	CALCULATE AND PRINT APPLICATION RUNDATES	1.4.3 A
EQQADDE5	APPLICATION CROSS REFERENCE OF EXTERNAL DEPENDENCIES	1.4.4.7
EQQADPRA5	APPLICATION PRINT PROGRAM	1.4.1
EQQADVDS5	APPLICATION OPERATION VARIABLE DURATION AND DEADLINE	1.4.4.9
EQQADXRS5	APPLICATION CROSS REFERENCE PROGRAM	1.4.4.5
EQQADX1S5	APPLICATION CROSS REFERENCE OF SELECTED FIELDS	1.4.4.6
EQQAMUUPS5	APPLICATION DESCRIPTION MASS UPDATE	1.4.5
EQQAPARS5	APAR TAPE	9.9
EQQAUDIS5	EXTRACT AND FORMAT TWSz JOB TRACKING EVENTS	10
EQQAUDNS5	CLIST TO CALL EXTRACT AND FORMAT TWSz JOB TRACKING EVENTS	10
EQQDBARS5	HISTORICAL DATA ARCHIVER FOR REPORTING FEATURE ENABLEMENT	3.6
EQQDPPEX5	DAILY PLANNING - PLAN NEXT PERIOD	3.2
EQQDPPRS5	DAILY PLANNING - PRINT CURRENT PERIOD RESULTS	3.4
EQQDPRCS5	DAILY PLANNING - REPLAN CURRENT PERIOD	3.1
EQQDPJS5	DAILY PLAN - PLAN NEXT SORT OF REPORT RECORDS (DBCS)	
EQQDPSTS5	DAILY PLAN - SORT OF REPORT RECORDS	
EQQDPTRS5	THIS JOB FIRST COPY VSAM DATA SETS, THEN RUN A DP TRIAL	
EQQJVPRS5	PRINT JCL VARIABLE TABLES	1.9.3
EQQLEXT5	LONG TERM PLANNING - EXTEND THE LONG TERM PLAN	2.3
EQQLMOAS5	LONG TERM PLANNING - MODIFY ALL OCCURRENCES	2.2.1
EQQLMOOS5	LONG TERM PLANNING - MODIFY ONE OCCURRENCE	2.2.2
EQQLPRAS5	LONG TERM PLANNING - PRINT THE LONG TERM PLAN	2.2.5
EQQLPRT5	LONG TERM PLANNING - PRINT LONG TERM PLAN	2.2.6
EQQLTRES5	LONG TERM PLANNING - CREATE THE LONG TERM PLAN	2.2.7
EQQLTRY5	LONG TERM PLANNING - PLAN A TRIAL PERIOD	2.2.4
EQQOIBAS5	OPERATOR INSTRUCTIONS - BATCH PROGRAM	1.5.4
EQQOIBLS5	OPERATOR INSTRUCTIONS - BATCH INPUT FROM A SEQ. DATA SET	1.5.5.2
EQQRGPRS5	RUN CYCLE GROUP PRINT PROGRAM	1.10
EQQSYRES5	DAILY PLANNING - RENEW SYMPHONY FROM CP	3.5
EQQTTPRS5	PRINT ALL PERIODS	1.3.9
EQQTPTRS5	PRINT ALL CALENDARS	1.2.9
EQQTRBLS5	EVENT DYNAMIC WORKLOAD AUTOMATION FEATURE ENABLEMENT	1.7.3
EQQWMIGS5	WORKSTATION MIGRATION	1.1.5
EQQWPRTS5	PRINT WORK STATION DESCRIPTION	1.1.9

EQQJOBS : Option 3 Datastore

```
----- Create Data Store samples -----
Command ===>

The dataset names specified on this panel should be fully qualified
names without any enclosing apostrophes.

Enter the name of the output library:

Sample job JCL      ===> IWS930.V9R3M0.INSTALL

Job statement information:

====> //IWS930XX JOB (1),TRAINING,CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID
====> //*
====> //*
====> //*

The following dataset names are used by one or more of the generated jobs.
Message library name ===> IWS930.V93RM0.SEQQMSG0
Parameter library      ===> IWS930.V9R3M0.SEQQPARM

Press ENTER to continue
```

EQQJOBS : Option 3 screen 2

```
----- Create Data Store samples -----
Command ===>

Enter the following required job stream parameters:
Non-VSAM dsn prefix    ===> IWS930.V9R3M0
VSAM dsn prefix        ===> IWS930.V9R3M0
Unit name               ===> 3390          Default unit name
Primary volume serial   ===> IWS930          Primary volume serial for VSAM

The following information is optional:
STEPLIB dsname          ===> IWS930.V9R3M0.SEQQLDM0
VSAMCAT dsname           ===>
VSAM password            ===>

Press ENTER to continue
```

EQQJOBS : Option 3 screen 3

```
----- Create Data Store samples -----
Command ==>

Enter the parameters to build DSTOPTS and DSTUTIL options samples:

Reserved destination      ==> DSTDEST
Connection type           ==> TCP (SNA/XCF/TCP)
    SNA Data Store luname ==>          (only for SNA connection)
    SNA Controller luname ==>          (only for SNA connection)
    Xcf Group             ==>          (only for XCF connection)
    Xcf Data store member ==>          (only for XCF connection)
    Xcf FL task member   ==>          (only for XCF connection)
    TCP Controller host name:
    ==> ZOS220            (only for TCP connection)
    TCP Controller port number ==> 444 (only for TCP connection)
Jobdata ret. period       ==> 5 (number of days)
JobLog retrieval          ==> Y (Y/N)
    Max n. lines to store ==> 100
    JobLog ret. period    ==> 5 (number of days)

Press ENTER to create sample job JCL
```

EQQJOBS : Datastore

MEMBER	DESCRIPTION
EQQCLEAN	Sample procedure invoking EQQCLEAN program
EQQDSCL	Batch Clean Up sample
EQQDSCLP	Batch Clean up sample parameters
EQQDSEX	Batch Export sample
EQQDSEXP	Batch Export sample parameters
EQQDSIM	Batch Import sample
EQQDSIMP	Batch Import sample parameters
EQQDSRG	Batch sample reorg
EQQDSRI	Batch Recovery index
EQQDSRIP	Batch Recovery index parameters
EQQDST	Sample procedure to start Data Store
EQQDSTP	Parameters for sample procedure to start Data Store
EQQPCS04	Allocate VSAM data sets for Data Store

EQQJOBS : Option 4 WAPL

```
----- Create WAPL samples -----
Command ===>

The data set names specified on this panel must be fully qualified
without any enclosing apostrophes.

Enter the name of the output library:

Sample job JCL      ===> IWS930.V9R3M0.INSTALL

The following data set names are used by one or more of the generated jobs.
WAPL data maps      ===> IWS930.V9R3M0.SEQQWAPL
MISC library        ===> TWS930.V9R3M0.SEQQMISC
Message library name ===> IWS930.V93RM0.SEQQMSG0
Steplib              ===> IWS930.V9R3M0.SEQQLDM0
REXX Libraries       ===> FAN140.SEAGALT
ISPF Messages        ===> ISP.SISP MENU
ISPF Panels          ===> ISP.SISPPENU
ISPF Skeletons       ===> ISP.SISPSENU
ISPF Tables           ===> ISP.SISPTENU

Subsys   ===> T93C    Language ===> EN    Version ===> 930

Press ENTER to continue
```

EQQJOBS : WAPL

MEMBER	DESCRIPTION
EQQILSON	ISPF LOADER STREAMED OUTPUT NOTATION UTILITY PROCEDURE
EQQWCMD1	RUN WAPL FROM IWS
EQQWCMD2	SCHEDULER OPERATIONAL ENVIRONMENT BATCH PROCEDURE
EQQWTSO1	This example completely sets up the WAPL environment for TSO
EQQWTSO2	TSO LOGON Proc or startup REXX/CLIST
EQQWTSO3	This example completely sets up the WAPL environment for TSO
EQQWTSO4	REXX which sets WAPL commands and the processes the results
EQQWTSX3	This example calls another REXX which sets up the WAPL environment
EQQWTSX4	Queue some commands to WAPL
EQQYXJCL	SCHEDULER OPERATIONAL ENVIRONMENT BATCH PROCEDURE

IWS SYSTEM

Odb2155318
Global Knowledge

Objective

In this chapter , you will learn about the z/OS system interface with IWS

After completing this training, you should be able to:

Define all the z/OS parameter for IWS

z/OS PARMLIB

MEMBER	ROLE
IEFSSNxx	Define Subsystem for Tracker and Controller
PROGxx	Define APF authorization and Linklist
SMPRMxx	Update SMF Parameters
IKJTSOxx	TSO/E Authorizations
SCHEDxx	Update PPT for performance
JESPRMxx	Add EXIT JES2

IEFSSN per version

version	component	
830	Controller	SUBSYS SUBNAME(C830) INITRTN(EQQINITG) INITPARM ('0,G')
830	Tracker	SUBSYS SUBNAME(T830) INITRTN(EQQINITG) INITPARM ('10,G')
850	Controller	SUBSYS SUBNAME(C850) INITRTN(EQQINITH) INITPARM ('0,H')
850	Tracker	SUBSYS SUBNAME(T850) INITRTN(EQQINITH) INITPARM ('10,H')
851	Controller	SUBSYS SUBNAME(C851) INITRTN(EQQINITI) INITPARM ('0,I')
851	Tracker	SUBSYS SUBNAME(T851) INITRTN(EQQINITI) INITPARM ('10,I')
860	Controller	SUBSYS SUBNAME(C860) INITRTN(EQQINITJ) INITPARM ('0,J')
860	Tracker	SUBSYS SUBNAME(T860) INITRTN(EQQINITJ) INITPARM ('10,J')
910	Controller	SUBSYS SUBNAME(C910) INITRTN(EQQINITK) INITPARM ('0,K')
910	Tracker	SUBSYS SUBNAME(T910) INITRTN(EQQINITK) INITPARM ('10,K')
920	Controller	SUBSYS SUBNAME(C920) INITRTN(EQQINITL) INITPARM ('0,L')
920	Tracker	SUBSYS SUBNAME(T920) INITRTN(EQQINITL) INITPARM ('10,L')
930	Controller	SUBSYS SUBNAME(C930) INITRTN(EQQINITM) INITPARM ('0,M')
930	Tracker	SUBSYS SUBNAME(T930) INITRTN(EQQINITM) INITPARM ('10,M')
860	Distributed Agent for z/OS	SUBSYS SUBNAME(D860) INITRTN(EELINITJ) INITPARM ('10,J')
930	Distributed Agent for z/OS	SUBSYS SUBNAME(D930) INITRTN(EELINITM) INITPARM ('10,M')

SETSSI command per version

version	component	
830	Controller	SETSSI ADD,SUBNAME=C830,INITRTN=EQQINITG,INITPARM='0,G'
830	Tracker	SETSSI ADD,SUBNAME=T830,INITRTN=EQQINITG,INITPARM='10,G'
850	Controller	SETSSI ADD,SUBNAME=C850,INITRTN=EQQINITH,INITPARM='0,H'
850	Tracker	SETSSI ADD,SUBNAME=T850,INITRTN=EQQINITH,INITPARM='10,H'
851	Controller	SETSSI ADD,SUBNAME=C851,INITRTN=EQQINITI,INITPARM='0,I'
851	Tracker	SETSSI ADD,SUBNAME=T851,INITRTN=EQQINITI,INITPARM='10,I'
860	Controller	SETSSI ADD,SUBNAME=C860,INITRTN=EQQINITJ,INITPARM='0,J'
860	Tracker	SETSSI ADD,SUBNAME=T860,INITRTN=EQQINITJ,INITPARM='10,J'
910	Controller	SETSSI ADD,SUBNAME=C910,INITRTN=EQQINITK,INITPARM='0,K'
910	Tracker	SETSSI ADD,SUBNAME=T910,INITRTN=EQQINITK,INITPARM='10,K'
920	Controller	SETSSI ADD,SUBNAME=C920,INITRTN=EQQINITL,INITPARM='0,L'
920	Tracker	SETSSI ADD,SUBNAME=T920,INITRTN=EQQINITL,INITPARM='10,L'
930	Controller	SETSSI ADD,SUBNAME=C930,INITRTN=EQQINITM,INITPARM='0,M'
930	Tracker	SETSSI ADD,SUBNAME=T930U,INITRTN=EQQINITM,INITPARM='10,M'
860	distributed Agent for z/OS	SETSSI ADD,SUBNAME=D860,INITRTN=EELINITJ,INITPARM='10,J'
930	Distributed Agent for z/OS	SETSSI ADD,SUBNAME=D930,INITRTN=EELINITM,INITPARM='10,M'

IEFSSN : Size of INITPARM

Calculation method for MAXECSA

Space requested = MAXECSA * 1024

Blocks = space requested / 1424 (round down to a whole number)

Space allocated = blocks * 1424

Events accommodated = blocks * 16

Number of jobs = Events accommodated / 6

MAXECSA

MAXECSA value	Amount of MAXECSA space requested	Blocks of ECSA space allocated (bytes)	Number of events accommodated	approximately Number of jobs
0	0	0 (0)	0	0
4	4 096	2 (2848)	32	5
8	8 192	5 (7120)	80	13
16	16 384	11 (15664)	176	29
36	36 864	25 (35600)	400	66
72	73 728	51 (72624)	816	136
100	102 400	71 (101104)	1136	189
200	204 800	143 (203632)	2288	381
400	409 600	287 (408688)	4592	765
500	512 000	359 (511216)	5744	957

EVENT Type

Event Type	Event Description	OP status	EXIT
1	Reader event. A job has entered the JES system	S Q	JES2 (EXIT51 EXIT7) JES3 (IATUX29)
2	Job-start event. A job has started to execute	S S	SMF (IEFUJI)
3S	Step-end event. A job step has finished executing	S S	SMF (IEFACTRT)
3J	Job-end event. A job has finished executing	C	SMF (IEFACTRT)
3P	Job-termination event. A job has been added to the JES output queues	C/E	JES2 (EXIT7) JES3 (IATUX19)
4	Print event. An output group has been printed	C	SMF (IEFU83)
5	Purge event. All output for a job has been purged from the JES system	C/E	SMF (IEFU83) JES2 (EXIT7)
S	Data Set Triggering Support		SMF (IEFU83)

EVENT FILE

```
Menu Utilities Compilers Help
BROWSE      TWS930.EV          Line 0000000780 Col 001
Command ===>          Scroll ===>
.....A3J..{. WS930A JOB04532...?..r...?.. .?....?..q....
.....A3P.\.. WS930A JOB04532...?.. .?....?.. .?....?..K...
.....A1 ..-. WS930A JOB04533...?-....?..^驥.....S0W1
.....A2 .... WS930A JOB04533...?./....?..^驥..?./....?
.....A3S...-. WS930A JOB04533...?..\?..^驥..?-5....?
.....A3S..{.. WS930A JOB04533...?....?..^驥..?..W....?..
.....A3J..{.. WS930A JOB04533...?....?..^驥..?-5....?..T...
.....A3P.\.. WS930A JOB04533...?....?..^驥..?-B...?..bK...
.....A1 ..-. WS930A JOB04534...?.>....?..>.....S0W
.....A2 .... WS930A JOB04534...?....?..>....?..?"....?
.....A3S...-. WS930A JOB04534...?..з....?..>....?..?..н
.....A3S...-. WS930A JOB04534...?..Л....?..>....?..ж....?..Л
.....A3J..{.. WS930A JOB04534...?..н....?..>....?..?..н...
.....A3P..{.. WS930A JOB04534...?..л....?..>....?..?..лK...
.....A1 ..-. WS930A JOB04535...?..n...?..mW.....S0W
.....A2 .... WS930A JOB04535...?..n?..mW...?..n....?
.....A3J..{.. WS930A JOB04535...?....?..mW...?..n'...?.. ...
.....A3P..{.. WS930A JOB04535...?..E...?..mW...?..n...?..K...
```

PROGxx : Authorization and Linklist

APF FORMAT(DYNAMIC)

APF ADD

DSNAME(IWS930.V9R3M0.SEQQLMD0) VOLUME(IWS930)

APF ADD

DSNAME(TWS930.V9R3M0.SEQQLMD0) SMS

LNKLST DEFINE NAME(LNKLST00)

LNKLST ADD NAME(LNKLST00) DSN(IWS930.V9R3M0.SEQQLMD0) VOLUME(IWS930)

LNKLST ADD NAME(LNKLST00) DSN(TWS930.V9R3M0.SEQQLMD0) SMS

LNKLST ACTIVATE NAME(LNKLST00)

SMF

IBM Workload Scheduler for z/OS uses the following SMF exits

- IEFUJI Job initiation exit
- IEFACTRT Job-end and step-end exits
- IEFU83 Record write exit. It is optional, and required only for data set triggering, automatic time change, and print event functions.

IBM Workload Scheduler for z/OS uses the following SMF record types:

- 6 (external Writer) → For PRINT (A4 and B4) events, used only for tracking work on PRINT workstations
- 14 (INPUT or RDBACK Data Set Activity) → Only for data set triggering with SRREAD=YES
- 15 (Data Set Activity) → For data set triggering with SRREAD=YES or SRREAD=NO
- 18 (Rename Non-VSAM Data Set Status) → Only if you want to monitor renaming data sets.
- 26 (JES2/3 Job Purge) → For all job tracking
- 30 (Common address space work) → For all job tracking
- 64 (VSAM Component or Cluster Status) → Only for data set triggering with VSAM data sets
- 90 (system status) → Only if you want automatic daylight savings time change

IKJTSO : TSO/E PARAMETER

AUTHSF : identifies programs that are authorized when invoked through the TSO/E Service Facility

AUTHCMD : identifies authorized commands to TSO/E

AUTHCMD NAMES(+

BACKUP +

BULKDISC +

JSUACT +

OPINFO +

OPSTAT +

SRSTAT +

EQQYLTOP)

AUTHSF NAMES(+

EQQMINOF /*8.2.0 */ +

EQQMINOR /*8.3.0 */ +

EQQMINOH /*8.5.0 */ +

EQQMINOI /*8.5.1 */ +

EQQMINOJ /*8.6.0 */ +

EQQMINOK /*9.1.0 */ +

EQQMINOL /*9.2.0 */ +

EQQMINOM /*9.3.0 */ +

)

SCHEDEXX : Performance parameter

PPT PGMNAME(EQQMAJOR) NOSWAP

Then Controller and Tracker are Non SWAP

JESPRM : EXIT PARAMETER

JES2 PARMLIB

LOAD(OPCAXIT7)	EXIT(7) ROUTINES=OPCAENT7,STATUS=ENABLED
LOAD(TWSXIT51)	EXIT(51) ROUTINES=TWSENT51,STATUS=ENABLED

DYNAMIC ADD EXIT

```
$ADD LOADMOD(OPCAXIT7),STORAGE=PVT  
$T EXIT(7),ROUTINES=OPCAENT7,STATUS=ENABLED
```

```
$ADD LOADMOD(TWSXIT51),STORAGE=PVT  
$T EXIT(51),ROUTINES=TWSENT51,STATUS=ENABLED
```

Event 1,3P,5

IWS Configuration

Odb2155318
Global Knowledge

Objective

In this chapter , you will learn about the configuration of IWS product

After completing this training, you should be able to:

- Design architecture
- Understand Parameter
- Understand Link parameter

Controller and Tracker PARMLIB

```
//T93C      EXEC PGM=EQQMAJOR,REGION=64M,PARM='T93C',TIME=1440  
//STEPLIB    DD  DISP=SHR,DSN=TWS930.SEQQLMD0  
//EQQMLIB    DD  DISP=SHR,DSN=TWS930.SEQQMSG0  
//EQQPARM   DD  DISP=SHR,DSN=TWS930.SEQQPARM
```

```
//T93T      EXEC PGM=EQQMAJOR,REGION=64M,PARM='T93T',TIME=1440  
//STEPLIB    DD  DISP=SHR,DSN=TWS930.SEQQLMD0  
//EQQMLIB    DD  DISP=SHR,DSN=TWS930.SEQQMSG0  
//EQQPARM   DD  DISP=SHR,DSN=TWS930.SEQQPARM
```

PARAMETER 1/4

NAME	TRACKER	CONTROLLER	SERVER	DATASTORE	BATCH	PIF	MEMBER	OPTIONS
ALERTS	Y	Y					P	Generating NetView®, IBM® Tivoli® Monitoring, message log, and WTO alerts.
AROPTS		Y					M	Automatic job recovery.
AUDIT		Y					P	Creating audit information for changes
AUDITCP		Y					P	Creating audit information for automatic status changes of an operation condition in the current plan.
AUTHDEF	Y	Y					P	Security checking
BATCHOPT					Y		M	Specifying options for all batch jobs
BKPTOPTS							P	Local attributes for TCP/IP Between 2 controllers
CPUREC					Y		M	Specifying the configuration options for a fault-tolerant workstation
DBCSOPTS		Y					M	Japanese language feature
DBOPT			Y		Y		M	Dynamic Workload Console reporting
DOMREC					Y		M	Defining a domain for distributed agents network
DSTOPTS				Y			P	Specifying options for the data store

PARAMETER 2/4

NAME	TRACKER	CONTROLLER	SERVER	DATASTORE	BATCH	PIF	MEMBER	OPTIONS
DSTUTIL				Y			M	Specifying options for data store batch utilities and the CleanUp subtask
ERDROPTS	Y	Y					M	Event reader task.
EWTROPTS	Y						P	Event writer task.
EXITS	Y	Y					P	Calling IWS for z/OS exits
FLOPTS		Y					P	Communicating with data store (allowing job log retrieval and restart and cleanup functions).
HTTPOPTS		Y					P	Tracking jobs running on z-centric agents and for retrieving their job execution logs
INCLUDE		Y					P	NOERROR table definition members
INIT			Y			Y	M	Run-time options for processing requests from a PIF application and a server
INTFOPTS		Y					P	Requests from programming interfaces
JCCOPTS	Y						M	Job completion checker task
JTOPTS		Y					P	Determining how operations behave at workstations and how they are submitted and tracked

PARAMETER 3/4

Name	Tracker	Controller	Server	Datastore	Batch	PIF	Member	Options
MONOPTS		Y					P	Enabling monitoring by an external agent. Used by IBM Tivoli Monitoring
MONPOL		Y					P	Defining the monitoring policy to be used by external monitors
NOERROR		Y					P	Treating job-tracking error codes as normal completion codes
OPCOPTS	Y	Y					P	Starting IBM Workload Scheduler for z/OS subtasks
RCLDDP		Y					P	Listing protected DDnames
RCLDSNP							M	Listing protected data set name
RCLSKIP		Y					P	Listing the INCLUDEs to keep at the beginning of a JCL when it is tailored by the Restart and Cleanup function.
RESOPTS		Y					P	Controlling special resources
RESOURCE					Y		P	Defining for which special resources daily planning should produce reports
RODMOPTS		Y					P	Monitoring special resources through RODM

PARAMETER 4/4

Name	Tracker	Controller	Server	Datastore	Batch	PIF	Member	Options
ROUTOPTS		Y					P	Communication routes to tracker, z-centric agent and remote engine destinations
SERVOPTS			Y				P	Defining options for a server
TCPOPTS	Y	Y	Y	Y	Y	Y	P	Local TCP/IP communication task.
TOPOLOGY			Y		Y		P	Specifying configuration options for end-to-end scheduling with fault tolerance capabilities
TRGOPT		Y					M	Event-driven workload automation support. Used by the Java™ program that creates configuration files for data set triggering
TRROPTS	Y						P	The communication route to the controller
USRREC					Y		M	Defining the user id and password to be used for the operations running on fault-tolerant workstations
XCFOPTS	Y	Y					P	XCF communications

CONFIGURATION

Statement	Keywords	Description
OPCOPTS	OPCHOST	Specifies the subsystem type (tracker, controller, or standby controller).
	NCFTASK	Starts NCF for communication through VTAM.
	EWTRTASK	Starts an event writer to collect events from the z/OS system.
	ERDRTASK	Starts an event reader to transfer events to the event.
	SERVERS	Starts one or more server queues at the controller.
	TPLGYSRV	Starts the IBM Workload Scheduler end-to-end enabler task.
EWTROPTS	EWSEQNO	Event writer also performs event reader function. Separate event reader is not required.
	SUREL	Specifies if the event writer reads a submit/release data set.
ERDROPTS	ERSEQNO	Specifies the event reader and defines the ddname of the input event dataset
	RELDDNAME	Specifies the submit/release data set that release commands are written to in a shared DASD environment.
ROUTOPTS		Identifies routes from the controller to tracker destinations.
TRROPTS		Identifies the route from a tracker to the controller.
XCFOPTS		Identifies XCF connections and specifies when a standby controller performs take over.
SERVOPTS	SUBSYS	Identifies the controller with which the server communicates

SECURITY

Statement	Keywords	Description
AUTHDEF		Specifies how IBM Workload Scheduler for z/OS resources are defined to RACF
AROPTS	AUTHUSER	Specifies where IBM Workload Scheduler for z/OS retrieves a name for authority checking
	USERREQ	Specifies if a valid user ID is required
AUDIT		Specifies when access to IBM Workload Scheduler for z/OS data is recorded
JTOPTS	JOBCHECK	Specifies if the job name in JCL must match the operation job name
USRREC	USRNAM	Specifies the user name.
	USRPSW	Specifies the user password.
SERVOPTS	USERMAP	Defines a member that contains all the associations between DWC and matching RACF user IDs.
TOPOLOGY	LOCALPSW	Specifies if the user ID and password to be used for Windows workstations are to be found locally, when missing from the Symphony file

Determining the success or failure of a job

Statement	Keywords	Description
EWTROPTS	RETCODE	Create job-end (3P) event with highest or last return code
	STEPEVENTS	Specifies when IBM Workload Scheduler for z/OS creates events for ending job-steps
JCCOPTS		Job completion checker actions
NOERROR	LIST	Error codes that are not errors
JTOPTS	NOERROR	Error codes that are not errors
	HIGHRC	Highest return code that is not an error
	ERRRES	Reset operation status to A (arriving) for these error codes

RESTART AND CLEANUP

Statement	Keywords	Description
BATCHOPT	RCLEANUP	Enables maintenance cleanup of local data store
OPCOPTS	RECOVERY	No automatic recovery action will be performed until cleanup actions are complete or discarded (when cleanup type is Immediate)
	RCLEANUP	Starts restart and cleanup tasks
FLOPTS	SNADEST	Specifies the table of tracker and data store destinations used to locate the data store used by the job log when an SNA connection is being used
	XCFDEST	Specifies the table of tracker and data store destinations used to locate the data store used by the job log when an XCF connection is being used
RCLOPTS	CLNJOBPX	Specifies the job name prefix to be used for stand-alone cleanup
	CLNJOBCARD	Specifies the job account information used while creating stand alone cleanup jobs.
	DDALWAYS	Lists the DD names that make step always re-executable
	DDNEVER	Lists the DD names that make step never re-executable
	DDNOREST	Lists the DD names that make step not restartable
	DDPRMEM	Contains the name of the PDS member of the parameter library containing the list of protected DD names
	DDPROT	Lists the DD names that identify protected data sets
	DSNPRMEM	Contains the name of the PDS member of the parameter library containing the list of protected data set names
	DSNPROT	Lists the protected data set names
	DSTCLASS	Specifies a JES class when JCC is used
	DSTDEST	Specifies the destination to be added in the JCL to create a sysout copy for the data store
	DSTRMM	RMM is active and cleanup will use the RMM API
	STEPRESCHK	Specifies the possibility to select a step restart range overriding the product logic checks

JOB LOG RETRIEVAL

Statement	Keywords	Description
OPCOPTS	RCLEANUP	Activates the FL task on the controller to connect to the data store
FLOPTS	SNADEST	Specifies the table of tracker and data store destinations used to locate the data store used by the job log when an SNA connection is used
	XCFDEST	Specifies the table of tracker and data store destinations used to locate the data store used by the job log when an XCF connection is used
	CTLLUNAM	Specifies SNA values to be used for SNA connection to the data store
	DSTGROUP, CTLMEM	Specifies XCF values to be used for XCF connection to the data store

AUTOMATIC JOB RECOVERY

Statement	Keywords	Description
OPCOPTS	RECOVERY	Determines if JCL is checked for RECOVER statements when an operation ends in error
	RCLEANUP	IBM Workload Scheduler for z/OS performs cleanup before recovery starts if clean type is immediate
AROPTS		Specifies recovery options
EWTROPTS	STEPEVENTS	Specifies when IBM Workload Scheduler for z/OS creates events for ending job-steps
	RETCODE	Highest or last return code
JTOPTS	ERRRES	Operation status reset to A for these error codes. Recovery is not performed.
	HIGHRC	Perform recovery only if the return code is greater than HIGHRC
	NOERROR	Error codes that are not errors. Recovery is not performed.
NOERROR		Error codes that are not errors. Recovery is not performed.

WORKLOAD RESTART

Statement	Keywords	Description
JTOPTS	WSFAILURE	System-level actions for workstation failure
	WSOFFLINE	System-level actions for workstation offline
	OPRESTARTDEFAULT	Action if restartable field in operation details is blank
	OPREROUTEDEFAULT	Action if reroutable field in operation details is blank
	OFFDELAY	Elapsed time before offline actions are taken
	PULSE	The time between handshakes for the controller and the trackers. If a tracker does not respond to two successive handshake requests, the controller forces the destination offline.

PERFORMANCE

Statement	Keywords	Description
AUDIT		Specifies how much audit information is produced
ALERTS	LATEOPER	Use only when deadlines are accurate
AUTHDEF	SUBRESOURCES	Specifies subresources that you want to check
EWTROPTS	STEPEVENTS	Specifies when IBM Workload Scheduler for z/OS creates events for ending job-steps
	PRINTEVENTS	Specifies if IBM Workload Scheduler for z/OS creates events for print tasks (type 4)
	HOLDJOB	Specifies if IBM Workload Scheduler for z/OS holds and releases jobs on the JES queue
	EWWAIT	Specifies the time between reads of a submit/release data set
JTOPTS	BACKUP	Specifies if a CP backup is performed automatically and how many records are written to the JT log between backups
	EVELIM	Specifies how often statistics messages related to the STATMSG keyword are issued
	MAXJSFILE	Specifies if a JS backup is performed automatically and how large the JCL repository file grows before IBM Workload Scheduler for z/OS performs a backup
	QUEUELEN	Specifies the number of operations IBM Workload Scheduler for z/OS starts when the workstation analyzer subtask gets the CP lock
	STATIM	Specifies when statistics messages are issued
	STATMSG	Determines if IBM Workload Scheduler for z/OS issues performance statistics for the current plan, event manager, general service, and WSA task
	TRACK	Determines which jobs IBM Workload Scheduler for z/OS tracks
OPCOPTS	RCLEANUP	Specifies if the restart and cleanup function is active. Performance is affected when the user chooses to archive the user sysouts for the majority of the operations.
	VARSUB	Determines which jobs are scanned for JCL variables and directives
TOPOLOGY	LOGLINES	Specifies the maximum number of lines that the Job Log Retriever returns for a single Job Log.

REPORTING

Statement	Keywords	Description
BATCHOPT	DATEFORM	Date format in reports
	DPROUT	The ddname of the file that reports are written to
	HDRS	Character strings used as report headers
	PAGESIZE	Number of lines per page
	PLANHOUR	Start of a plan period for reporting purposes
	PREVRES	Previous period results (the 24 hours before PLANHOUR)
JTOPTS	PLANSTART	Start of a plan period for reporting purposes
RESOURCE	FILTER	Specifies which special resources should be reported on

OUTPUT PROCESSING

Statement	Keywords	Description
EWTROPTS	PRINTEVENTS	Specifies if IBM Workload Scheduler for z/OS creates events for print tasks
JCCOPTS	CHKCLASS	SYSOUT classes that IBM Workload Scheduler for z/OS checks
	JCWAIT	Time JCC waits before rechecking the SYSOUT queue for a job
	MAXDELAY	Maximum time the JCC tries to find SYSOUT
	SYSOUTDISP	Specifies if SYSOUT is held, deleted, or requeued after processing
	UMAXLINE	Specifies how many lines to check in user SYSOUT
	USYSOUT	Specifies if user SYSOUT is scanned
JTOPTS	OUTPUTNODE	Specifies which NJE node that SYSOUT is spooled to is used to create JES2 job-termination events

END TO END 1/2

Statement	Keywords	Description
CPUREC	CPUNAME	Specifies the name of the IBM Workload Scheduler workstation.
	CPUOS	Specifies the host CPU operating system related to the IBM Workload Scheduler workstation.
	CPUNODE	Specifies the node name or the IP address of the CPU.
	CPUTCPIP	Specifies the TCP port number of NETMAN on the current CPU.
	CPUDOMAIN	Specifies the name of the IBM Workload Scheduler domain of the CPU.
	CPUHOST	Specifies the name of the host CPU of the agent.
	CPUACCESS	Specifies the name of the access method.
	CPUTYPE	Specifies the CPU type.
	CPUAUTOLNK	Specifies the autolink between the agent and the domain manager.
	CPUFULLSTAT	Specifies that the link from the domain manager operates in Full Status mode.
	CPURESDEP	Specifies that the agent's Production Control process operates in Resolve All Dependencies mode.
	CPUSERVER	Identifies a server (Mailman) process on the domain manager that sends messages to the agent.
	CPULIMIT	Specifies the number of jobs that can run at the same time in a CPU.
	CPUTZ	Specifies the local timezone of the fault-tolerant workstation.
	CPUUSER	Specifies the default user for the workstation.
	SSLLEVEL	Specifies if the workstation uses SSL authentication when it connects with its domain manager.
	SSLPORT	Defines the port used to listen for incoming SSL connections.
	FIREWALL	Specifies if the communication between a workstation and its domain manager must cross a firewall.
SERVOPTS	TPLGYPRM	Defines a member in the file identified by the EQQPARM DD statement in the server startup job. The member contains the fault-tolerant end-to-end options defined by the TOPOLOGY statement. It is used to activate the end-to-end scheduling with fault tolerance capabilities in the server.
	PROTOCOL	Identifies the types of communication used by the server.
	SUBSYS	Identifies the controller for which this server is started.

END TO END 2/2

Statement	Keywords	Description
TOPOLOGY	BINDIR	Specifies the base file system directory where binaries, catalogs, and other files are installed and shared among subsystems.
	CODEPAGE	Specifies the name of the host codepage.
	HOSTNAME	Specifies the hostname or the IP address that will be used by the server in the end-to-end with fault tolerance capabilities environment.
	LOCALPSW	Specifies if the user ID and password to be used for Windows workstations are to be found locally, when missing from the Symphony file.
	LOGLINES	Specifies the maximum number of lines that the job log retriever returns for a single job log.
	PLANAUDITLE	
	VEL	Enables or disables plan auditing for distributed agents.
	PORTRNUMBER	Defines the TCP/IP port number used by the server to communicate with the distributed agents.
	SSLLEVEL	Specifies if the server uses SSL authentication.
	SSLPORT	Defines the port used to listen for incoming SSL connections on the server.
	TCPIPJOBNAME	
	E	Specifies the TCP/IP started-task name used by the server.
	TIMEZONE	Local time zone in the z/OS system where the controller runs.
	TPLGYMEM	Specifies the PARMLIB member where the domain and CPU definition are.
	TRCDAYS	Specifies the number of days the trace files are kept before being deleted.
	USRMEM	Specifies the PARMLIB member where the user definitions are.
	WRKDIR	Specifies the location of the files of a subsystem.
USRREC	USRCPU	Identifies the workstation on which the user can launch jobs. It is valid only on Windows workstations.
	USRNAM	Specifies the user name. It is valid only on Windows workstations.
	USRPWD	Specifies the user password. It is valid only on Windows workstations.

JOB DEFINITIONS

Statement	Keywords	Description
JOBREC	JOBSCR	Specifies the name of the shell script or executable file to run
	JOBCMD	Specifies the name of the shell command to run
	JOBUSR	Specifies the name of the user submitting the script or command
	INTRACTV	Specifies that a Windows job runs interactively on the Windows desktop
	RCCONDSUC	Specifies a Boolean expression which determines the return code (RC) required to consider a job successful.
RECOVERY	OPTION	Specifies the action that IBM Workload Scheduler for z/OS must take when a job ends unexpectedly
	MESSAGE	Specifies the text of a recovery prompt
	JOBCMD	Specifies the name of the shell command to run if the job abends
	JOBSCR	Specifies the name of the shell script or executable file to be run if the job abends
	JOBUSR	Specifies the name of the user submitting the recovery job action
	JOBWS	Specifies the name of the workstation
	INTRACTV	Specifies that the recovery job runs interactively on a Windows desktop
	RCCONDSUC	Specifies a Boolean expression which determines the return code required to consider a recovery job successful.
VARSUB	TABLES	Identifies the variable tables that must be searched, and the search order
	PREFIX	Specifies a non-alphanumeric character that precedes a variable
	BACKPREF	Specifies a non-alphanumeric character that delimits a variable to form simple and compound variables
	VARFAIL	Specifies whether IBM Workload Scheduler for z/OS is to issue an error message when a variable substitution error occurs
	TRUNCATE	Specifies if keywords are to be truncated

WLM

Statement	Keywords	Description
OPCOPTS	WLM	Provides information about the WLM service class integration function (class name, policy name).
	SECHECK	Specifies if and how integration with the WLM scheduling environment is to be activated.
	JESPLEX	Specifies the list of systems comprising the JESplex to which the tracker belongs.
	SYSPLEXID	Specifies the number identifying the sysplex to which the tracker belongs.
	SUPPRESSENF	Specifies if activation of the ENF 57 and ENF 41 listener exits is to be suppressed.

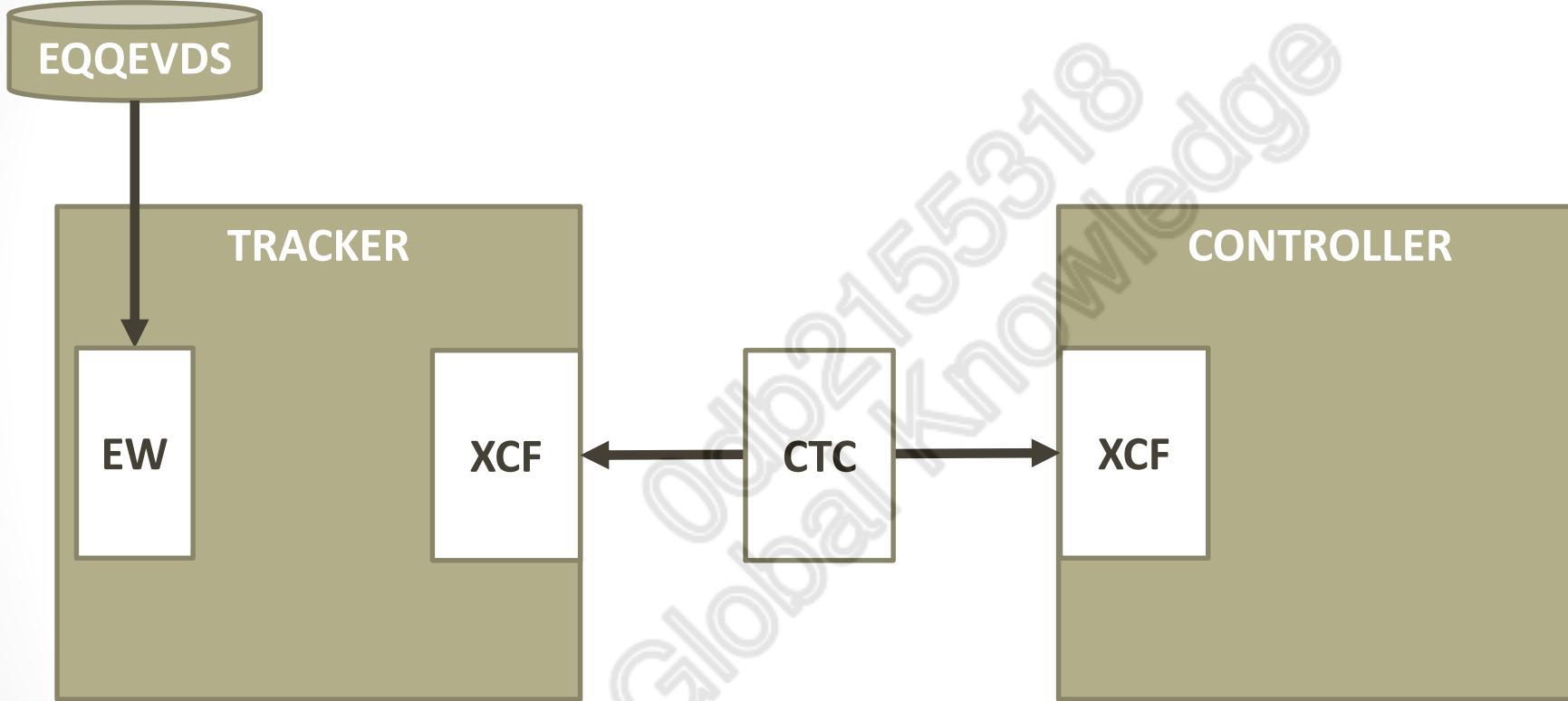
EXTERNAL MONITORING

Statement	Keywords	Description
ALERTS	MONALERT	Defines the conditions under which a generic alert will be sent to IBM Tivoli Monitoring.
	MONOPER	This parameter determines whether the error conditions specified by the MONALERT keyword will be in effect for monitored jobs only or for all jobs. It is used with IBM Tivoli Monitoring.
OPCOPTS	EXTMON	Specifies if Integration with Tivoli Business Systems Manager is enabled.
	CODEPAGE	Specifies the host code page to be used for the data collected by the monitoring task
ALERTS	MONALERT	Defines the conditions under which a generic alert will be sent to IBM Tivoli Monitoring.
	MONOPER	This parameter determines whether the error conditions specified by the MONALERT keyword will be in effect for monitored jobs only or for all jobs. It is used with IBM Tivoli Monitoring.
OPCOPTS	EXTMON	Specifies if Integration with Tivoli Business Systems Manager is enabled.
	CODEPAGE	Specifies the host code page to be used for the data collected by the monitoring task
MONOPTS	MONHOSTNAME	Identifies the host name or IP address of the remote monitoring application. This parameter is used for the integration with the IBM Tivoli Monitoring product.
	MONPORT	Specifies the port number of the remote monitoring application. It is used for the integration with IBM Tivoli Monitoring.
	LOCHOSTNAME	Specifies the local host name or IP address that will be used to communicate with IBM Tivoli Monitoring.
	LOCPORT	Specifies the local port number used by the controller to communicate with IBM Tivoli Monitoring.
	BULKDISC	Defines if and how the bulk discovery is to be performed. This keyword is used for the integration with IBM Tivoli Monitoring.
MONPOL	CONN TIMEOUT	Defines the connection establishment timeout to be used when communicating with IBM Tivoli Monitoring.
	OPERATION	Specifies the types of operations that will be automatically selected for monitoring by IBM Tivoli Monitoring and Tivoli Business Systems Manager.

Configuration sample

- A controller and a tracker connected via XCF
- A controller and a tracker connected via TCPIP
- A controller and a datastore connected via XCF
- A z/OS SYSPLEX
- An Hybrid architecture between IWS for z/OS and IWS Distributed
- z/Centric - z/OS Agent
- DWC

XCF Connection



XCF Configuration

T93C CONTROLLER MEMBER

OPCOPTS OPCHOST(YES) ERDRTASK(0)

ROUTOPTS XCF(T93T)

XCFOPTS

GROUP(T93CC2T)

MEMBER(T93C)

T93T TRACKER MEMBER

OPCOPTS OPCHOST(NO) ERDRTASK(0)
EWTRTASK(YES) EWTRPARM(T93TEWTR)

TRROPTS HOSTCON(XCF)

XCFOPTS GROUP(T93CC2T)
MEMBER(T93T)

T93TEWTR

EWTROPTS EWSEQNO(01)

XCF GROUP

D XCF,GRP,T93CC2T

RESPONSE=SOW1

IXC332I 08.01.54 DISPLAY XCF 110

GROUP T93CC2T: T93C T93T

Odb2155318
Global Knowledge

XCF CURRENT PLAN WORKSTATION STATUS

```
----- BROWSING IWSz TRACKER SYSTEM INFORMATION -----
Command ===>

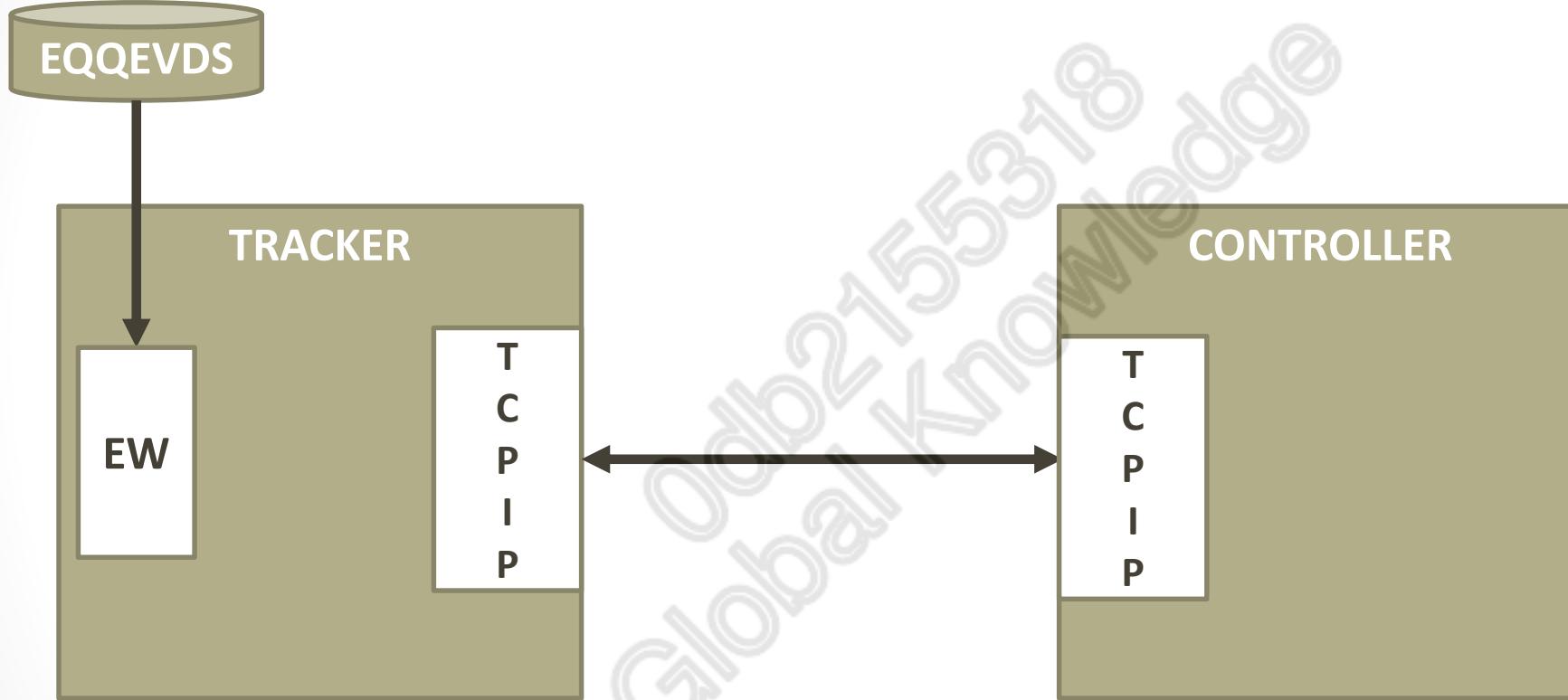
Work station          : XCF0           XCF TRACKER
FT Work Station       : No
Destination           : T93T           XCF
Events received from : T93T           T93T

System Information:
Operating system       : MVS/ESA        SP7.2.2
Job entry subsystem   : JES2           HJE77A0
NJE node name         : S0W1
IWSz tracker FMID    : HWSZ930

Tracker Functions:
Submittor              : Active
JCC                    : Active
RODM                  : Not Active

Tracking Options:
Stepevents             : ALL            Holdjob      : USER
Event writer retcode   : HIGHEST        Printevents: ALL
```

TCPIP Connection



TCPIP Configuration

T93C CONTROLLER MEMBER

OPCOPTS OPCHOST(YES) ERDRTASK(0)

ROUTOPTS

TCPIP(T931:'ZOS220'/427)

TCPOPTS

HOSTNAME(ZOS220)

TRKPORTNUMBER(424)

T931 TRACKER MEMBER

OPCOPTS OPCHOST(NO) ERDRTASK(0)
EWTRTASK(YES) EWTRPARM(T931EWTR)

TRROPTS HOSTCON(TCP)

TCPHOSTNAME(ZOS220)

TCPPORTNUMBER(424)

TCPOPTS

TRKPORTNUMBER(427)

T93TEWTR

EWTROPTS EWSEQNO(01)

TCPIP CURRENT PLAN WORKSTATION STATUS

```
----- MODIFYING A WORK STATION IN THE CURRENT PLAN -----
Command ===>

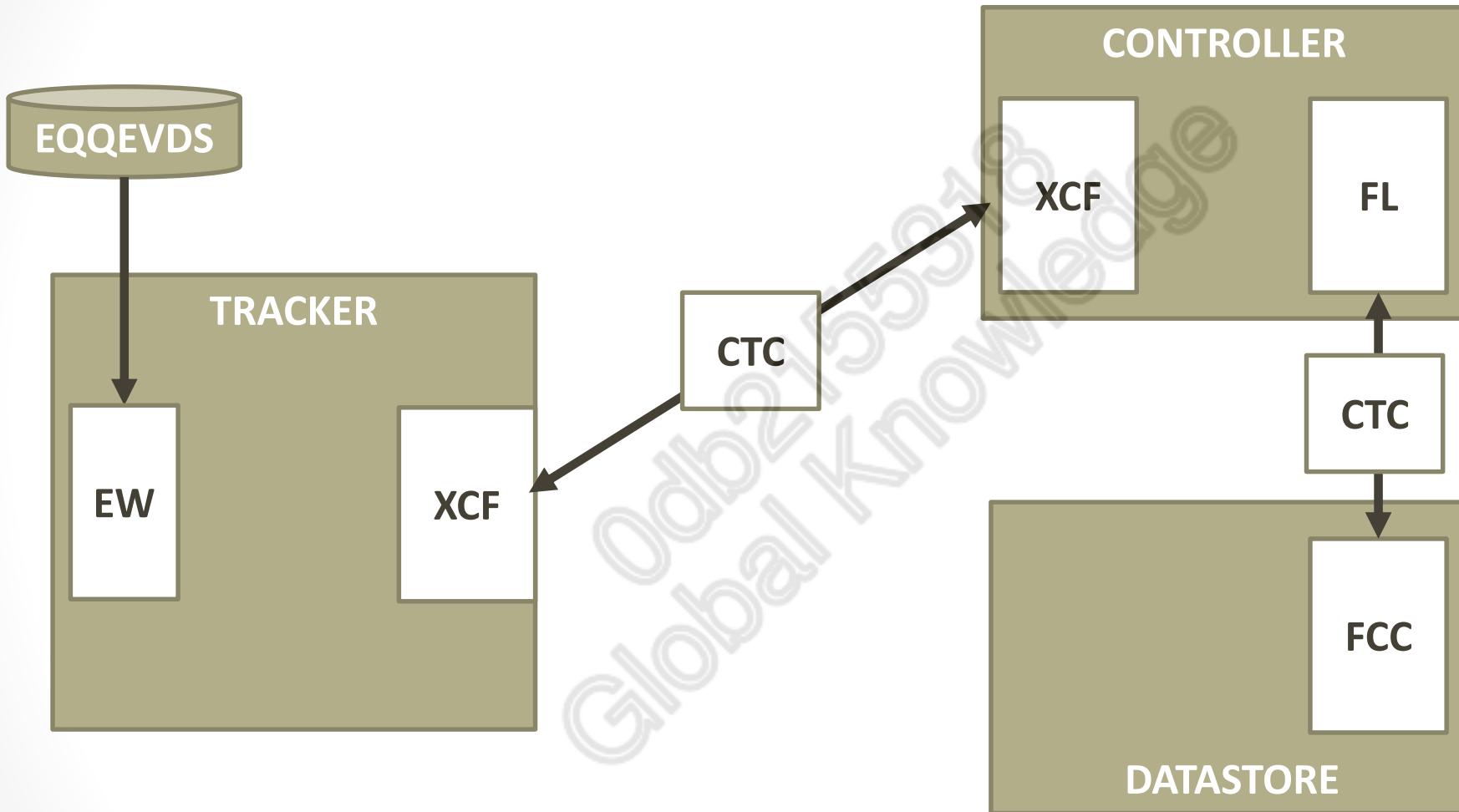
Enter/change data below:
Enter the O command above to modify open time intervals
Enter the V command above to vary work station status

Work station      : TCP0          TCPIP TRACKER
Type              : Computer     JOB ability
Status            : Active

REPORTING ATTR    ===> A      A automatic, S manual start and complete
                           C complete only or N non reporting

CONTROL ON SERVERS ===> Y      Control on parallel servers, Y or N
CONTROL ON R1      ===> N      Control on critical resource R1 , Y or N
CONTROL ON R2      ===> N      Control on critical resource R2 , Y or N
```

XCF Connection with Datastore



DATASTORE Configuration

T93C CONTROLLER MEMBER

OPCOPTS OPCHOST(YES) ERDRTASK(0)

RCLEANUP(YES)

ROUTOPTS XCF(T93T)

XCFOPTS

GROUP(T93CC2T)

MEMBER(T93C)

FLOPTS

DSTGROUP(T93CC2D)

CTLMEM(T93C)

XCFDEST(T93T.T93D)

T93T TRACKER MEMBER

OPCOPTS OPCHOST(NO) ERDRTASK(0)
EWTRTASK(YES) EWTRPARM(T93TEWTR)

TRROPTS HOSTCON(XCF)

XCFOPTS GROUP(T93CC2T)
MEMBER(T93T)

T93D DATASTORE MEMBER

DSTOPTS

HOSTCON(XCF)

DSTGROUP(T93CC2D)

DSTMEM(T93D)

CTLMEM(T93C)

DATASTORE LOG RETRIEVAL

```
----- MODIFYING OPERATIONS IN THE CURREN          JOBLOG REQUESTE
Command ====>                                         Scroll ====> CSR

Enter the GRAPH command above to view operations graphically or
change data in the rows, and/or enter any of the following row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
J - Edit JCL, O - Browse operator instructions, S - Modify operation details
L - Browse joblog, LJ - Browse joblog via ITOM

Application      : TCP000
Owner            : TCP000
Input arrival    : 18/03/03 10.00
Status           : Completed

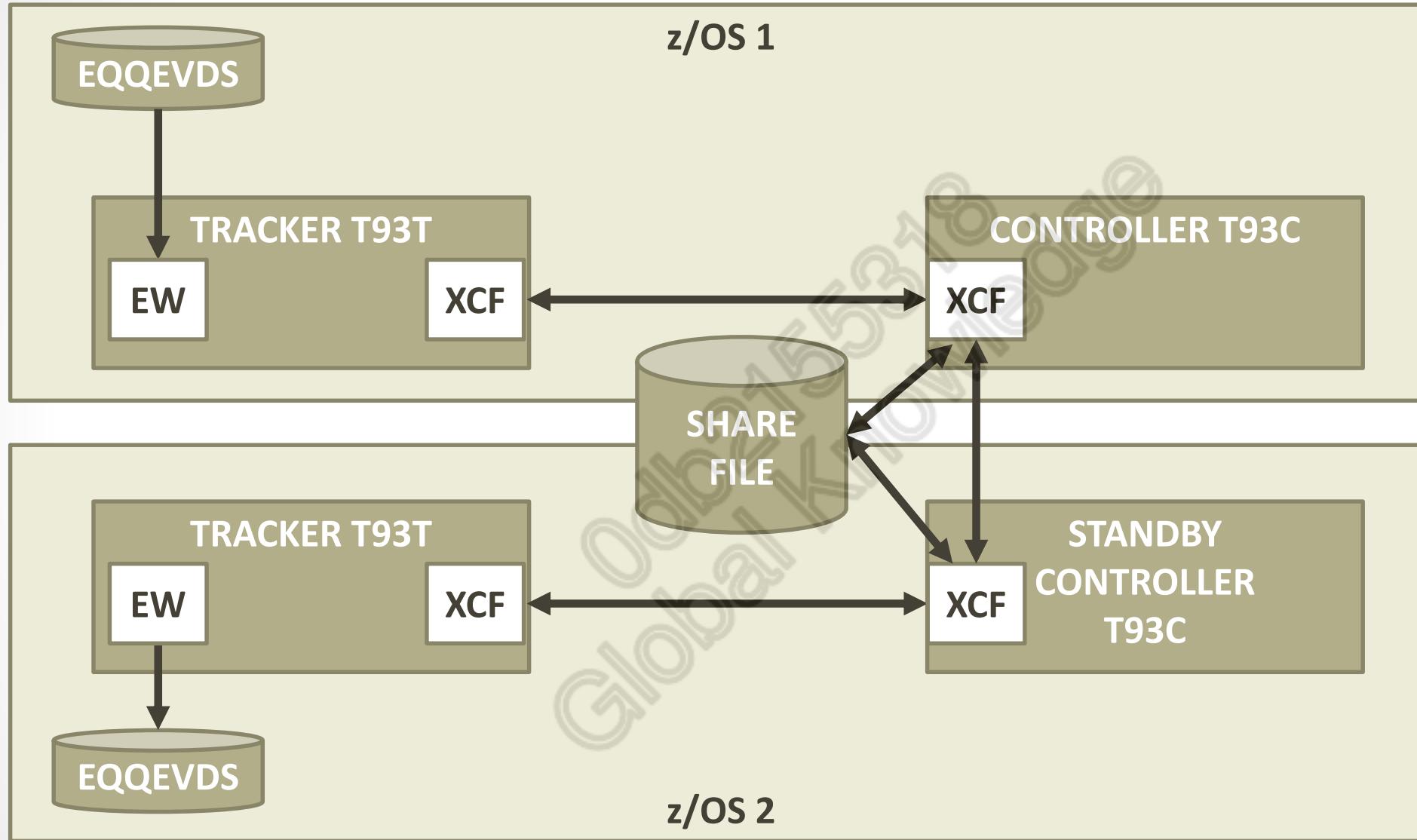
Row Operation          Jobname PS Duration Opt Dep Res St
cmd ws   no. text          HH.MM.SS S T S/P S R1 R2 N
*** TCP0 001      TCP00000 1 00.00.01 Y N     0 0
***** Bottom of data *****
```

```
----- JES2 JOB LOG -- SYSTEM SYS1 -- N
Menu Utilities Compilers Help
BROWSE  SYS18062.T095517.RA000.TWS930.OPCAWRK1.H Li
Command ====> _____ Top of Data _____
$EQQFSWWU-START-JESMSG
J E S 2 J O B L O G -- S Y S T E M S Y S 1 -- N
09.53.58 JOB05722 ---- SATURDAY, 03 MAR 2018 ----
09.53.58 JOB05722 IRR0101 USERID START1 IS ASSIGNED TO THIS JOB.
09.53.59 JOB05722 ICH700011 START1 LAST ACCESS AT 09:45:43 ON SATURDAY, MAR
09.53.59 JOB05722 $HASP373 TCP00000 STARTED - INIT 1 - CLASS A - SY
09.54.00 JOB05722 IEF403I TCP00000 - STARTED - TIME=09.54.00
09.54.00 JOB05722 - -TIMINGS (MINS.)
09.54.00 JOB05722 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB CLO
09.54.00 JOB05722 -TCP00000 RETWT1 00 8 .00 .00 .
09.54.00 JOB05722 IEF404I TCP00000 - ENDED - TIME=09.54.00
09.54.00 JOB05722 -TCP00000 ENDED. NAME-TWS930 TOTAL CPU TIME=
09.54.00 JOB05722 $HASP395 TCP00000 ENDED - RC=0000
----- JES2 JOB STATISTICS -----
03 MAR 2018 JOB EXECUTION DATE
      5 CARDS READ
      40 SYSOUT PRINT RECORDS
      0 SYSOUT PUNCH RECORDS
```

03/03 09.54.34 EQQM923I JOBLOG FOR TCP00000 (JOB05722) ARRIVED CN(INTERNAL)

03/03 09.54.34 EQQM923I FOR OPERATION: TCP000 , 1803031000, 001 CN(INTERNAL)

z/OS SYSPLEX



SSYPLEX Configuration

T93C CONTROLLER MEMBER

OPCOPTS OPCHOST(HOST/PLEX)

XCFOPTS

GROUP(T93CC2T)

MEMBER(T93C)

T93C STANDBY CONTROLLER MEMBER

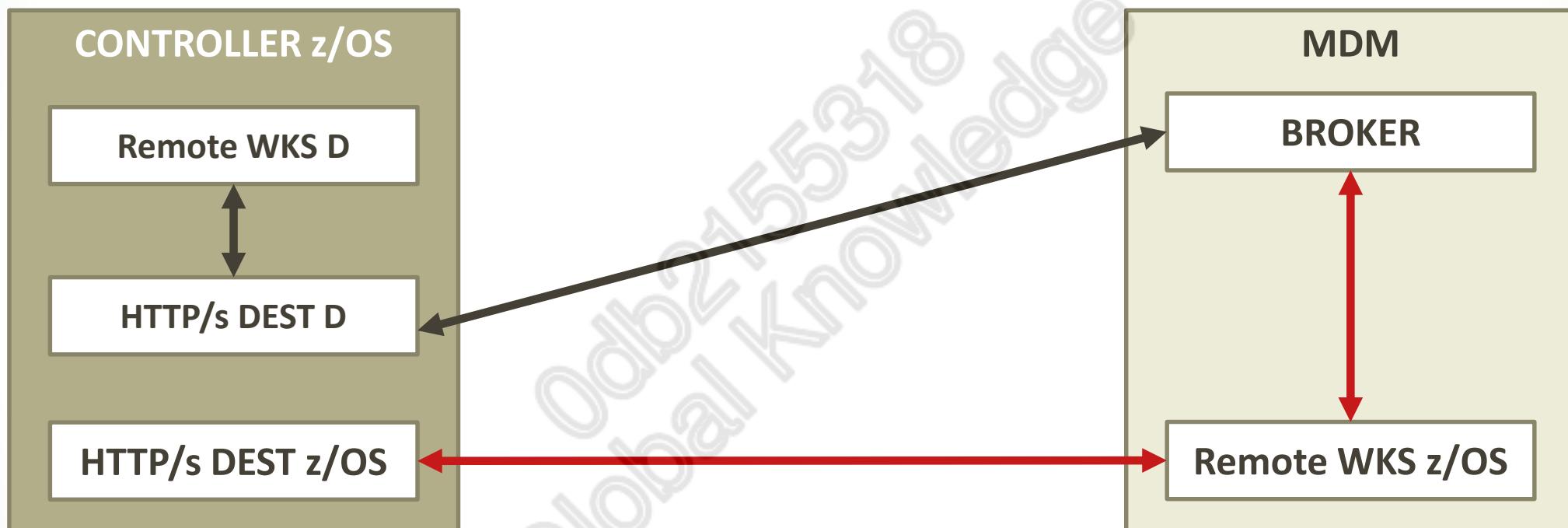
OPCOPTS OPCHOST(STANDBY/PLEX)

XCFOPTS

GROUP(T93CC2T)

MEMBER(T93B)

Hybrid architecture : Cross dependencies

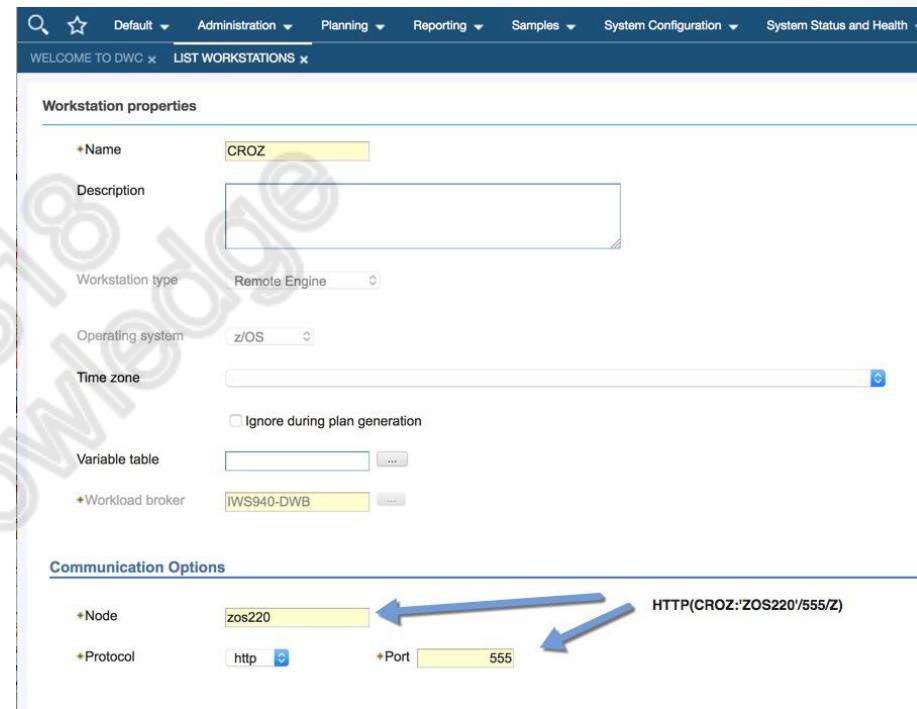


Hybrid z/OS Remote Workstation

T93C CONTROLLER MEMBER

ROUTOPTS

HTTP(CROZ:'ZOS220'/555/Z)



Hybrid Distributed Remote Workstation

T93C CONTROLLER MEMBER
ROUTOPTS
HTTP(CROD:'RHEL73F'/41114/D)

The screenshot shows the 'Workstation properties' section of the T93C Controller Member interface. It includes fields for Name (IWS940-DWB), Domain (MASTERDM), Description (This workstation was automatically created.), Workstation type (Workload Broker), Time zone, Variable table (ROUTOPTS HTTP(CROD:'RHEL73F'/41114/D)), and Communication Options (Node name (localhost), TCP/IP port (41114)). A red arrow points from the 'HTTP(CROD:'RHEL73F'/41114/D)' text to the corresponding entry in the variable table.

```
----- MODIFYING GENERAL INFORMATION ABOUT A WORK STATION -----
Command ===>
Enter the command R for resources A for availability O for end-to-end option
or D for Destinations above, or enter data below:
More:

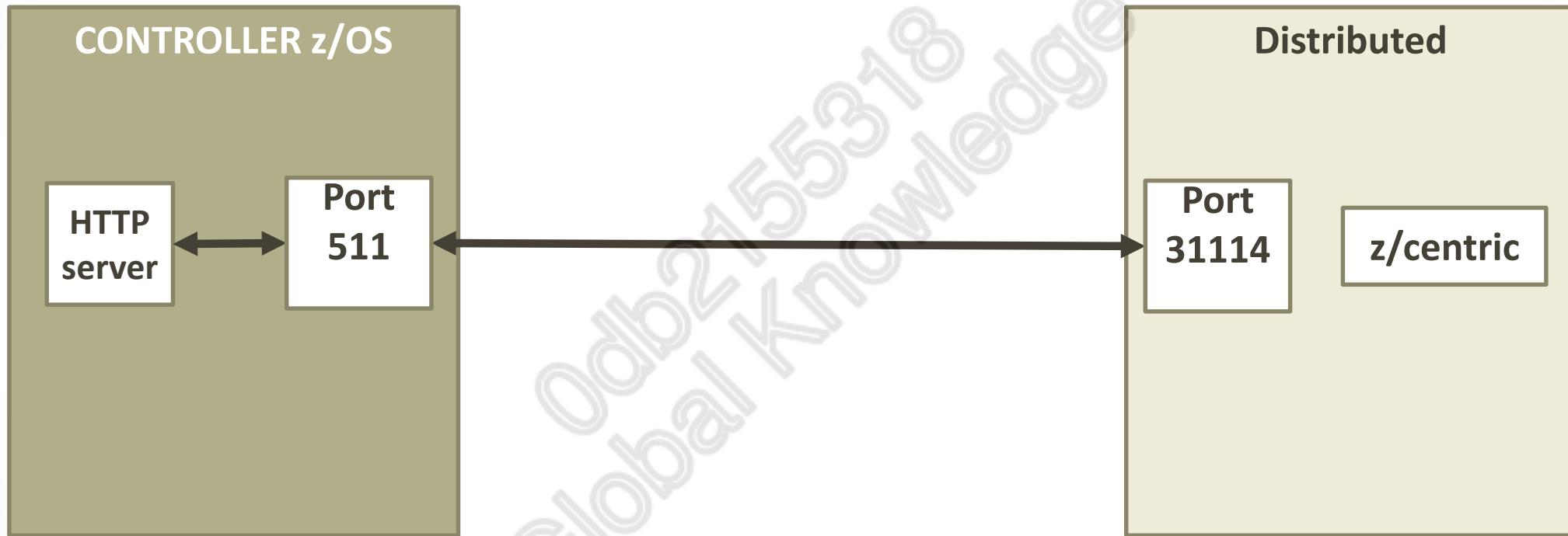
Work station name      : CROD
DESCRIPTION             ===>
WORK STATION TYPE       ===> R

REPORTING ATTR          ===> A

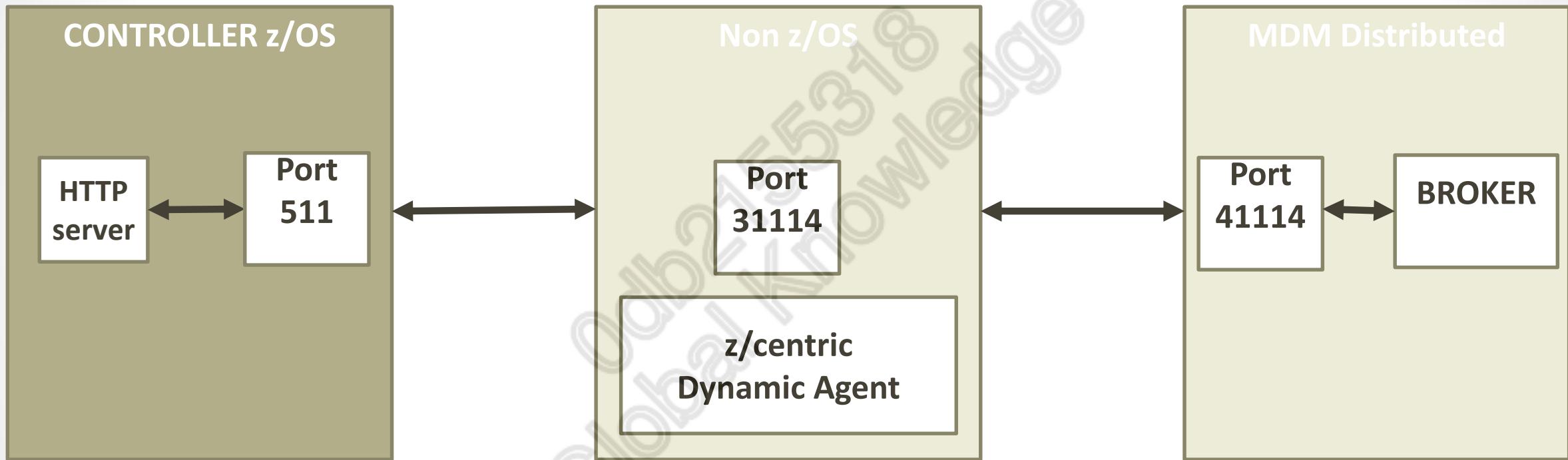
PRINTOUT ROUTING        ===> SYSPRINT
SERVER USAGE             ===> N
DESTINATION              ===> CROD
Options: allowed Y or N
SPLITTABLE               ===> N
STARTED TASK, STC       ===> N
AUTOMATION                ===> N
WAIT                      ===> N
VIRTUAL                  ===> N
JOB SETUP                 ===> N
WTO                      ===> N
FAULT-TOLERANT AGENT    ===> N
Z-CENTRIC AGENT          ===> N
DYNAMIC                  ===> N

REMOTE ENGINE TYPE       ===> D
Defaults:
TRANSPORT TIME           ===> 00.00
Time from previous work station HH.MM
```

z/centric – z/OS Agent



z/centric – dynamic agent



z/centric

T93C CONTROLLER MEMBER

ROUTOPTS

HTTP(Z930:'RHEL73F'/31114)

HTTPOPTS

HOSTNAME(ZOS220)

HTTPPORTNUMBER(511)

```
----- BROWSING IWSz Z-CENTRIC SYSTEM INFORMATION -----
Command ===>

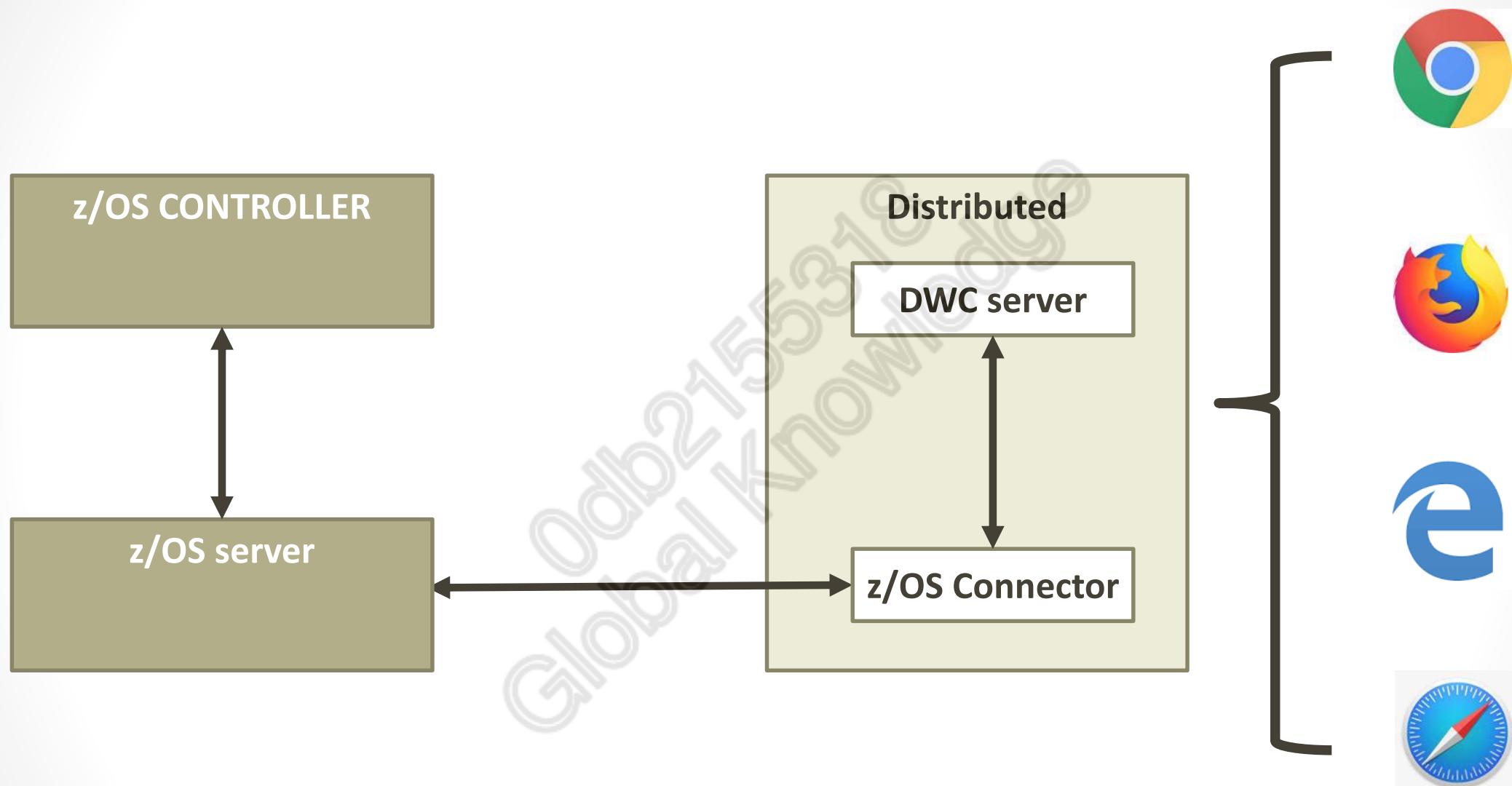
Work station      : Z930          z/centric
Destination       : Z930          HTTP
Code page         : IBM-1147
Hostname          : RHEL73F
Port number       : 31114

System Information:
Operating system name : LINUX_X86_64
Operating system level: 1 SMP Mon Feb 2
Agent version       : 9.4.0
Agent fixpack level: 0

Default JOBREC values:
JOBUSR   :
JOBPWD   :
JOBTYPE  :

User name
Password (Y/N/A
Access Method
```

DWC



DWC z/OS parameter

T93C CONTROLLER MEMBER

T93W SERVER MEMBER

SERVOPTS

SUBSYS(**T93C**)

PROTOCOL(TCP)

USERMAP(**T93WUSER**)

JSCHOSTNAME(ZOS220)

PORTNUMBER(5930)

CODEPAGE(IBM-1147)

INIT CALENDAR (DEFAULT)

USER 'IWS940à*' RACFUSER(TWS930)
USER 'user01à*' RACFUSER(EXPL30)

If no USERMAP then RACF TMEDADMIN CLASS

DWC non z/OS parameter

On TWAUI/wastools directory

```
createZosgengine.sh
-name T93C ←
-hostName zos220
-portNumber 5930
```

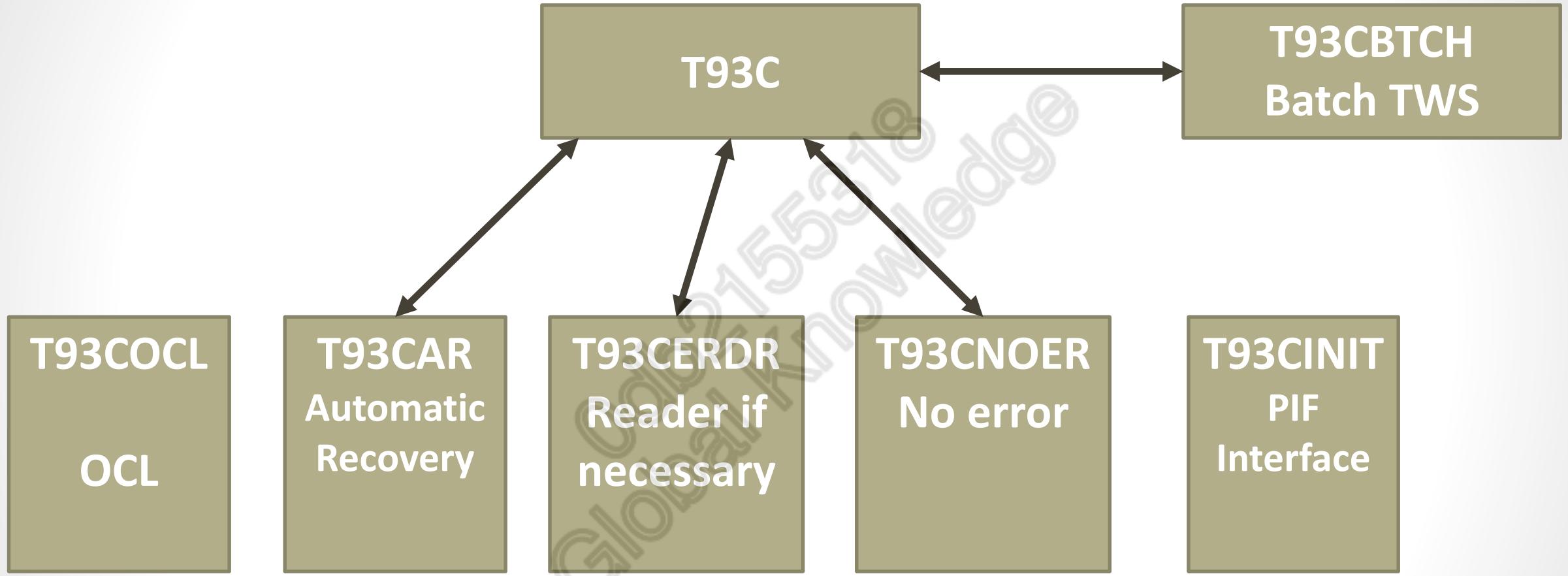
The screenshot shows the 'Manage Engines' section of the DWC interface. At the top, there are buttons for New, Connection Properties, Delete, Share, Unshare, and Test Connection. Below is a table listing two engines:

Name	Type	Version	Host	Port	Remote Server	Default Plan	Owner	Share
mdm940	Distributed		localhost	31116		Current Plan	iws940	None
T93C	z/OS		localhost	16312	T93C	Current Plan	iws940	None

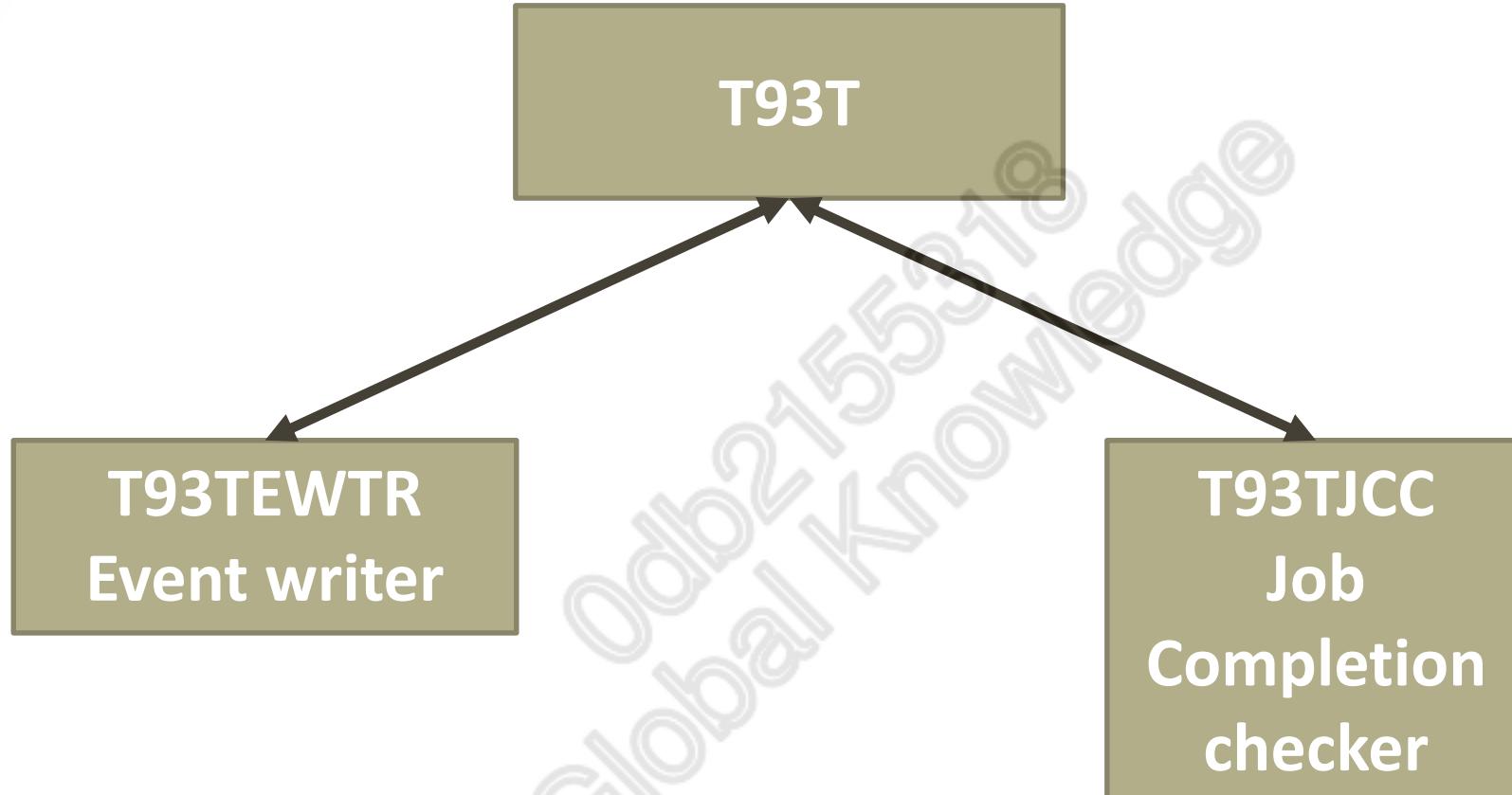
The screenshot shows the 'Engine Connection Properties' dialog box. It has several sections:

- Information:** Engine Name: T93C
- Connection Data:** Engine Type: z/OS, Host Name: localhost, Port Number: 16312, Remote Server Name: T93C
- Connection Credentials:** User ID: iws940, Password: [REDACTED], Share credentials:
- Plan:** Default Plan: Current Plan, Select...
- Database Configuration for Reporting:** Enable Reporting: Database User ID: [REDACTED], Password: [REDACTED]
- Dashboard:** Show in dashboard:

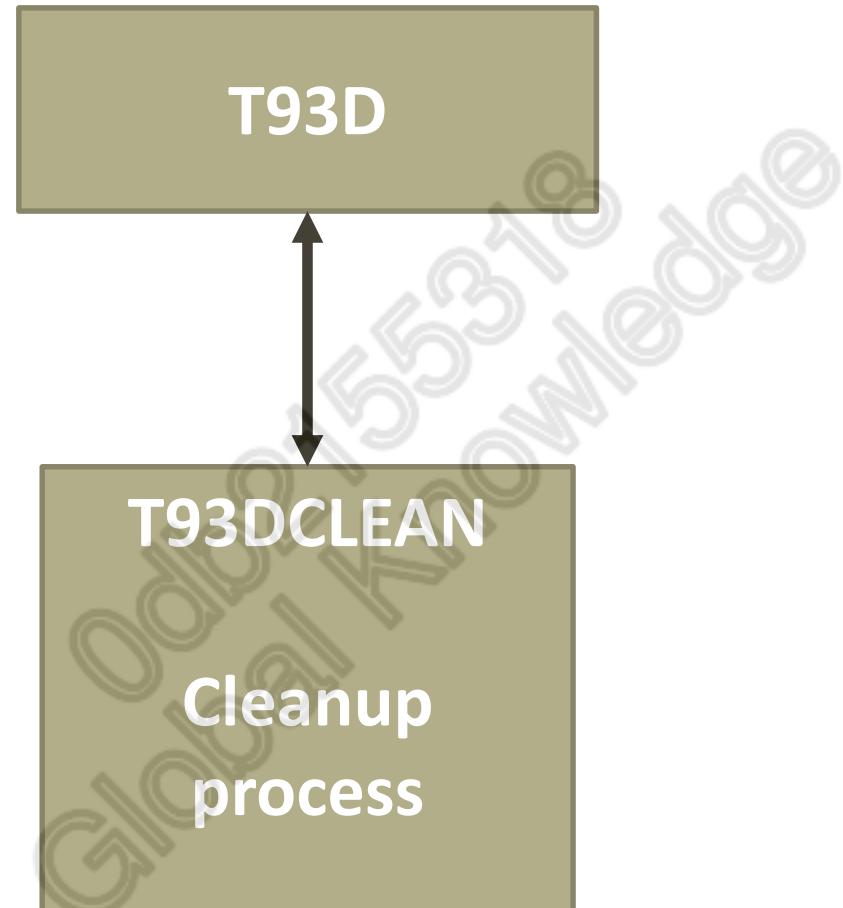
Controller PARMLIB



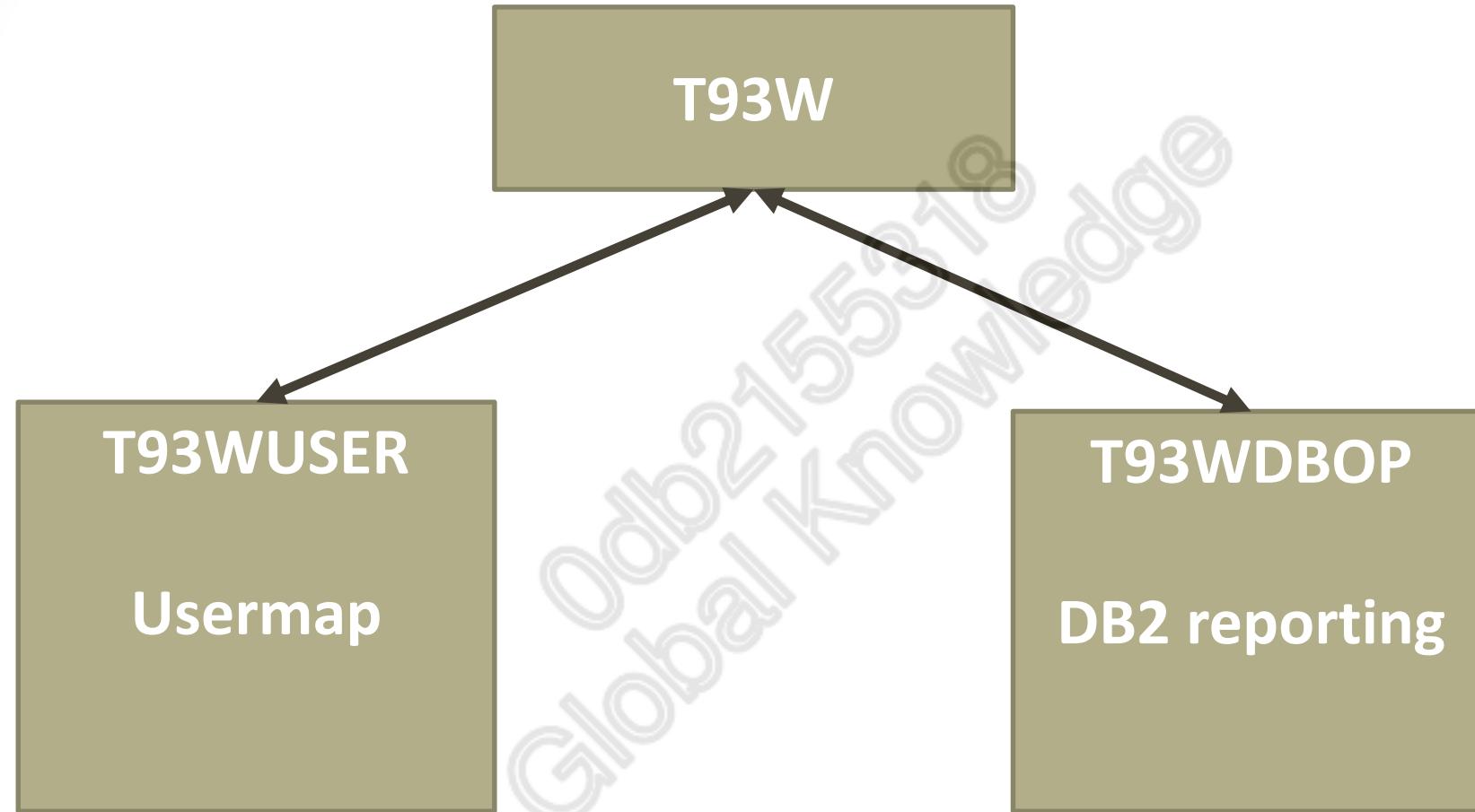
Tracker PARMLIB



Datastore PARMLIB



TCPIP server PARMLIB



IWS Disaster Recovery Procedure

Odb25318
Global Knowledge

Objective

In this chapter , you will learn about the architecture of IWS product

After completing this training, you should be able to:

- Manage the save of File
- Organize your production in case of problem

z/OS Architecture : Controller File

- Database file
- Current Plan
- Long Term Plan
- End To End scheduling
- Event
- JCL Repository
- Job Tracking
- DataStore

Odb2155318
Global Knowledge

Controller File : Database

DATABASE	Physical VSAM Dataset	DD NAME	Update	Dump
Application Description (1.4)	AD	EQQADDS	Daily	Y
Calendar (1.2)	WS	EQQWSDS	Yearly	Y
Job Descriptions (1.8)	AD	EQQADDS	Daily	Y
Operator Instruction (1.5)	OI	EQQOIDS	Daily	Y
Period (1.3)	WS	EQQWSDS	Weekly	Y
Special resource (1.6)	RD	EQQRDDS	Daily	Y
Side Information (ETT) (1.7)	SI	EQQSIDS	Daily	Y
Variable Table (1.9)	AD	EQQADDS	Weekly	Y
Workstation Description (1.1)	WS	EQQWSDS	Monthly	Y
Run Cycle Group (1.10)	WS	EQQWSDS	Weekly	Y

Controller File : Current Plan

Current Plan	Physical VSAM Dataset	DD NAME	Update	Dump
Primary Current Plan	CP1	EQQCP1DS	Continuous	N
Alternate Current Plan	CP2	EQQCP2DS	Continuous	N
New Current Plan	NCP	EQQNCPDS	Daily	N
New Current Plan Extension	NCX	EQQNCXDS	Continuous	N
Current plan Extension	CX	EQQCXDS	Continuous	N
Primary extended Data	XD1	EQQXD1DS	Continuous	N
Alternate extended data	XD2	EQQXD2DS	Continuous	N
New Extended data	NXD	EQQNXDDS	Continuous	N

Controller File : Long Term Plan

Long Term Plan	Physical VSAM Dataset	DD NAME	UPDATE	DUMP
Long Term Plan	LT	EQQLTDS	<ul style="list-style-type: none">• SOMETIMES• During the LTP	DAILY

Controller File : End To End scheduling

End To End scheduling	Physical Dataset	DD NAME	UPDATE	DUMP
Centralized script dataset for end-to-end scheduling with fault tolerance capabilities	CS (PDS)	EQQTWSCS	DAILY	Y
Input Event data sets for end-to-end scheduling with fault tolerance capabilities	WSIN (SEQ)	EQQWSIN	CONTINOUS	N
Output Event data sets for end-to-end scheduling with fault tolerance capabilities	WSOUT (SEQ)	EQQWSOUT	CONTINOUS	N
Script library for end-to-end scheduling with fault tolerance capabilities	SCLIB (PDS)	EQQSCLIB	DAILY	Y
Current plan backup for the creation of Symphony	SCP (VSAM)	EQQSCPDS	CONTINOUS	N

Controller File : Event

Event	Physical VSAM Dataset	DD NAME	UPDATE	DUMP
Event Log	EV	EQQEVD\$	CONTINUOUS	N
Event for Event reader (01-16)	EVxx	EQQEVDxx	CONTINUOUS	N

Controller File : JCL

JCL Repository	Physical VSAM Dataset	DD NAME	UPDATE	DUMP
Primary JCL repository	JS1	EQQJS1DS	CONTINUOUS	N
Secondary JCL repository	JS2	EQQJS2DS	CONTINUOUS	N
JCL Library	JBLIB	EQQJBLIB	DAILY	Y
PROC	PRLIB	EQQPRLIB	DAILY	Y
PARMLIB	PARM	EQQPARM	SOMETIMES	Y

Controller File : Job Tracking

JCL Repository	Physical VSAM Dataset	DD NAME	UPDATE	DUMP
Job Tracking Log	JTxx	EQQJTxx	Current Plan turnover	N
Dual Job Tracking Log	DLxx	EQQDLxx	Current Plan turnover	N
JT archive data set	JTARC	EQQJTARC	<ul style="list-style-type: none">• Current Plan turnover• New Current Plan	N

Controller File : DataStore

JCL Repository	Physical VSAM Dataset	DD NAME	UPDATE	DUMP
Structured data files	SDFxx	EQQSDFxx	CONTINUOUS	N
Primary index	PKIxx	EQQPKIx	CONTINUOUS	N
Unstructured data files	UDFxx	EQQUDFxx	CONTINUOUS	N
Secondary index	SKIxx	EQQSKIxx	CONTINUOUS	N

- Structured data file is for JESJCL,JESYSMSG, JESMSGLG
- Unstructured data file is for SYSOUT

EQQWSDS

- It contains the Workstation, calendar, period, Run cycle group
- Must be saved daily
 - Using WAPL for Workstation, Calendar, Period
 - Using EQQYCAIN for Run Cycle Group
 - Using DFDSS
- Used by
 - Controller
 - Daily Planning job
 - Long Term Planning job

EQQADDS

- It contains the applications, groups and variable table
- Must be saved daily
 - Using WAPL for variable table
 - Using EQQYCAIN for Applications and group
 - Using DFDSS
- Used by
 - Controller
 - Daily Planning
 - Long Term Planning Job

EQQOIDS

- It contains the operator instructions
- Must be saved daily
 - Using EQQYCAIN for Applications and group
 - Using DFDSS
- Used by
 - Controller

EQQRDDS

- It contains the Special Resource
- Must be saved daily
 - Using WAPL
 - Using DFDSS
- Used by
 - Controller
 - Daily Planning job

Odlb2155318
Global Knowledge

EQQSIDS

- It contains the ETT
- Must be saved daily
 - Using WAPL
 - Using DFDSS
- Used by
 - Controller

EQQSIDS

- It contains the ETT
- Must be saved daily
 - Using WAPL
 - Using DFDSS
- Used by
 - Controller

EQQLTDS / EQQLTBKP / EQQLDDDS

- It contains the
 - The LTP LTDS
 - The Backup LTP LTBKP
 - The LTP Work
 - Using DFDSS
 - Used by
 - Controller
 - Daily Planning Job
 - Long Term planning job
- Must be saved Daily

Current Plan turnover process (1/7)

- A daily planning job starts and is recognized by the normal-mode-manager subtask
- The active current plan is backed up
- The daily planning job builds a new current plan
- The new current plan is taken over
- The newly created active current plan is backed up
- The Symphony File is updated
- The Symphony File is distributed

Current Plan turnover process (2/7)

A daily planning job starts and is recognized by the normal-mode-manager subtask

1. A daily planning batch job starts, signals the Controller using an ENQ, and then waits.
2. NM detects the ENQ from the batch job and locks the current plan preventing more updates.
3. The active current plan is updated with the latest information from in-storage control blocks representing workstations and active operations. The extension file (CX), which is held in a data space, is refreshed to DASD.

Current Plan turnover process (3/7)

The active current plan is backed up

4. The inactive current plan is erased.
5. The active current plan is copied to the inactive current plan. They are now identical.
6. The inactive current plan becomes the active current plan, and the previously active becomes the inactive.
7. The JT logs are switched
8. The current plan is unlocked, and normal processing continues.
9. NM signals to the batch job that backup processing is complete.
10. The data from the inactive JT log is appended to the JT archive log (EQQJTARC).

Current Plan turnover process (4/7)

The daily planning job builds a new current plan

11. The batch job starts executing again. The inactive current plan is used (together with the LTP, AD, RD, and WS for a CP extend) as input, and a new current plan is created in the NCP, NCX, and NXD data sets. While the batch job is building the new current plan, IWS continues normal processing except that a current plan backup is not permitted because the batch job is using the inactive current plan data set.
12. The contents of the job-tracking archive data set are copied to the data set in the daily planning job referenced by the EQQTROUT DD name.
13. When the new current plan is ready, the checkpoint data set is updated to show if the new current plan was successfully created. The NM subtask is notified that the daily planning process has completed.
14. The subsystem investigates the checkpoint data set to see if a new current plan was successfully created. If not IWS assumes a daily planning process had not run. If successful, processing continues with the next step.

Current Plan turnover process (5/7)

The new current plan is taken over

15. The current plan is locked, preventing it from being updated.
16. The NCP is copied to the active current plan and the NCX is copied to the current plan extension (CX).
17. The job-tracking archive log is emptied.
18. The active job-tracking log now contains a record of the updates to the current plan that were made while the daily plan job was running. These are read, and the current plan is updated accordingly.
19. In-storage control blocks representing workstations and active operations are rebuilt from the active current plan, and a data space is created from the current-plan-extension data set.

Current Plan turnover process (6/7)

The newly created active current plan is backed up

20. A current plan backup is performed.
21. The current plan is unlocked and normal processing continues.
22. The data from the now inactive job-tracking log is copied into the JT archive log EQQJTARC).
23. A current plan backup is performed. The inactive current plan is erased.
24. The active current plan is copied to the inactive one, and EQQSCPDS is produced. This VSAM is a copy of the current plan and will be used by data processing to add job information to the Symphony file.
25. The next available job-tracking log becomes active.
26. The current plan is unlocked and normal processing continues. All changes to jobs in the Symphony file are sent to the distributed agents, when the Symphony file is available.

Current Plan turnover process (7/7)

The Symphony File is updated

27. Starting from EQQSCPDS that contains the updates for the job-tracking file, the job and job stream information is added to the new Symphony file. If a problem occurred during the building of the Symphony file, the Symphony file is not built. To create the Symphony file, you must perform a Symphony renew after correcting the errors.

The Symphony File is distributed

28. IWS z/OS sends the Symphony file to IWS.

29. Before distributing the new Symphony file, all the events of the old Symphony file that are still outstanding are applied to the new Symphony file.

30. In IWS, the new Symphony file is distributed.

Scenarios :

- A Reset Plan has occurs (9.5)

Odlb2155318
Global Knowledge

Recover the current plan

- Cancel the Controller
- Modify Currplan(current) by Currplan(New) into the parmlib
- Start the controller
- The controller read the JTARC+JTxx and rebuild the planning
- At the end of the process the planning is the same.

Odb2155318
Global Knowledge

IWS SECURITY

Odb2155318
Global Knowledge

Objective

In this chapter , you will learn about the architecture of IWS product

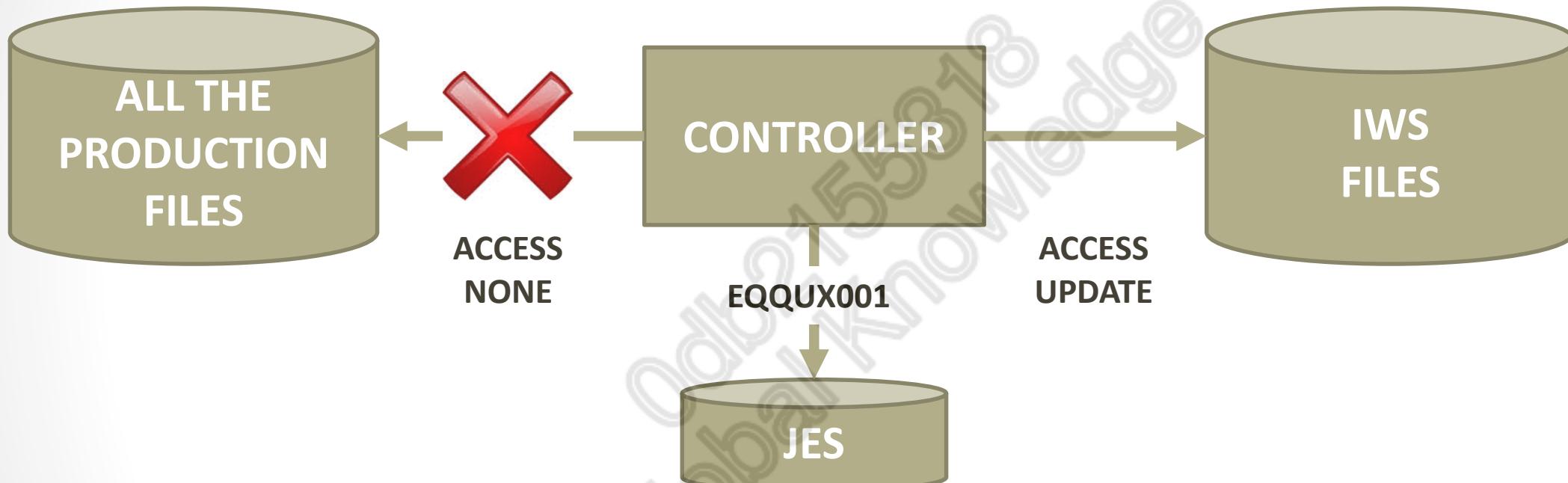
After completing this training, you should be able to:

Setup security in IWS

Manage security in RACF

Manage security on LDAP

Principle

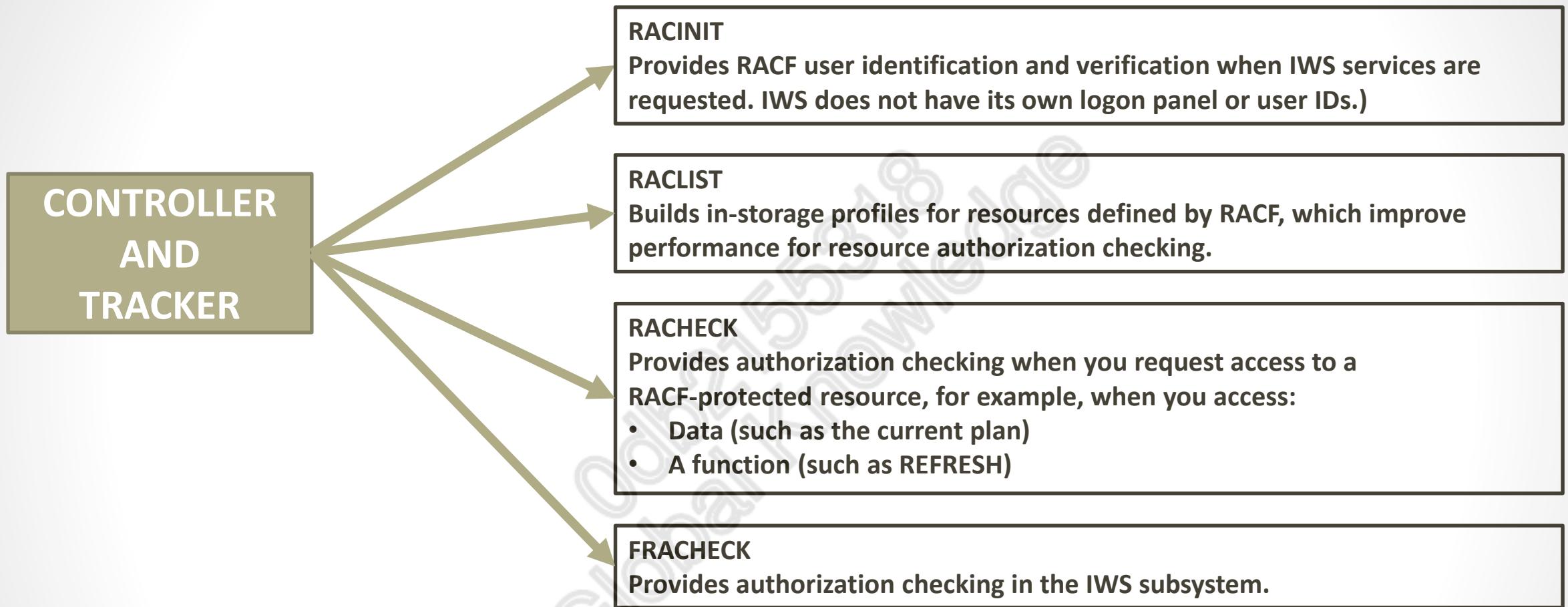


- IWS can submit jobs
- IWS can update its own file
- IWS has no right into the rest of your production

RACF

Odb2155318
Global Knowledge

RACF



Identifying Users

IWS needs access to non IWS resources for the work it schedules. The user ID associated with IWS can be obtained from:

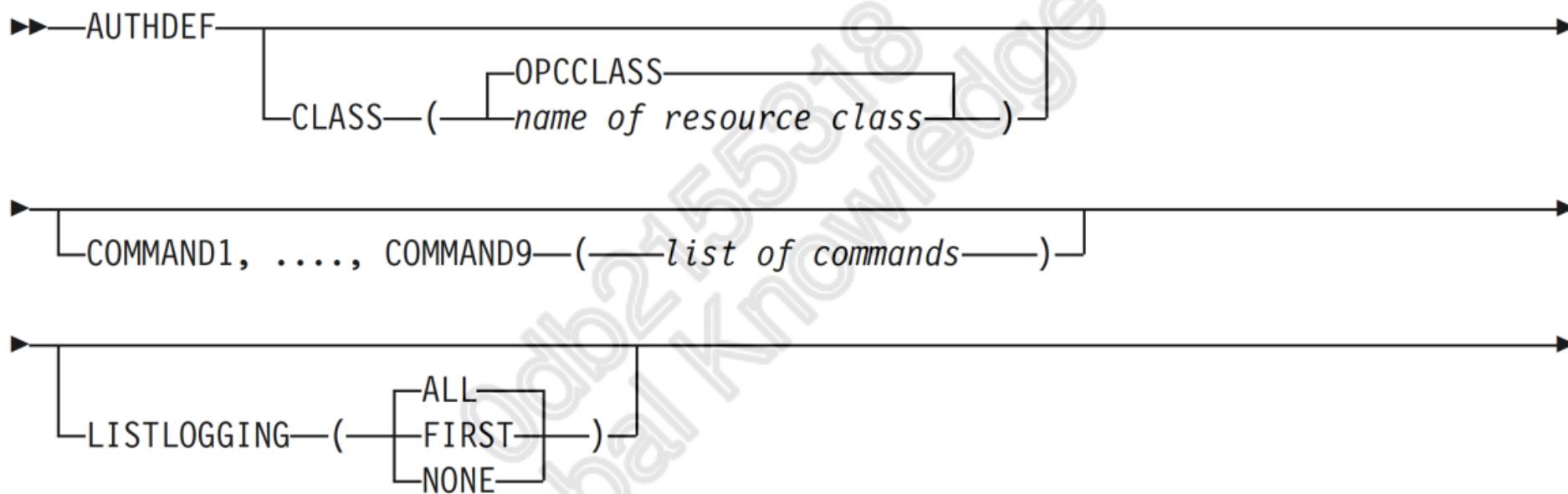
- The IWS address space that accesses data sets used by the work it schedules, and that submits work and issues JES and MVS commands.
- The user= parameter on the JOB card of a batch job.
- The IBM Workload Scheduler for z/OS job-submit exit, EQQUX001, which is called when IWS is about to submit a job or start a STC, and which can pass back a user ID.
- The USRREC statement, which specifies the name and password of the user on a supported Windows workstation.
- The LOCALPSW statement, which specifies whether the name and password of a user on a Windows workstation is defined either on z/OS using the USRREC statement (LOCALPSW set to NO) or on the Windows workstation using a local file (LOCALPSW set to YES). If you set LOCALPSW to YES, the scheduler looks for the USRREC statement first, then for the local file.
- The USERMAP keyword of the SERVOPTS statement.

Who are the userid

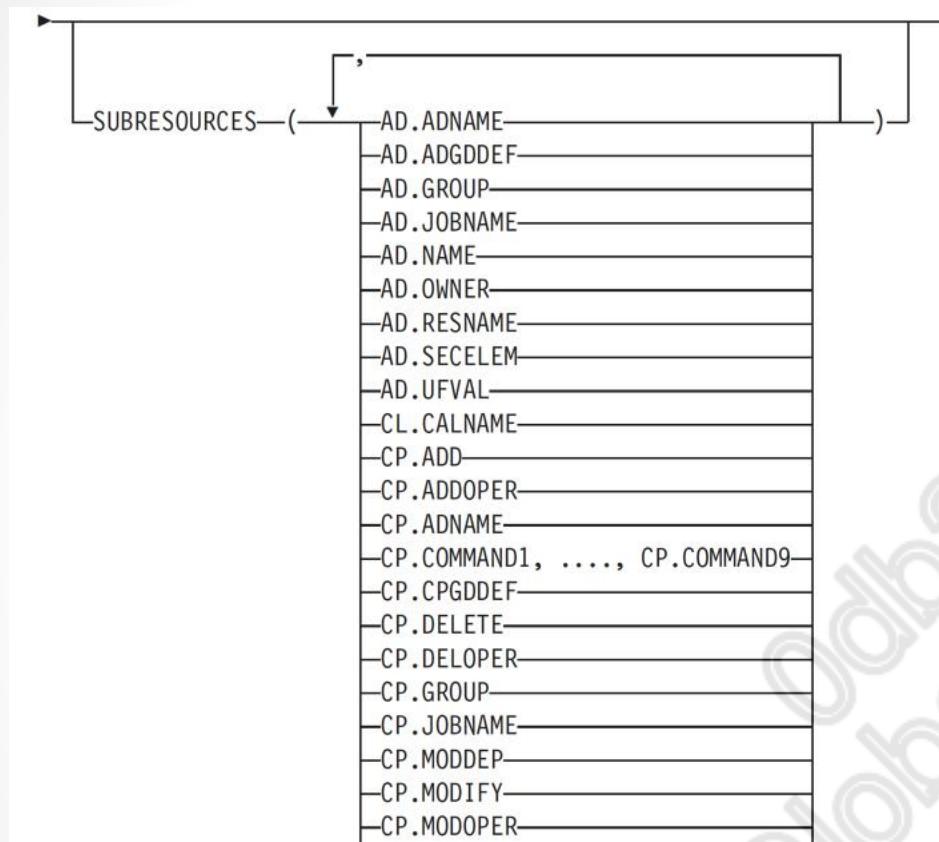
User IDs that access IBM Workload Scheduler for z/OS resources can be:

- A TSO user ID that accesses the IWS Dialogs , submits batch jobs that access IWS resources, and issues IWS TSO commands.
- An IWS address space, which must be permitted access to IWS resources.
- Other STC that pass requests to IWS address space.
- A user ID supplied by a transaction program (TP) that uses the IWS API to communicate with the controller.
- A user ID defined by the USERMAP keyword of the SERVOPTS statement to work with Dynamic Workload Console.

IWS PARMLIB : AUTHDEF



IWS PARMLIB



CP.MODOPSTAT
CP.NAME
CP.OWNER
CP.SECELEM
CP.UFVAL
CP.WSNAME
CP.ZWSOPER
ET.ADNAME
ET.ETNAME
JL.DSNAME
JL.MEMBER
JS.ADNAME
JS.GROUP
JS.JOBNAME
JS.OWNER
JS.WSNAME
JV.OWNER
JV.TABNAME
LT.ADNAME
LT.LTGDEF
LT.OWNER
OI.ADNAME
PR.PERNAME
RD.RDNAME
RG.RGNAME

RG.OWNER
RL.ADNAME
RL.GROUP
RL.OWNER
RL.WSNAME
RL.WSSTAT
SR.SRNAME
WS.WSNAME

COMMANDn

COMMAND	DESCRIPTION
C	Complete an occurrence
CG	Complete Group
DG	Delete Group
R	Rerun
RG	Remove from group
W	Set Waiting
ARC	Attempt Automatic Recovery
BND	Bind Operation
DJ	Delete JCL
EX	Execute

COMMAND	DESCRIPTION
J	Edit JCL
JR	JR, Fast Path JR
K	Kill
MH	Manual Hold
MR	Manual Release
NP	NOP
RI	Recovery Info (PY and PN)
SC	SC, Fast Path SC
SJR	Simple Job Restart
SR	SR, Fast Path SR
UN	UN NOP

COMMANDn SAMPLE

```
AUTHDEF CLASSNAME(OPCCLASS
    LISTLOGGING(ALL)
    TRACE(0)
    SUBRESOURCES(AD.ADNAME
        CP.COMMAND1...
    )
COMMAND1(C,CG,DG,R,RG,W)
```

IWS EQQUX001

SET_WSCHEV: SET WSCHEV ACCORDING TO SPECIAL RESOURCE NAME

SET_RUSE1: SET THE RACF USER ACCORDING TO JOBNAM CHECK

SET_RUSE2: SET THE RACF USER WITH THE AUTHORITY GROUP.

SET_RUSE3 : SET THE RACF USER WITH THE JOB CARD USER= VALUE

SET_RUSE4 : SET THE RACF USER WITH THE OWNER

SUR_CHECK: PERFORM RACF SURROGATE CHECK WITH SUBMIT USER VALUE

Protected fixed resources and sub resources

Fixed Resource	Subresource	RACF resource name	Description
AD	AD.ADNAME	AD	Application-description file
	AD.ADGDDEF	ADA.name	Application name
	AD.NAME	ADD.name	Group-definition-ID name
	AD.OWNER	ADN.name	Operation extended name in application
	AD.GROUP	ADO.name	Owner ID
	AD.JOBNAME	ADG.name	Authority group ID
	AD.RESNAME	ADJ.name	Operation job name in application
	AD.SECELEM	ADR.resname	Special resource name
	AD.UFVAL	ADM.name	Security element name
		ADU.field_name.field_value	User field name and value.
ADEP		ADEP	Selecting all dependencies in the QCP dialog

Protected fixed resources and sub resources

Fixed Resource	Subresource	RACF resource name	Description
CL	CL.CALNAME	CL CLC.name	Calendar data Calendar name
CP	CP.ADD	CP	Current-plan file
	CP.DELETE	CP.ADD	Add workload
	CP.MODIFY	CP.DELETE	Delete workload
	CP.ADDOPER	CP.MODIFY	Modify workload
	CP.DELOPER	CP.ADDOPER	Add operation
	CP.MODOPER	CP.DELOPER	Delete operation
	CP.MODDEP	CP.MODOPER	Modify operation
	CP.MODOPSTAT	CP.MODDEP	Modify dependencies
	CP.COMMANDn	CP.MODOPSTAT	Modify operation status
	CP.ADNAME	CP.COMMANDn	List of commands
	CP.CPGDDEF	CPA.name	Occurrence name
	CP.NAME	CPD.name	Occurrence group-definition-ID
	CP.OWNER	CPN.name	Operation extended name
	CP.GROUP	CPO.name	Occurrence owner ID
	CP.JOBNAME	CPG.name	Occurrence authority-group ID
	CP.WSNAME	CPJ.name	Occurrence operation name
	CP.ZWSOPER	CPW.name	Current plan workstation name
	CP.SECELEM	CPZ.name	Workstation name used by an operation
	CP.UFVAL	CPM.name	Security element name
	CP.RESNAME	CPU.field_name.field_value	Operation user field name and value
		CPR.resname	Special resource name

Protected fixed resources and sub resources

Fixed Resource	Subresource	RACF resource name	Description
ETT	ET.ETNAME ET.ADNAME	ETT ETE.name ETA.name	ETT dialog Name of triggering event Name of application to be added
JL	JL.DSNAME JL.MEMBER	JL JLD.name JLM.name	Job library data sets Job library data set name JCL member name
JS	JS.ADNAME JS.OWNER JS.GROUP JS.JOBNAME JS.WSNAME	JS JSA.name JSO.name JSG.name JSJ.name JSW.name	JCL and job-library file Occurrence name Occurrence owner ID Occurrence authority group ID Occurrence operation name Current plan workstation name

Protected fixed resources and sub resources

Fixed Resource	Subresource	RACF resource name	Description
JV	JV.OWNER JV.TABNAME	JV JVO.name JVT.name	JCL variable-definition file Owner ID of JCL-variable-definition table Name of JCL-variable table
LT	LT.ADNAME LT.LTGDEF LT.OWNER	LT LTA.name LTD.name LTO.name	Long-term-plan file Occurrence name Occurrence group-definition ID Occurrence owner ID
OI	OI.ADNAME	OI OIA.name	Operator-instruction file Application name
PR	PR.PERNAME	PR PRP.name	Period data Period name
RD	RD.RDNAME	RD RDR.name	Special resources file Special resource name

Protected fixed resources and sub resources

Fixed Resource	Subresource	RACF resource name	Description
RG	RG.RGNAME RG.OWNER	RG RGY.name RGO.name	Run cycle group Run cycle group name Run cycle group owner
RL	RL.ADNAME RL.OWNER RL.GROUP RL.WSNAME RL.WSSTAT	RL RLA.name RLO.name RLG.name RLW.name RLX.name	Ready list data Occurrence name Occurrence owner ID Occurrence authority-group ID Current-plan workstation name Current-plan workstation changed by WSSTAT

Protected fixed resources and sub resources

Fixed Resource	Subresource	RACF resource name	Description
RP	RP.REPTYPE	RP RPT.reptype	<p>Dynamic Workload Console reports</p> <p>Report type depending on the report you request:</p> <ul style="list-style-type: none">• RUNHIST For job run history reports.• RUNSTATS For job run statistics.• WWR For workstation workload runtimes reports.• WWS For workstation workload summary.• SQL For reports obtained by customized SQL queries
SR	SR.SRNAME	SR SRS.name	<p>Special resources in the current plan</p> <p>Special resource name</p>
WS	WS.WSNAME	WS WSW.name	<p>Workstation data</p> <p>Workstation name in workstation database</p>

Protected fixed resources and sub resources

Fixed Resource	Subresource	RACF resource name	Description
ARC		ARC	Activate/deactivate automatic recovery
BKP		BKP	Request backup of a resource data set
BUL		BUL	Initiate bulk discovery for the monitoring agent
CMAC		CMAC	Data set and Catalog Cleanup used by the Restart and Cleanup function.
CONT		CONT	Refresh RACF subresources
ETAC		ETAC	Activate/deactivate event-triggered tracking
EXEC		EXEC	EX (execute) row command
JSUB		JSUB	Activate/deactivate job submit
REFR		REFR	Refresh LTP and delete CP
WSCL		WSCL	All-workstations-closed data

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
Workstation	Browse workstation	WS	Read
	Update Workstation	WS WSCL	Update Read
	Browse workstation close	WSCL	Read
	Update workstation closed	WSCL	Update
	Print	None	None
Calendar	Browse	CL	Read
	Update	CL	Update
	Print	None	None
Period	Browse	PR	Read
	Update	PR	Update
	Print	None	None

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
Application	Browse	AD CL WS OI RD	Read Read Read Read Read
	Update	AD CL PR WS OI JV RD	Update Read Read Read Update Read Read
	PRINT	WS	Read
	Mass Update	AD CL PR WS JV RD	Update Read Read Read Read Read

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
Operator Instructions	Browse	OI	Read
	Update	OI	Update
	Print	None	None
	Mass Update	None	None
Special Resource	Browse	RD WS	Read Read
	Update	RD WS	Update Read

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
JCL Variable Tables	Browse	JV	Read
	Update	JV	Update
	Print	JV	Read
JCL in job library	Browse	JL	Read
	Update	JL	Update

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
LTP	Browse	LT AD CL PR WS	Read Read Read Read Read
	Update (delete or modify or add)	LT AD CL PR WS JV	Update Read Read Read Read Read

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
LTP	Job Setup	LT	Read
		AD	Read
		CL	Read
		PR	Read
		WS	Read
		JS	Update
	Batch	LT	Read
	Display Status	LT	Read
	Set defaults	None	None
Daily Planning	Batch	CP	Read

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
Work Station Communication	Using ready lists	RL CP JS OI JV EXEC	Update Read Update Read Read Update
	Waiting List	CP JS OI	Read Update Read
	Job setup	CP JS OI	Read Update Read
	Review workstation status	CP	Read
	Define Ready List	None	None

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
Modify Current Plan (MCP)	Add	AD CP JS JV SR	Read Update Read Read Update
	Update (delete or modify), change status of workstations	CP JS JV SR	Update Update Read Update
	Change status, rerun, error handling	CP JS OI EXEC	Update Update Read Update

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
Modify Current Plan (MCP)	Restart and Cleanup	CP JS CMAC	Update Update Update
	Browse	CP JS OI SR	Read Read Read Read
	Job Setup	CP JS	Read Update
	Delete Error Lists	None	None
	History Data Request	HIST	read

Access requirements to fixed resources for dialog users

Dialog	Function	Fixed Resources	Access Type
Query Current Plan (QCP)	All	CP JS OI SR	Read Read Read Read
Service Functions	Activate/deactivate job submission	JSUB CP	Update Update
	Activate/deactivate automatic recovery	ARC CP	Update Update
	Refresh (delete current plan and reset long-term plan)	REFR LT	Update Update
	Activate RACF resources	CONT	Update
	Activate/deactivate ETT	ETAC	Update
	Produce APAR Tape	None	None

User

- System → SYST
- IWS administrator → IADM
- IWS Operator → IOPR
- IWS Developer → IDEV
- Development team → DEVL
-

Protected fixed resources and sub resources

Fix Resources	IADM	IOPR	IDEV	SYST	DEVL	Description
AD	U	R	U	U	R	Application-description file
ADEP	U	N	N	U	N	Selecting all dependencies in the QCP dialog
CL	U	R	R	U	R	Calendar
CP	U	U	R	U	R	Current Plan
ETT	U	R	R	U	R	Event triggered Tracking
HIST	U	R	R	U	R	Retrieving history data with HIST command
JL	U	R	U	U	R	Job library data sets
JS	U	R	R	U	R	JCL and job-library file
JV	U	R	U	U	R	JCL variable-definition file
LT	U	U/R	R	R	R	Long-term-plan file

Protected fixed resources and sub resources

Fix Resources	IADM	IOPR	IDEV	SYST	DEVL	Description
OI	U	R	U	U	R	Operator Instruction
PR	U	R	R	U	R	Period
RD	U	R	R	U	R	Special Resource definition
RG	U	R	R	U	R	Run Cycle Group
RL	U	R	R	U	R	Ready List
RP	U	R	R	U	R	Dynamic Workload console report
SR	U	R	R	U	R	Special Resource in the current plan
WS	U	R	R	U	R	Workstation

Protected fixed resources and sub resources

Fix Resources	IADM	IOPR	IDEV	SYST	DEVL	Description
ARC	U	N	N	N	N	Activate/deactivate automatic recovery
BKP	U	N	N	N	N	Request backup of a resource data set
BUL	U	N	N	N	N	Initiate bulk discovery for the monitoring agent
CMAC	U	N	N	N	N	Data set and Catalog Cleanup used by the Restart and Cleanup function.
CONT	U	N	N	U	N	Refresh RACF subresources
ETAC	U	U	N	N	N	Activate/deactivate event-triggered tracking
EXEC	U	N	N	N	N	EX (execute) row command
JSUB	U	U	N	U	N	Activate/deactivate job submit
REFR	U	N	N	U	N	Refresh LTP and delete CP
WSCL	U	U	N	N	N	All-workstations-closed data

LDAP

Odb2155318
Global Knowledge

LDAP Configuration

You must define LDAP configuration in DWC

The supported LDAP are

- IBM Tivoli Directory Server
- z/OS Integrated Security services LDAP server
- IBM Lotus Domino
- Novell Directory Services
- Sun Java System Directory Server
- Microsoft Windows Active Directory
- Microsoft Active Directory Application Mode
- Custom (Open LDAP)

WebSphere definition interface

<https://iws940:16316/ibm/console/>

Global security > Federated repositories > Manage repositories > CN=Users,DC=SETD,DC=local

Specifies the configuration for secure access to a Lightweight Directory Access Protocol (LDAP) repository with optional failover servers.

General Properties

Repository identifier: CN=Users,DC=SETD,DC=local

Repository adapter class name: com.ibm.ws.wim.adapter.ldap.LdapAdapter

LDAP server

Directory type: Microsoft Windows Active Directory

Primary host name: 192.168.117.148 Port: 389

Failover server used when primary is not available:

Support referrals to other LDAP servers: Ignore

Support for repository change tracking: none

Custom properties:

Select	Name	Value
<input type="checkbox"/>		

Security

Bind distinguished name: CN=setdadmin,CN=Users,DC=SETD,DC=local

Bind password: *****

Federated repository properties for login:

uid

LDAP attribute for Kerberos principal name: userprincipalname

Certificate mapping: EXACT_DN

Certificate filter:

Require SSL communications

Centrally managed

- [Manage endpoint security configurations](#)

Use specific SSL alias

NodeDefaultSSLSettings

- [SSL configurations](#)

Global knowledge

WebSphere definition interface

<https://iws940:16316/ibm/console/>

Global security

Global security > Federated repositories > Repository reference

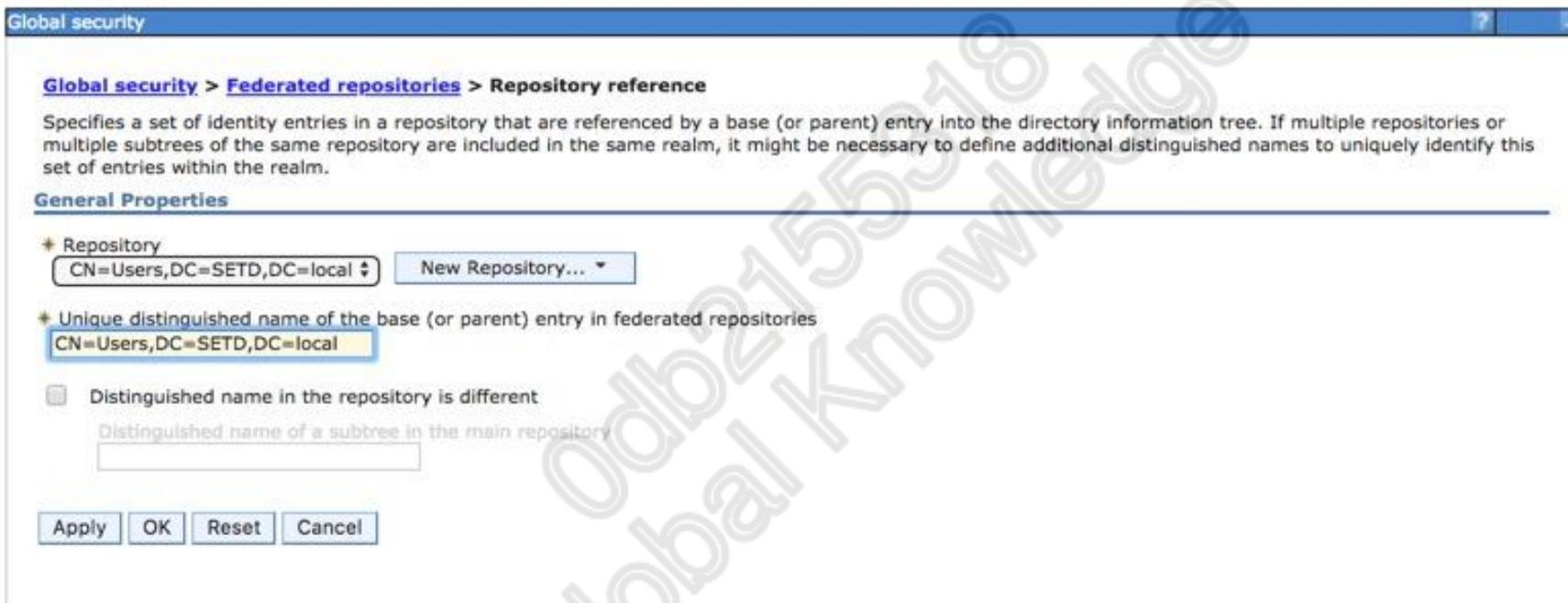
Specifies a set of identity entries in a repository that are referenced by a base (or parent) entry into the directory information tree. If multiple repositories or multiple subtrees of the same repository are included in the same realm, it might be necessary to define additional distinguished names to uniquely identify this set of entries within the realm.

General Properties

* Repository
CN=Users,DC=SETD,DC=local

* Unique distinguished name of the base (or parent) entry in federated repositories
CN=Users,DC=SETD,DC=local

Distinguished name in the repository is different
Distinguished name of a subtree in the main repository



WebSphere definition interface

<https://iws940:16316/ibm/console/>

[Global security > Federated repositories](#)

By federating repositories, identities stored in multiple repositories can be managed in a single, virtual realm. The realm can consist of identities in the file-based repository that is built into the system, in one or more external repositories, or in both the built-in repository and one or more external repositories.

General Properties

- Realm name: defaultWIMFileBasedRealm
- Primary administrative user name: iws940

Server user identity

- Automatically generated server identity
- Server identity that is stored in the repository
 - Server user ID or administrative user on a Version 6.0.x node: iws940
 - Password: *****

Ignore case for authorization

Allow operations if some of the repositories are down

Repositories in the realm:

Add repositories (LDAP, custom, etc...)	Use built-in repository	Remove	
Select	Base Entry	Repository Identifier	Repository Type
You can administer the following resources:			
<input type="checkbox"/>	CN=Users,DC=SETD,DC=local	CN=Users,DC=SETD,DC=local	LDAP:AD
<input type="checkbox"/>	o=defaultWIMFileBasedRealm	InternalFileRepository	File
Total 2			

WebSphere definition interface

After the stop and start of the WAS you can see group and user

The screenshot shows the 'Manage Users' interface in the WebSphere Management Console. The left sidebar has a 'View: All tasks' dropdown and a list of categories: Welcome, Guided Activities, Servers, Applications, Services, Resources, Security, Environment, System administration, Users and Groups (selected), Monitoring and Tuning, Troubleshooting, Service integration, and UDDI. Under 'Users and Groups', there are sub-options: Administrative user roles, Administrative group roles, Manage Users, and Manage Groups. The main panel title is 'Manage Users' with a 'Search for Users' section containing fields for 'Search by' (User ID), 'Search for' (*), and 'Maximum results' (100). A 'Search' button is present. Below this, a message says '9 users matched the search criteria.' A table lists the users:

Select	User ID	First name	Last name	E-mail	Unique Name
<input type="checkbox"/>	AGTB	AGTB			CN=AGTB,CN=Users,DC=SETD,DC=local
<input type="checkbox"/>	Administrateur	Administrateur			CN=Administrateur,CN=Users,DC=SETD,DC=local
<input type="checkbox"/>	Invité	Invité			CN=Invité,CN=Users,DC=SETD,DC=local
<input type="checkbox"/>	TWSADMIN1	TWSADMIN1	TWSADMIN1		CN=TWSADMIN1 TWSADMIN1,CN=Users,DC=SETD,DC=local
<input type="checkbox"/>	TWSPILOTE1	TWSPILOTE1	TWSPILOTE1		CN=TWSPILOTE1 TWSPILOTE1,CN=Users,DC=SETD,DC=local
<input type="checkbox"/>	iws940	iws940		iws940	uid=iws940,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	krbtgt	krbtgt			CN=krbtgt,CN=Users,DC=SETD,DC=local
<input type="checkbox"/>	oracle	oracle			CN=oracle,CN=Users,DC=SETD,DC=local
<input type="checkbox"/>	setdadmin	setdadmin			CN=setdadmin,CN=Users,DC=SETD,DC=local

At the bottom, it says 'Page 1 of 1' and 'Total: 9'.

DWC PARAMETER on z/OS

SERVOPTS

SUBSYS(T93C)

PROTOCOL(TCP)

USERMAP(T93WUSER)

JSCHOSTNAME(ZOS220)

PORTNUMBER(5930)

CODEPAGE(IBM-1147)

INIT

CALENDAR(DEFAULT)

MEMBER T93WUSER

USER 'IWS940à*' RACFUSER(TWS930)

USER 'USER01à*' RACFUSER(EXPL30)

IF NO USERMAP THEN TMEADMIN
CLASS IS USED

IWS Migration of version

Odb2155318
Global Knowledge

Objective

In this chapter , you will learn about the architecture of IWS product

After completing this training, you should be able to:

- Migrate from one version of IWS to a new version of IWS
- Back out to the previous version in case of problem

Methodology

- Auditing actual IWS
- Setup the new JOBLIB
- Use the alias dataset
- Verify the File (SIZE, recordsize, DCB)
- Install STEPLIB tools for TSO/E
- Installing toleration PTF
- Generate all TWS job using the ISPF Panels
- Create Parameter member for all the IWS utilities
- Modify the STC and system PARAMETER for the new release
- Create all Migration JOB
- Prepare and test the backout on test environment
- **All the migration process must be prepared into a PDS in sequential order**

Auditing

- Check all JCL which have EQQ somewhere
- Check all program, rexx, clist which call IWS
- Check the actual IWS parmlib
- Verify the connection between the new IWS and all the other IWS
- Check the size and recordsize of all VSAM dataset.

Odb218318
Global Knowledge

setup the new Joblib

- Check all JCL which have EQQ somewhere
- Copy all this JCL into a new JOBLIB and modify the future controller PROC
`//EQQJBLIB DD DISP=EXP.JOBLIB.V930.CNTL`
`// DD DISP=EXP.JOBLIB.CNTL`
- Create all the IWS JCL from the ISPF Dialog
 - Plan extension
 -
 - Update the new joblib with the new version of JCL

Alias Dataset

Standard Dataset	Alias
TWS.V9R3M0.SEQQLMD0	TWS.PROD.SEQQLMD0
TWS.V9R3M0.SEQQMSG0	TWS.PROD.SEQQMSG0
TWS.V9R3M0.SEQQCLIB	TWS.PROD.SEQQCLIB
TWS.V9R3M0.SEQQPENU	TWS.PROD.SEQQPENU
TWS.V9R3M0.SEQQLENU	TWS.PROD.SEQQLENU
TWS.V9R3M0.SEQQTBL0	TWS.PROD.SEQQTBL0
TWS.V9R3M0.SEQQGNEU	TWS.PROD.SEQQGENU
TWS.V9R3M0.SEQQMISC	TWS.PROD.SEQQMISC
TWS.V9R3M0.SEQQWAPL	TWS.PROD.SEQQWAPL

Custom Dataset	Alias
CUST.V9R3M0.SEQQLMD0	CUST.PROD.SEQQLMD0
CUST.V9R3M0.SEQQMSG0	CUST.PROD.SEQQMSG0
CUST.V9R3M0.SEQQCLIB	CUST.PROD.SEQQCLIB
CUST.V9R3M0.SEQQPENU	CUST.PROD.SEQQPENU
CUST.V9R3M0.SEQQLENU	CUST.PROD.SEQQLENU
CUST.V9R3M0.SEQQTBL0	CUST.PROD.SEQQTBL0
CUST.V9R3M0.SEQQGNEU	CUST.PROD.SEQQGENU
CUST.V9R3M0.SEQQMISC	CUST.PROD.SEQQMISC
CUST.V9R3M0.SEQQWAPL	CUST.PROD.SEQQWAPL

VSAM, NON VSAM

- Check all VSAM file into the STC and the Job
 - DCB, Recordsize, Size
- Check all Non VSAM into the Job
 - DCB, Size
- Change the EQQJSDS recordszie
 - Default value is 180 000 then 3120 JCL lines into the EQQJSDS
 - The maximum value is 735480 then 9193 lines
- Change the EQQLTDS, EQQLLDDS recordszie
 - Default value is 131072 then maximum 3071 dependencies per Jobstream
 - Maximum value is 735480 then 18181 dependencies

STEPLIB

- CBTAPE
- File # 452 Dan Dalby's MVS-JES2 Utilities + STEPLIB, etc.

The steplib module must be in APF library

You need to add this line on the CLIST which start every version of IWS

```
PROC 0
CONTROL NOLIST NOFLUSH NOMSG NOCONLIST
STEPLIB ADD DATASETS('TWS930.SEQQLMD0') FIRST NOMSGS
```

Or if REXX

Address TSO

```
"STEPLIB ADD DATASET('TWS920.SEQQLMD0') FIRST NOMSGS"
```

You can have multiple version of IWS on the same z/OS and you can work with them from the same TSO/E session

Toleration PTF

Technote (troubleshooting)

JCLI error for every job if PI24927 not installed correctly

Problem(Abstract)

Normally the JES2 exits for TWSz (EXIT7 and EXIT51) are downwards compatible. However there is a restriction involving PI24927 which modified JES2 EXIT51

Resolving the problem

If while still running TWSz tracker or controller subsystem from earlier releases (8.5.0, 8.5.1, 8.6, 9.1, 9.2) the JES2 EXIT51 is assembled using the higher level code (for example IWSz 9.3 or an older release that has PI24927and all submitted jobs will receive JCLI error. To avoid this problem if the JES2 EXIT51 is assembled using the highest release of TWSz that is installed and this release contains the PI24927 PTF, then the PI24927 PTF must also be applied to the earlier release of TWSz.

The PTFs for PI24927 are:

- UI23864 8.5.0**
- UI23865 8.5.1**
- UI23866 8.6.0**
- UI23867 9.1.0**
- UI23868 9.2.0**

Generate All JCL form ISPF Panels

- LTP
 - 2.1 Modify ALL applications
 - 2.2 Modify one applications
 - 2.3 Extend LTP
 - 2.4 Print
 - 2.5 Print One
 - 2.7 Create
- CP
 - 3.1 Replan
 - 3.2 Extend
 - 3.3 Trial
 - 3.4 Print Current
 - 3.5 Symphony Renew
 - 3.6 Archive

Modify all your existing JCL with the new version

Create Parameter member for all the IWS utilities

For all IWS JCL add EQQYPARM

```
//TWS930A1 JOB NOTIFY=&SYSUID,REGION=0M  
//ISCD EXEC PGM=EQQYCAIN,  
// PARM='MSGOFF,,MISSCP',  
// REGION=0M  
//STEPLIB DD DISP=SHR,DSN=TWS930.SEQQLMD0  
//EQQYPARM DD DISP=SHR,DSN=TWS930.SEQQPARM(PRODINIT)  
//EQQMLIB DD DISP=SHR,DSN=TWS930.SEQQMSG0  
//.....
```

Create Parameter member for all the IWS utilities

Content of EQQYPARM

INIT

SUBSYS(PROD)

/* REMHOSTNAME(ZOS220) */ use if TCPIP connection

/* REMPORTNUMBER(5930) */ use if TCPIP connection

CWBASE(00)

HIGHDATE(711231)

STC

- **SEQQSAMP STC**
 - EQQCON → Controller and Tracker
 - EQQCONO → Controller
 - EQQTRA → Tracker
 - EQQDST → DataStore
 - EQQSER → End To end
 - EQQSER → TCPIP server
 - EQQOUC → Output Collector
- **SEQQSAMP PARAMETER**
 - EQQCONP → Controller and Tracker
 - EQQCONOP → Controller
 - EQQTRAP → Tracker
 - EQQDSTP → DataStore
 - EQQE2EP → End To end
 - EQQSERP → TCPIP server
 - EQQOUCP → Output Collector
 - EQQOUCH → Output Collector Header

Migration process 1/3

ORDER	DESCRIPTION
1	Deactivate job submission and pause the JES init
2	REPLAN
3	EQQYCAIN list occurrences into the plan
4	EQQYCAIN list operations into the plan
5	6.6 to check the active JS and CP if not JS1 then BACKUP RESDS JS SUBSYS(TRKT) if not CP1 then BACKUP RESDS CP SUBSYS(TRKT)
6	Stop all the IWS STC
7	Save all the TWS file using ADRDSSU
8	EQQPCS01-EQQPCS15
9	SAVE OLD STC
10	Allocate new SEQQCLIB, SEQQPENU, SEQQSKLO, SEQQTBL0,SEQQMSG0,LINKLIB,
11	Generate the JCL EDIT PANEL
12	Copy the customized Skeletons

Migration process 2/3

ORDER	DESCRIPTION
13	Copy the load customized program (Exit, eqqadd, ...)
14	Copy the customized Tables EQQACMDS → CNTLCMDS EQQAEDIT -> CNTLEDIT
15	DELDEF new alias 930
16	Convert all VSAM from previous version to new version CP1 → NCP CP1 → CP1 CP2→ CP2 JS1→ JS1 JS2→ JS2
17	Copy new STC into the PROCLIB Copy new PARM into the PARMLIB Copy JCLIB to new JCLIB
18	Check SUBMIT(NO) and CURRPLAN(NEW) into the Controller PARMLIB Check subsys level into the controller and tracker PARMLIB
19	Start the Tracker and check the event
20	Start all STC
21	EQQYCAIN list occurrences into the plan
22	EQQYCAIN list operations into the plan

Migration process 3/3

ORDER	DESCRIPTION
23	Compare the list occurrences and list operations and check the difference
24	Replan 930
25	If all OK modify Controller PARMLIB JOBSUMIT(YES) CURRPLAN(CURRENT)
26	Check utilities and ISPF
27	If all is OK activate job submission

Migration Backout

ORDER	DESCRIPTION
1	STOP the STC
2	SAVE ALL 930 FILE USING ADRDSSU
3	ALTER OLD VSAM AD,WS,SI,OI,RD REUSE CLEANUP ALL VSAM USING REPRO DUMMY AD,WS,NCP,CP1,CP2,LT,SI,OI,RD,JS1,JS2,CX,NCX ALTER OLD VSAM AD,WS,SI,OI,RD NOREUSE
4	VSAM conversion from 930 to old version
5	COPY the old STC
6	RENAME dataset alias to old version
7	MODIFY OLD Controller PARMLIB to define JOBSUBMIT(NO)
8	Start STC
9	REPLAN
10	Activate Job Submission

IWS TOOLS

Odb2155318
Global Knowledge

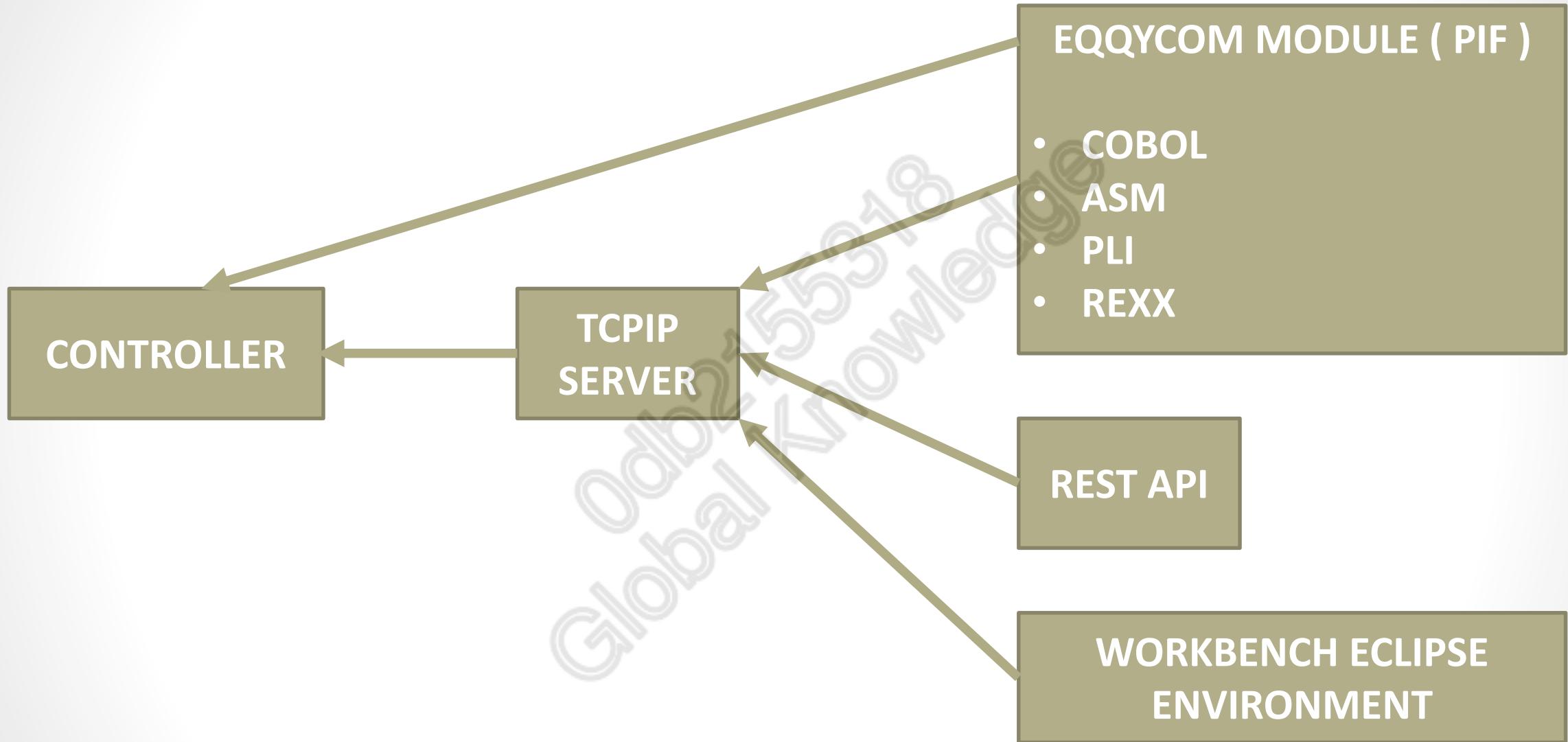
Objectives

In this chapter , you will learn about the tools, variables, JCL statement

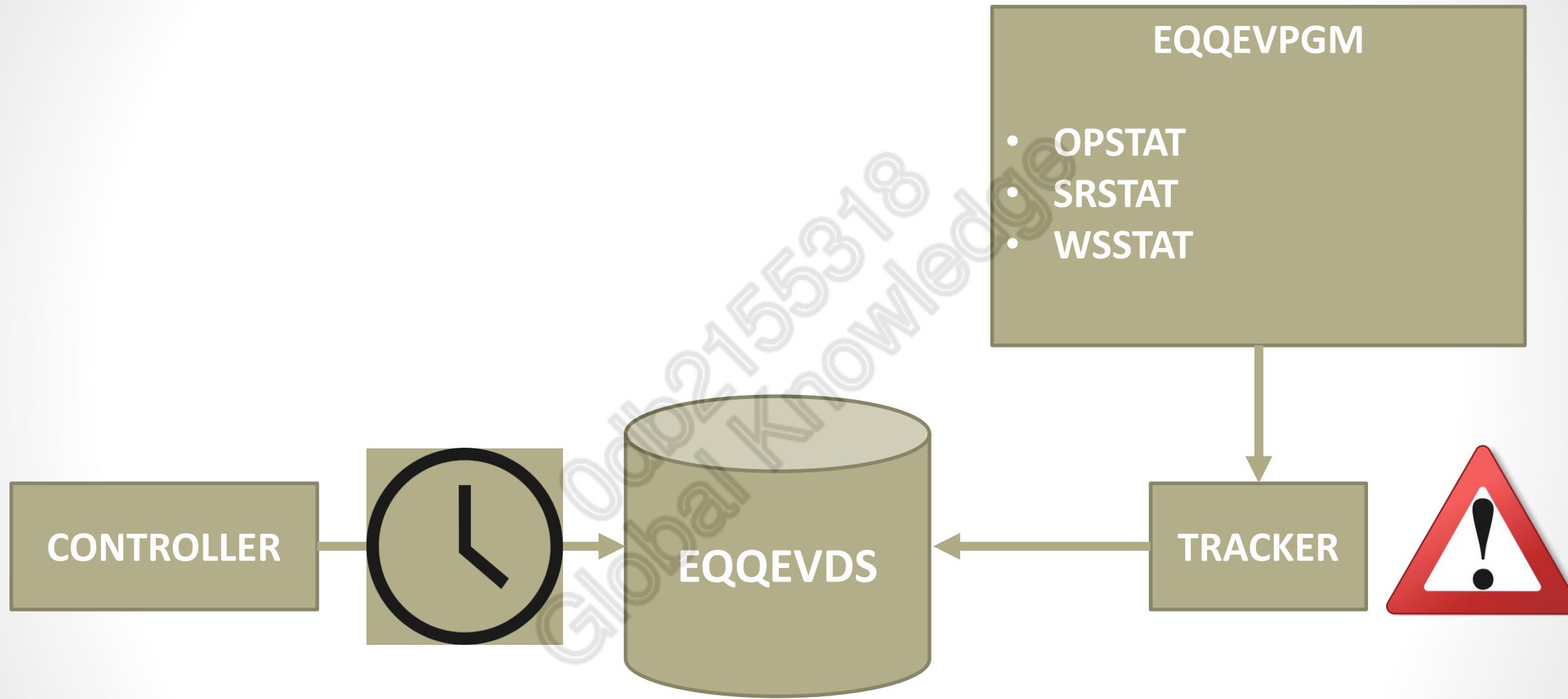
After completing this training, you should be able to work with:

- PIF
- EQQYCAIN (Batch Command Interface Tool)
- OCL
- EQQEVPGM
- VARIABLES
- OPC RECOVER
- JCL DIRECTIVE
- WAPL (replace SOE) manage into another PPT

Interface architecture



Interface architecture



EQQEVPGM

Odb2155318
Global Knowledge

EQQEVPGM function

- BACKUP Command to backup CP and JS
- BULKDISC Initial status of monitoring objects
- JSUACT Activate or Inactivate Job Submission Function
- OPINFO Command to set the user data of an operation in the current plan
- OPSTAT Set the status of an operation
- SRSTAT Change the availability, quantity, and deviation of a SR
- WSSTAT Change the status of a workstation in the CP

EQQEVPGM JCL

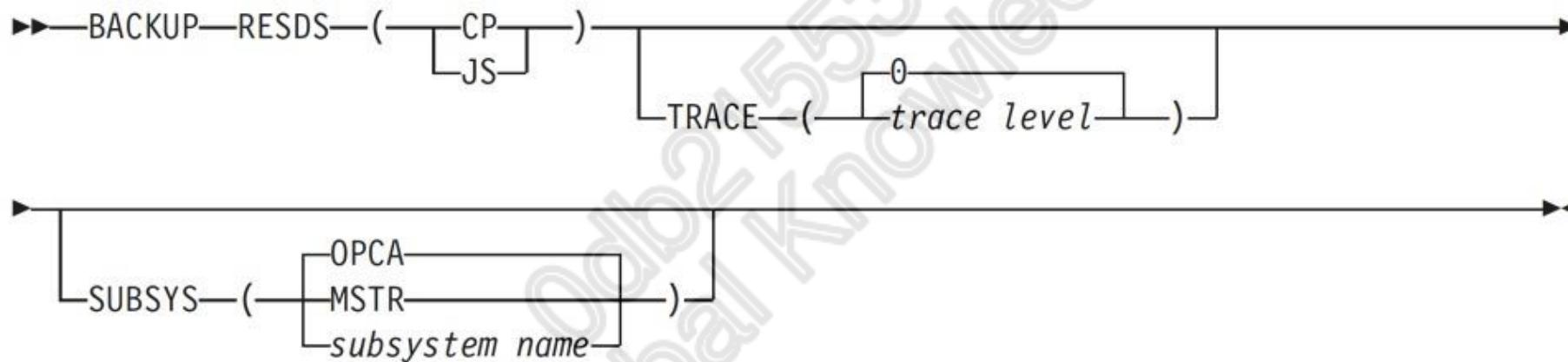
```
//EQQEVPGM EXEC PGM=EQQEVPGM  
//STEPLIB DD DISP=SHR,DSN=TWS930.SEQQLMD0  
//EQQMLIB DD DISP=SHR,DSN=TWS930.SEQQMSG0  
//EQQMLOG DD SYSOUT=*  
//SYSIN DD *  
COMMAND .... SUBSYS(OPCA/MSTR/T93T)
```

SUBSYS MSTR contact all the tracker active on this z/OS

SUBSYS T93T contact the tracker T93T

EQQEVPGM BACKUP

Syntax



EQQEVPGM BACKUP JS

BACKUP RESDS(JS) SUSBSYS(T93T)

RESULT ON CONTROLLER LOG

EQQN017I THE JCL REPOSITORY DATA SET WILL BE COPIED

EQQN015I A VSAM DATA SET WAS SUCCESSFULLY COPIED: FROMDD=EQQJS2DS, TODD=EQQJS1DS

EQQN016I DDNAME OF CURRENT JCL REPOSITORY DATA SET IS EQQJS1DS

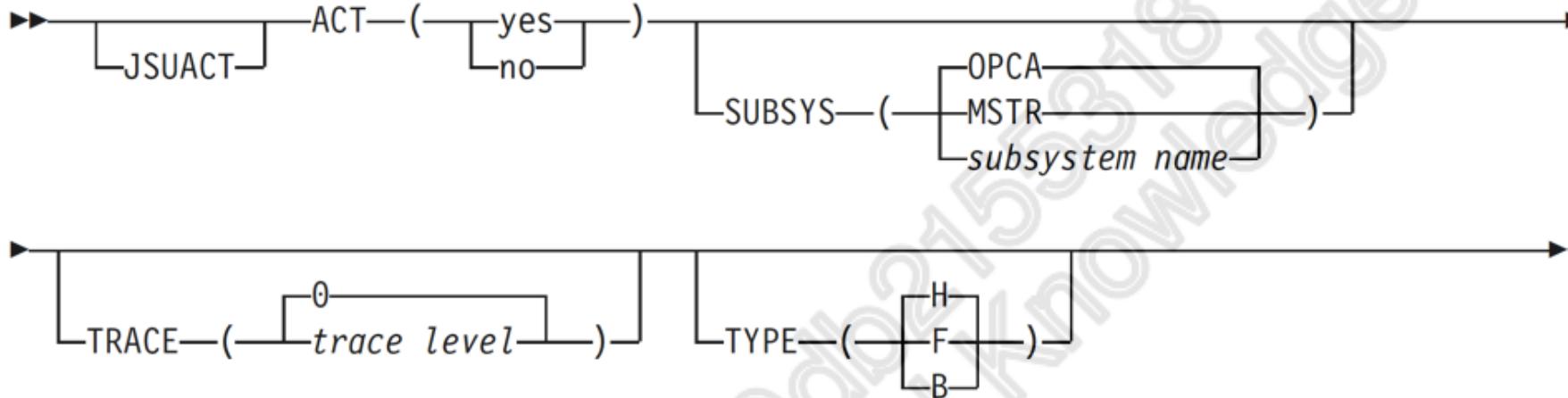
EQQEVPGM BACKUP CP

BACKUP RESDS(CP) SUSBSYS(T93T)

RESULT ON CONTROLLER LOG

EQQN051I A CURRENT PLAN BACKUP PROCESS HAS STARTED. TRIGGER WAS: BACKUP CMD
EQQN012I JOB TRACKING EVENTS ARE NOW BEING LOGGED ON FILE EQQJT01
EQQN223I EXTENDED AUDITING EVENTS ARE NOW BEING LOGGED ON FILE EQQDB01
EQQQ507I A SPECIAL RESOURCE DATASPACE BACKUP PROCESS HAS STARTED
EQQQ508I A SPECIAL RESOURCE DATASPACE BACKUP PROCESS HAS ENDED.EQQQ508I 00000000 RECORDS WERE WRITTEN TO CX
EQQN056I A CURRENT PLAN COPY PROCESS HAS STARTED
EQQN057I A CURRENT PLAN DATA SET WAS SUCCESSFULLY COPIED: FROMDD=EQQXD1DS, TODD=EQQXD1DS
EQQN023I VSAM LSR BUFFERS HAVE BEEN SUCCESSFULLY DELETED FOR VSAM FILE EQQXD1DS
EQQN018I VSAM LSR BUFFERS HAVE BEEN SUCCESSFULLY ALLOCATED FOR VSAM FILE EQQXD2D
EQQN018I NUMBER OF INDEX BUFFERS ARE 000006 WITH SIZE 001024
EQQN018I NUMBER OF DATA BUFFERS ARE 000005 WITH SIZE 032768
EQQN056I A CURRENT PLAN COPY PROCESS HAS STARTED
EQQN057I A CURRENT PLAN DATA SET WAS SUCCESSFULLY COPIED: FROMDD=EQQCP1DS, TODD=EQQCP1DS
EQQN023I VSAM LSR BUFFERS HAVE BEEN SUCCESSFULLY DELETED FOR VSAM FILE EQQCP1DS
EQQN018I VSAM LSR BUFFERS HAVE BEEN SUCCESSFULLY ALLOCATED FOR VSAM FILE EQQCP2DS

EQQEVPGM JSUACT



EQQEVPGM JSUACT

JSUACT ACT(YES/NO) SUBSYS(T93T) TYPE(H/F/B)

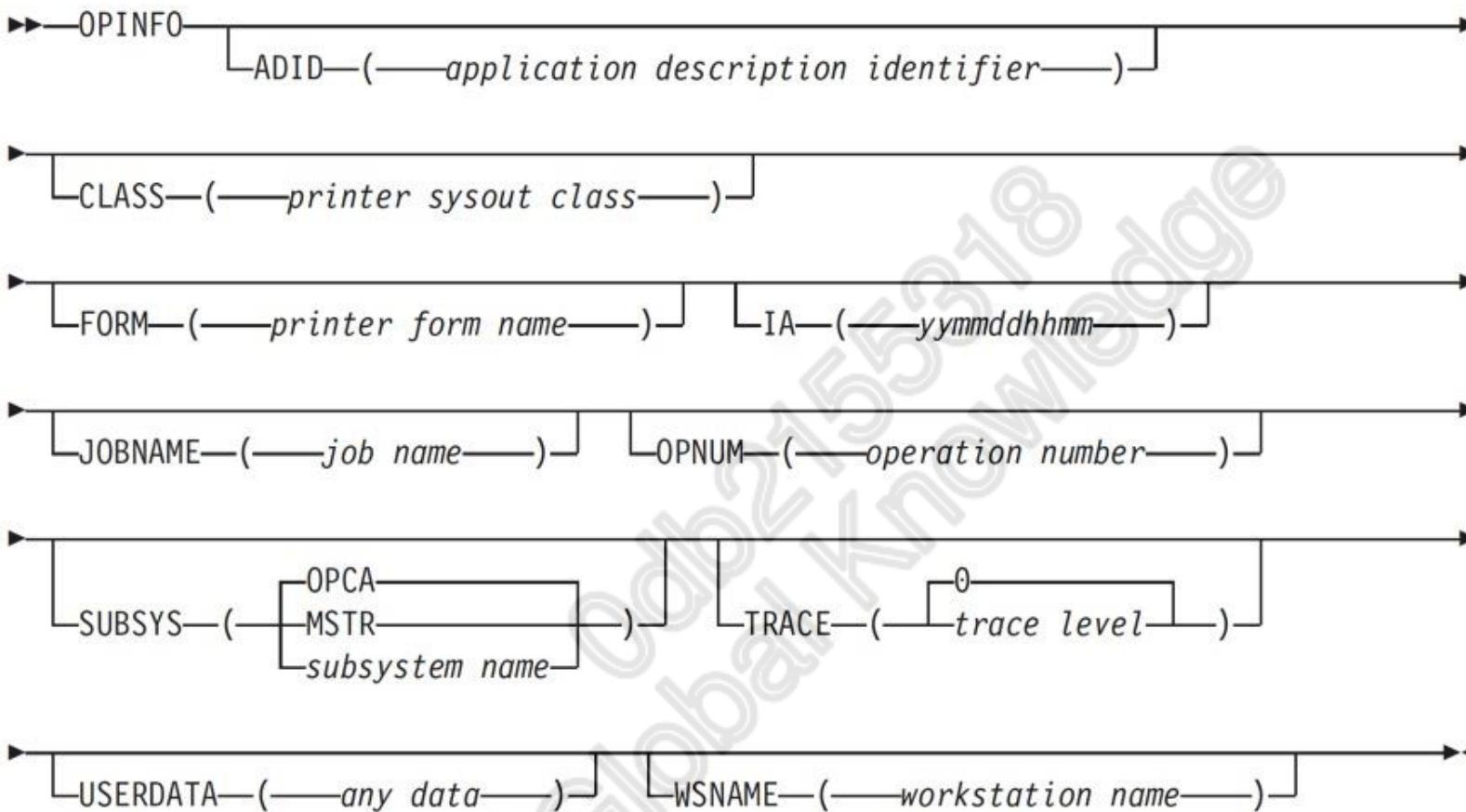
RESULT ON CONTROLLER LOG

EQQE204I SUBMIT FUNCTION DEACTIVATED FOR HOST

EQQE204I SUBMIT FUNCTION DEACTIVATED FOR Z-CENTRIC, DYNAMIC AND REMOTE ENGINE WORKSTATIONS

EQQE205E DEACTIVATE JOB SUBMISSION REQUEST REJECTED, E2E FEATURE NOT ACTIVE

EQQEVPGM OPINFO

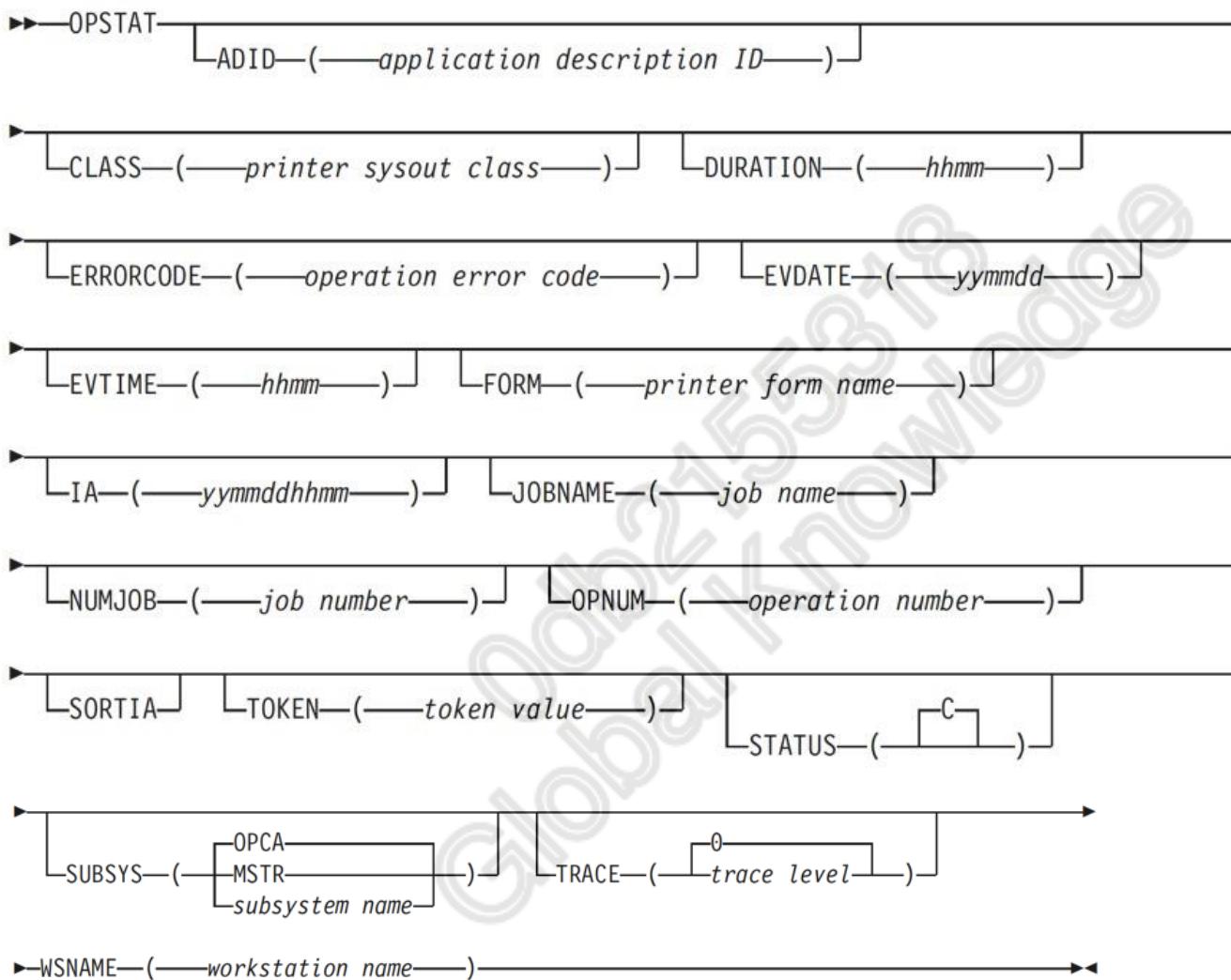


EQQEVPGM OPINFO

OPINFO W(TCP0) SUBSYS(T93T) J(TCP00000) A(TCP000) U(TRAININGIWS)

```
----- BROWSING DETAILED OPERATION INFORMATION -----
Command ==>
Application : TCP000
Operation   : TCP0 1
Occurrence token : D401A225A88F7000 Sdest: Xdest:
Jobname and Jobid : TCP00000
Reader date and time :
Status      : Arriving
on Work Station :
Job or Sysout class :
Form number :
Priority    : 5
Deadline WTO : No
COND Recovery Job : No
WLM Policy   :
Date and time for : Planned          Actual
Input arrival   : 18/03/10 10.00
Start          : 18/03/10 10.00
End            : 18/03/10 10.00
Deadline       : 18/03/10 11.00
Duration        : 00.00.01
Latest start    : 18/03/10 10.59
Resources      : Parallel servers R1   R2   Special resources
Required number :           1         0     0       1
                                                               User data:
                                                               TRAININGIWS
                                                               Applied run cycle
                                                               R1
```

EQQEVPGM OPSTAT



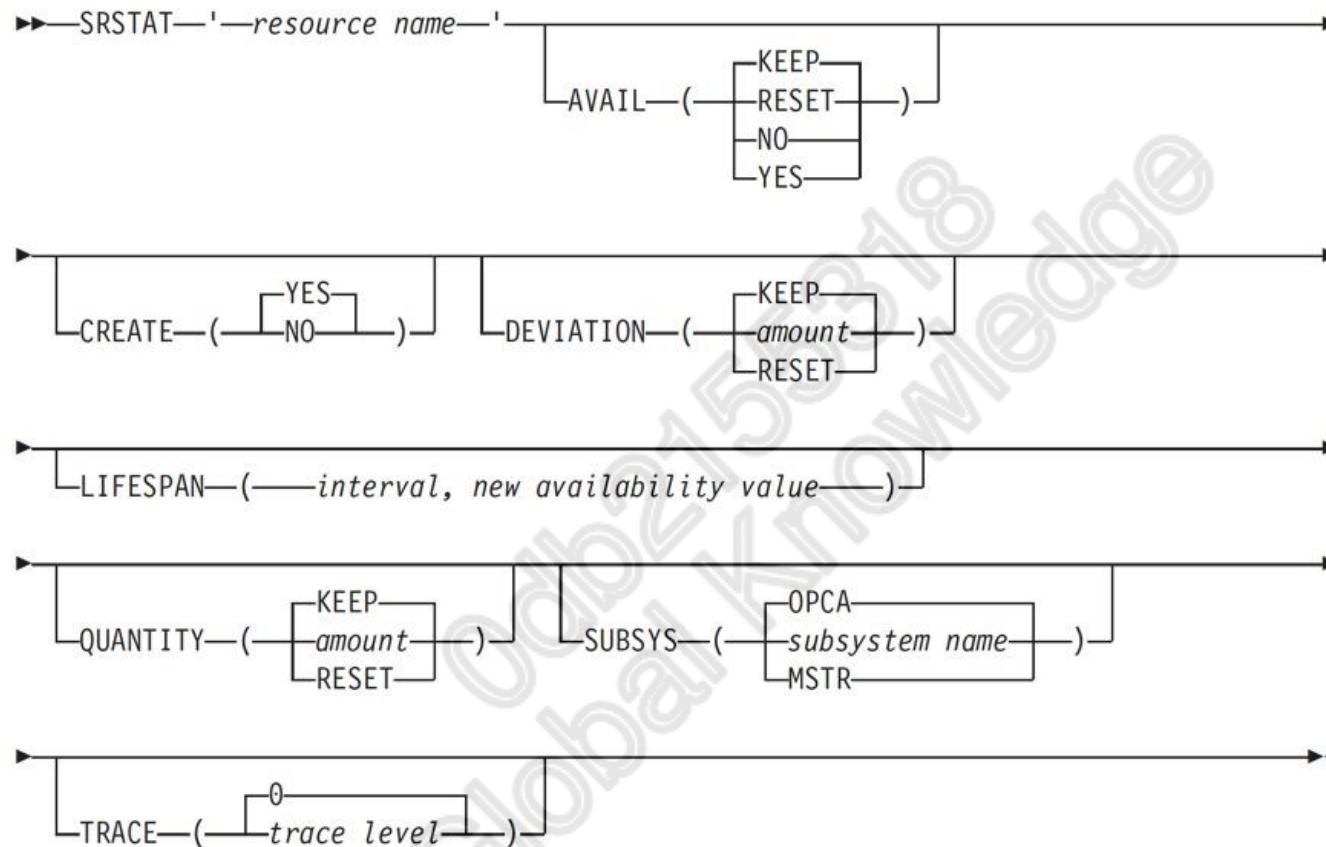
EQQEVPGM OPSTAT

OPSTAT SUBSYS(T93T) WSNAME(TCP0) ADID(TCP000) OPNUM(001) STATUS(C)

COMPLETE the OPNO(1) of the application TCP000

Odb2155318
Global Knowledge

EQQEVPGM SRSTAT



EQQEVPGM SRSTAT

PARAMETER	VALUE	DESCRIPTION
'Resname'		Special Resource Name (Maximum 44 characters)
AVAIL	<ul style="list-style-type: none"> • <u>KEEP</u> • RESET • NO • YES 	<ul style="list-style-type: none"> • KEEP, the default, does not change the availability status • RESET sets the overriding availability to blank so default value is used • NO The resource is not available • YES The resource is available
CREATE	<ul style="list-style-type: none"> • <u>YES</u> • NO 	<ul style="list-style-type: none"> • YES the SR is created into the CP even it doesn't exist in DB • NO if the SR doesn't exist in DB it's not created in the CP
DEVIATION	<ul style="list-style-type: none"> • <u>KEEP</u> • Amount • RESET 	<ul style="list-style-type: none"> • <u>KEEP</u> does not alter the deviation • Amount (-n or +n) is added to subtract to the quantity • RESET the value
LIFESPAN	(interval,new value)	<ul style="list-style-type: none"> • Interval in minutes before the value is changed • New value can be (YES, NO, RESET)
QUANTITY	<ul style="list-style-type: none"> • <u>KEEP</u> • Amount • RESET 	<ul style="list-style-type: none"> • <u>KEEP</u> does not alter the deviation • Amount 1 to 999999 • RESET to restore the planned quantity

EQQEVPGM SRSTAT

SRSTAT 'TRAINING' SUBSYS(T93T) AVAIL(N)

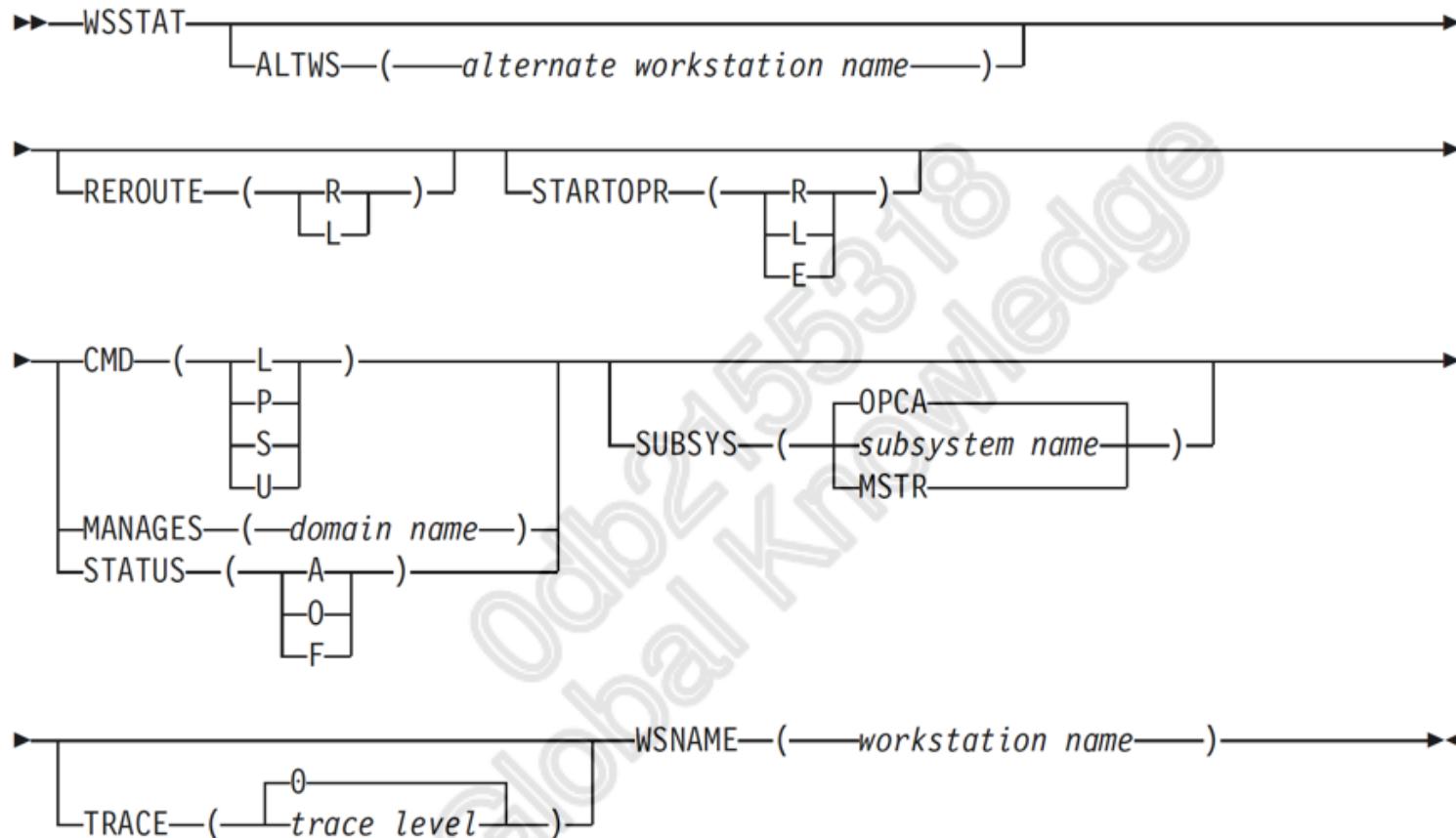
The SR Training is not available

SRSTAT 'TRAINING' SUBSYS(T93T) LIFESPAN(10,NO)

The SR Training is not available in 10 minutes

```
Defaults
  Quantity      : 1
  Available     : Yes
Active LIFESPAN : Action= N  Expiration Date= 18/03/11  11.27
```

EQQEVPGM WSSTAT



EQQEVPGM WSSTAT

PARAMETER	VALUE	DESCRIPTION
ALTWS	Alternate ws name	When the workstation status is set to offline or failed set the ALTWS
REROUTE	R/L	<ul style="list-style-type: none"> • R for operations to be rerouted to the alternate workstation • L for no rerouting
STARTOPR	R/E/L	<ul style="list-style-type: none"> • R Restart operations automatically on the alternate workstation. • E Set all started operations to ended-in-error. • L Leave the operations in started status.
CMD	L/P/S/U	<ul style="list-style-type: none"> • L To link the workstation • P To stop workstation • S To start workstation • U To unlink the workstation
MANAGES	DOMAIN	it instructs the WS to become the new domain manager of the domain specified by MANAGES
STATUS	A/O/F	<ul style="list-style-type: none"> • A Active • O Offline • F Failed
WSNAME	workstation	

EQQEVPGM WSSTAT

- Activate a workstation

WSSTAT SUBSYS(T93T) WSNAME(TCPO) STATUS(A)

EQQWL10W WORK STATION TCPO HAS BEEN SET TO ACTIVE STATUS

EQQWL15I REROUTING IS WITHDRAWN

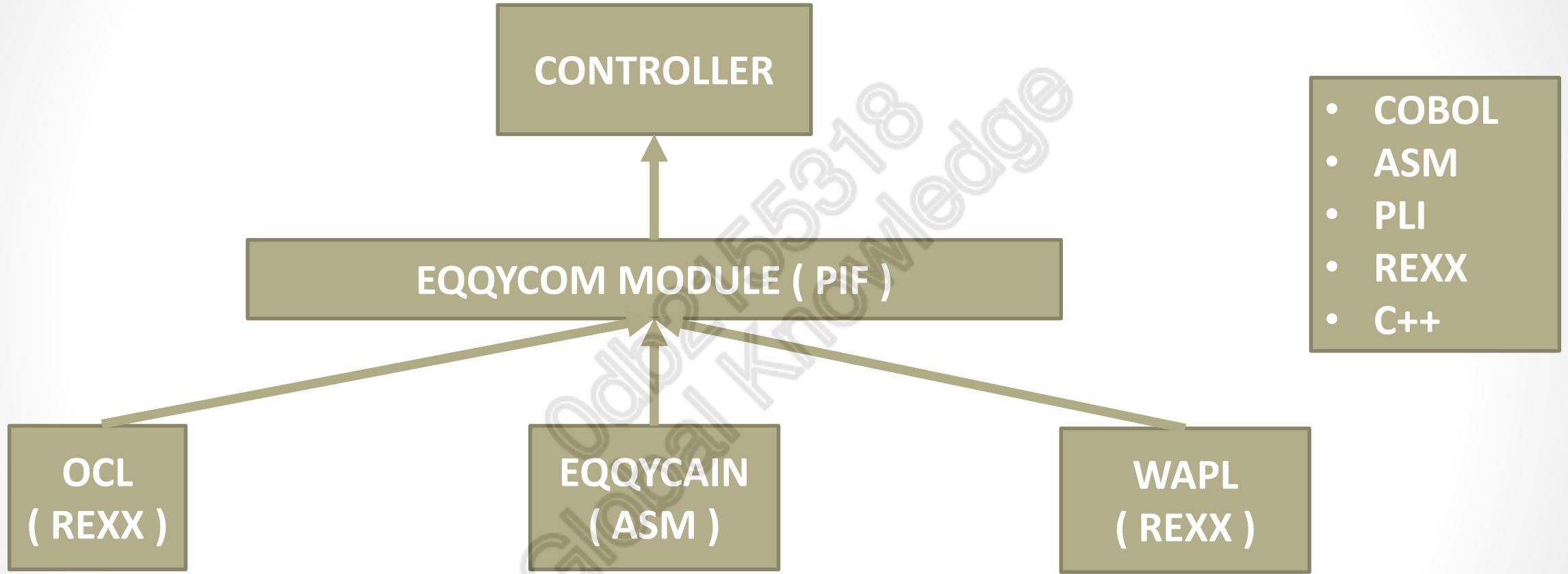
- Stop a workstation and reroute to alternate workstation

WSSTAT SUBSYS(T93T) WSNAME(TCPO) STATUS(O) START(R) ALTWS(CPU1)

EQQWL10W WORK STATION TCPO HAS BEEN SET TO OFFLINE STATUSE

QQWL13I REROUTING IN EFFECT FOR WORK STATION TCPO

PIF architecture



PIF SAMPLE 1/2

MEMBER	DESCRIPTION
EQQPIFAD	PLI : ADD APPLICATION INTO AD DATABASE
EQQPIFAP	PLI : GETJCL to resolved variables into ISPF before submitting a job
EQQPIFCB	ASM : EQQADD - add an application to the CP or LTP EQQDEL - delete an application from the CP or LTP EQQARES - add an resource to an operation in the CP EQQDRES - delete a resource from an operation in the CP EQQAPRE - add a predecessor operation to an application in the CPEQQDPRE - delete a prereq operation from an application in the CP
EQQPIFCL	ASM: DAYSTAT TSO command to determine if an input date is work day or free day
EQQPIFDJ	ASM : Delete JCL in JS file
EQQPIFJC	COBOL : Manipulate variable Tables
EQQPIFJD	PLI : Delete JCL in JS file

PIF SAMPLE 2/2

MEMBER	DESCRIPTION
EQQPIFJX	ASM : Sample to maintain the JCL repository
EQQPIFOP	REXX : Add an ADHOC application into the Current Plan
EQQPIFPR	REXX : list all cyclic period
EQQPIFWI	PLI : sample to modify capacity values in an open interval of a current plan workstation

PIF REQUEST TYPE

TYPE	Description
DELETE	Delete Data items
EXECUTE	Perform an actual update of the current plan
INIT	Initialize session with the controller
INSERT	Inserts new data items or additional information into existing data items.
LIST	Retrieve a list of data items
MODIFY	Modifies data fields in the LTP or current plan, or identifies CP or LTP data items for further modifications
OPTIONS	Specifies options to be used when performing PIF requests
REPLACE	Replaces an existing application description or operator instruction
RESET	Cancels a series of modify current plan requests if performed before the EXECUTE request
SELECT	Retrieves a single data item in detail
SETSTAT	Modifies the status of a condition dependency
TERM	Terminates the communication session between your program and the Controller

PIF SEGMENT/COMMAND

Type	Description	Delete	Insert	List	Modify	Replace	Select
AD	Application description record	Y (C)	Y (C)			Y	Y (C)
ADCOM	Application description common segment			Y (C)			Y (C)
ADKEY	Application description key			Y (C)			
AWSCL	All workstations closed record	Y	Y			Y	Y
CL	Calendar Record	Y	Y			Y	Y (C)
CLCOM	Calendar Common Segment			Y (C)			Y (C)
CPCOND	Current Plan Condition	Y (C)	Y (C)		Y (C)		Y (C)
CPCONDCO	Current Plan Condition common segment			Y (C)			Y (C)
CPEXT	Current Plan operation extended name			Y	Y (C)		
CPOC	Current Plan occurrence	Y (C)	Y (C)	Y (C)	Y (C)		Y (C)
CPOCPRE	A predecessor of a current plan occurrence	Y (C)	Y (C)				
CPOP	Current Plan operation	Y (C)	Y (C)		Y (C)		Y (C)

PIF SEGMENT/COMMAND

Type	Description	Delete	Insert	List	Modify	Replace	Select
CPOPCOM	Current plan operation common segment			Y (C)			Y (C)
CPOPSRU	Current Plan Operation related to SR	Y	Y				Y
CPPRE	Current Plan Predecessor	Y (C)	Y (C)				
CPREND	Distributed Remote Job info				Y (C)		
CPRENZ	z/OS Remote Job Info				Y (C)		
CPSAI	System Automation for CP oper		Y		Y (C)		
CPSIMP	Current Plan Condition dependency	Y (C)	Y (C)				
CPSR	Current Plan Special Resource	Y (C)	Y (C)				
CPST	Current plan status record						Y (C)
CPSUC	Current Plan Successor	Y	Y				
CPUSRF	Current Plan User Field	Y	Y		Y		Y
CPWS	Current Plan Workstation				Y		Y (C)
CPWSV	Current Plan Virtual Workstation				Y		Y

PIF SEGMENT/COMMAND

Type	Description	Delete	Insert	List	Modify	Replace	Select
CPWSCOM	Current Plan Workstation			Y (C)			Y (C)
CPWSVCOM	Current Plan Virtual Workstation			Y (C)			Y
CSR	Current Plan Special Resource				Y	Y	Y
CSRCOM	Current Plan Special Resource			Y			Y
ETT	Event Triggered Tracking	Y	Y			Y	Y
GENDAYS	Run Date			Y			
IVL	Current Plan Workstation Interval	Y	Y		Y		
JCLPREP	Retrieve promptable setup variables			Y			Y (C)
JCLPREPA	Resolve all nonpromptable setup variables						Y (C)
JCLV	JCL variable table record			Y		Y	Y (C)
JCLVCOM	JCL variable table common segment	Y		Y (C)			Y (C)
JL	JS file JOBLOG record	Y					
JLCOM	JS file job log common segment			Y			Y

PIF SEGMENT/COMMAND

Type	Description	Delete	Insert	List	Modify	Replace	Select
JS	Job control language record	Y (C)	Y			Y	Y (C)
JSCOM	JCL common segment	Y (C)		Y (C)			Y (C)
LTCPRE	LTP conditional predecessor	Y (C)					
LTOC	LTP occurrence	Y (C)	Y (C)		Y (C)		Y (C)
LTOCCOM	LTP occurrence common segment			Y (C)			Y (C)
LTPRE	LTP predecessor segment	Y (C)	Y (C)				
OI	Operator instruction	Y (C)	Y			Y	Y (C)
OICOM	Operator instruction Common Segment			Y (C)			Y (C)
PR	Period	Y	Y			Y	Y (C)
PRCOM	Period common segment			Y (C)			Y (C)
RG	Run cycle group	Y (C)	Y			Y	Y (C)
RGCOM	Run cycle group common segment			Y (C)			Y (C)
RGKEY	Run cycle group key segment			Y (C)			

PIF SEGMENT/COMMAND

Type	Description	Delete	Insert	List	Modify	Replace	Select
SR	Special resource	Y	Y			Y	Y
SRCOM	Special resource common segment			Y			
VIVL	Current plan virtual workstation destination interval	Y	Y		Y		
WS	Workstation	Y	Y			Y	Y (C)
WSCOM	Workstation description common segment			Y (C)			Y (C)
WSV	Virtual workstation destination	Y	Y			Y	Y
WSVCOM	Virtual workstation destination common segment			Y (C)			Y
XENV	Execution environment segment						Y

EQQYCAIN

Odb2155318
Global Knowledge

EQQYCAIN FUNCTION

- BATCH TOOL
- LIST APPLICATION/GROUP
- MODIFY CP/LTP
- DELETE OBJECTS
- INSERT IN CP/LTP
- COPY OBJECTS
- EXPORT/IMPORT

Odb2155318
Global Knowledge

EQQYCAIN JCL

```
//STEP EXEC PGM=EQQYCAIN,  
// PARM='T93C,MSGOFF,,MISSCP',  
// REGION=0M  
//STEPLIB DD DISP=SHR,DSN=TWS930.SEQQLMD0  
//EQQYPARM DD DISP=SHR,DSN=TWS930.SEQQPARM(T93CINIT)  
//EQQMLIB DD DISP=SHR,DSN=TWS930.SEQQMSG0
```

The EQQYPARM MEMBER OVERRIDE THE CONTROLLER NAME

EQQYCAIN PARM

Description	Maximum Length	Default	Values
SUBSYSTEM NAME	4	OPCA	Controller Name
WTO Desired	6	MSGON	MSGOFF, MSGNONE
IA date	6	Current Date	CPSTDA / yyymmdd
IA Time	4	Current Time	hhmm
CP needed	6	MUSTCP	MISSCP/MUSTCP

EXAMPLES:

PARM='T93C, MSGOFF, CPSTDA, 1400'

PARM='T93C,,180312'

PARM='T93C,,,MISSCP'

EQQYCAIN PARM

- **CPSTDA** Means that the default IA date is the starting date of the CP.
- **MISSCP** Means that the requested BCIT functions do not require the existence of a CP and that BCIT should not try to access it.
- **MSGNONE** Means that WTOs (including those issued in case of errors or for commands that do not have performance impacts) are not to be issued any longer.
- **MSGOFF** Means that WTOs (except those issued in case of errors or for commands that do not have performance impacts) are not to be issued any longer.
- **MSGON** Means that WTOs are to be issued in any case. It is strongly recommended not to use the default MSGON when the BCIT commands invoked might produce too many WTOs.
- **MUSTCP** Requests that BCIT accesses the CP, no matter if the requested functions need the existence of CP.

EQQYCAIN OPTIONS

OPTION	DESCRIPTION	VALUE
ADOICHK	AD/OI consistency when application is modified or deleted	Y/N
ADVERS	With Yes the validity is like in ISPF	Y/N
BL	Ask the Batch Loader format	Y/N
BLPRT	If Y write in AD or OI File	Y/N
CPEDPR	resolve automatically external dependences during insert in CP	Y/N
ERROR	When Y is specified, processing of SYSIN statements stops in case of failure	Y/N
LTPEPR	resolve automatically external dependences during insert in LTP	Y/N
LTP	If Y insert is specified, if Insert or Delete into the CP and the IA is later than CP end the requested action will performed into the LTP	Y/N

EQQYCAIN DDNAME

DDNAME	DESCRIPTION	LRECL
AD	If arguments BL=Y and BLPRT=N are coded in the action OPTIONS, the identifier, status, and VALTO of each listed application (AD) will be written in the file referenced by the AD DD card	23
BATCHL	If arguments BL=Y and BLPRT=Y are coded in the action, LIST and SELECT of ADs and OI results will be formatted as BATCH LOADER	80
CPCOND	If argument BL=Y is coded in the OPTIONS action, the ACTION=LIST,RESOURCE=CPCONDCO,ADID=XXXX*	80
CPOC	If argument BL=Y is coded in the OPTIONS action, the ACTION=LIST,RESOURCE=CPOC,ADID=XXXX*	80
CPOP	If argument BL=Y is coded in the OPTIONS action, the ACTION=LIST,RESOURCE=CPOPCOM,ADID=XXXX*	80
DATAFI	File that contains the output of LIST JCLVCOM	
SYSPRINT	Required, it is used to print results of LIST and SELECT actions	
ERREUR	Required	

EQQYCAIN DDNAME

DDNAME	DESCRIPTION	LRECL
EQQDUMP	File that contains information to understand diagnostic data set error codes.	
EQQMLOG	File that contains the scheduler messages for return code of 6 or higher.	
EXPORTAD	If the action is an EXPORT and the resource is an AD, the file will contain AD	
EXPORTOI	If the action is an EXPORT and the resource is an OI, the file will contain OI	
EXPORTRG	If the action is an EXPORT and the resource is an RG, the file will contain RG	
IMPORTAD	If the action is an IMPORT and the resource is an AD the file will cotain AD	
IMPORTOI	If the action is an IMPORT and the resource is an OI the file will cotain OI	
IMPORTRG	If the action is an IMPORT and the resource is an RG the file will cotain RG	
OI	If arguments BL=Y and BLPRT=N are coded in the action OPTIONS, the identifier and operation number of each listed operator instruction(OI) will be written in the file referenced by the OI DD card	
RG	If arguments BL=Y and BLPRT=N are coded in the action OPTIONS, the identifier of each listed run cycle group (RG) will be written in the file referenced by the RG DD card	

EQQYCAIN ACTION

ACTION	DESCRIPTION
COPY	The COPY instruction copies an AD, JCLV, OI, or RG
DELETE	The DELETE instructions deletes a record from the database, CP or LTP.
EXPORT	EXPORT AD,OI,RG
IMPORT	IMPORT AD,OI,RG
INSERT	The INSERT instruction inserts resources into the corresponding tables.
LIST	With the LIST instruction, you can list all the resources of the scheduler
LISTSTAT	The LISTSTAT instruction returns a code that reflects the status of the resource. You can list the status for CPOC or CPOPCOM resources
MODIFY	The MODIFY instruction modifies some resources in the CP and LTP
SELECT	The SELECT instruction selects a resource
SETSTAT	The SETSTAT request change to CP condition dependency

EQQYCAIN SAMPLE

File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT TWS930.CNTL (SAVEAD1) - 01.02 Member SAVEAD1 sav
Command ==> Scroll ==> CS

```
***** Top of Data *****  
000001 //TWS930CA JOB NOTIFY=&SYSUID,REGION=0M  
000002 //STEP EXEC PGM=EQQYCAIN,  
000003 //      PARM='T93C,MSGOFF,,,MISSCP',  
000004 //      REGION=0M  
000005 //STEPLIB   DD    DISP=SHR,DSN=TWS930.SEQQLMD0  
000006 //EQQMLIB   DD    DISP=SHR,DSN=TWS930.SEQQMSG0  
000007 //BATCHL    DD    DISP=SHR,DSN=TWS930.BATCHL(APPLI1)  
000008 //AD         DD    SYSOUT=*  
000009 //ERREUR    DD    SYSOUT=*  
000010 //SYSOUT    DD    SYSOUT=*  
000011 //EQQMLOG   DD    SYSOUT=*  
000012 //EQQDUMP   DD    SYSOUT=*  
000013 //SYSPRINT  DD    SYSOUT=*  
000014 //SYSUDUMP  DD    SYSOUT=*  
000015 //SYSIN     DD    *  
000016 ACTION=OPTIONS,BL=Y,BLPRT=Y;  
000017 ACTION=LIST,RESOURCE=ADCOM,ADID=APPLI1,TYPE=*. 
```

EQQYCAIN BATCH LOADER RESULT

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
EDIT          TWS930.BATCHL(APPLI1) - 01.00          Columns 00001 0007
Command ==> _____ Scroll ==> CSR
***** * ***** Top of Data *****
000001 OPTIONS DURUNIT(SECONDS)
000002 ADSTART ACTION(ADD)
000003 ADID(APPLI1)           ) ADVALFROM(180228)
000004 ADTYPE(A)
000005 OWNER('APPLI1')        ')
000006 PRIORITY( 5) ADSTAT(A)
000007 ADRUN ACTION(ADD)
000008 NAME(R1      )       RULE(3) VALFROM(180228)   VALTO(711231)
000009 SHIFT( 0)             SHSIGN(F)
000010 TYPE(R)
000011 IATIME(1000) DLDAY( 0)   DLTIME(1100)
000012 ADRULE EVERY DAY(DAY) WEEK
000013 ADOP ACTION(ADD)
000014 OPNO( 1)               WSID(BEG )   ADOPCATM(N)
000015 CONDRJOB(N)
000016 ADOPNOP(N)
000017 ADOPMH(N)
000018 DURATION(000001)
000019 AEC(Y)
```

EQQYCAIN SAMPLE

INSERT occurrence into the CP

```
ACTION=OPTIONS,BL=N,BLPRT=N;  
ACTION=INSERT,RESOURCE=CPOC,ADID=APPLI1.
```

DELETE APPLICATION INTO THE AD

```
ACTION=OPTIONS,BL=N,BLPRT=N;  
ACTION=DELETE,RESOURCE=AD,ADID=APPLI1.
```

The last SYSIN Line must finished by ":" otherwise there is a return code 2

OCL

Odb2155318
Global Knowledge

OCL FUNCTION

Using OCL, you can adjust and experiment with:

- JCL variables
- Applications and related dependencies
- Application occurrences status
- Special resources
- Applications structure
- User-supplied dates

OCL is standard REXX function + IWS command

OCL JCL

```
//T93COCL1 JOB OPTM,BX670M,NOTIFY=&SYSUID,CLASS=B,MSGCLASS=B  
//MYJCLLIB JCLLIB ORDER=TWS930.CNTL  
//EQQOCL EXEC OCLPROC  
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FB,LRECL=133,BLKSIZE=1330)  
//SYSTSPRT DD SYSOUT=*  
//EQQOCL.SYSIN DD *  
CHKAPPL APPL(CRITICAL)
```

OCL PROC

```
//EQQYRPRC PROC  
//EQQOCL      EXEC PGM=IKJEFT01,PARM='EQQOCL1',REGION=0M  
//STEPLIB     DD DISP=SHR,DSN=TWS930.SEQQLMD0  
//           DD DISP=SHR,DSN=FAN140.SEAGLPA  
//           DD DISP=SHR,DSN=FAN140.SEAGALT  
//OCLLOG      DD SYSOUT=*  
//EQQMLIB     DD DISP=SHR,DSN=TWS930.SEQQMSG0  
//OCLPARM     DD DISP=SHR,DSN=TWS930.SEQQPARM(T93COCL)  
//OCLMLIB     DD DISP=SHR,DSN=TWS930.SEQQSAMP(EQQYRMSG)  
//SYSEXEC     DD DISP=SHR,DSN=TWS930.SEQQMISC  
//SYSPRINT    DD SYSOUT=*,DCB=(RECFM=FB,LRECL=133,BLKSIZE=1330)  
//CARDIN      DD UNIT=SYSDA,SPACE=(TRK,(20,200)),  
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)  
//SYSTSPRT    DD SYSOUT=*  
//SYSTSIN     DD DUMMY  
//EQQMLOG     DD SYSOUT=*  
//EQQDUMP     DD SYSOUT=*
```

OCL PARMLIB

INSTRUCTION	DESCRIPTION
SUBSYS(T93C)	Identifies the IWSz subsystem Controller
OPCTRK(T93T)	Identifies the IWSz subsystem Tracker
REMHOST(ZOS220)	Identifies the IWSz Remote Server
REMPORT(5930)	Identifies the IWSz Remote port
DEFONO(001)	Specifies the default operation number
DEFPREOPNO(255)	Specifies the default external predecessor operation number used only for external dependency
DEFCONDID(001)	Specifies the default condition id; used only for conditional dependency
SORT(MAX)	MAX OCL will take the occurrence with the latest occurrence IA MIN OCL will take the occurrence with the earliest occurrence IA
TSOCMD(YES)	YES the user can specify TSO commands or REXX NO the user can specify only OCL instructions
INCVALUE(10)	This value indicates the increasing value for the variables calculation
DUPROC(N)	Y The BPX1SDD routine is invoked by using the DUBPROCESS parameter N The BPX1SDD routine is invoked by using the DUBPROCESSDEFER parameter

OCL COMMAND

INSTRUCTION	DESCRIPTION
ADD	Insert Application/group into the CP
ADDCOND	Adds a condition definition to an operation in the CP
ADDOP	Inserts an operation into an occurrence in the CP
ADDPRED	Adds a predecessor dependency definition to an occurrence in the CP
ADDRES	Adds a special resource to an operation in the CP
ADDSIMP	Adds a single dependency definition to a condition in the CP
ADOPSAI	Specifies system automation information for the operation
CALL	Invokes a REXX procedure
CHGEXTNAME	Defines or modifies the extended job into the CP
CHJOB	Changes the jobname associated with the operations of an occurrence in the CP
CHJOBSAI	Defines or modifies the system automation information associated with the operations in the CP
CHKAPPL	Verifies the presence of occurrences in the CP

OCL COMMAND

INSTRUCTION	DESCRIPTION
CHKDATE	Checks the validity of an input date and generate 200 variables
COMPL	Complete occurrences or operation into the CP
DEL	Delete occurrences and operations from CP or LTP
DELCOND	Deletes a condition from an operation in the CP
DELPRED	Deletes a dependency definition from an occurrence in the CP
DELRES	Deletes a special resource from an operation in the CP
DELSIMP	Deletes a single dependency definition from a condition in the CP
EXIT	Sets the program exit return code
FORCE	Forces the execution of the application
GOTO	Moves to a point specified by a LABEL statement within the OCL program
HOLD	Sets the operations of an occurrence to HOLD status
IF-THEN-ELSE	Performs a relational test and, depending on the result, executes OCL function

OCL COMMAND

INSTRUCTION	DESCRIPTION
JSUACT	Activates or inactivates the job submission function in the z/OS
INIT	Specifies the ID of a table
KILLJOB	Kill job on distributed workstation
KILLREC	Kill recovery job on distributed workstation
LABEL	Defines labels within the OCL program SYSIN
MODCOND	Modifies the condition defined for an operation in the CP
MODOP	Modifies the operations details
NOP	NOP an operation into the CP
OPSTAT	Changes the status of an operation in the CP (EQQEVPGM Also)
PROMPTN	Specifies that NO is the reply to a recovery prompt issued for an abended operation
PROMPTY	Specifies that YES is the reply to a recovery prompt issued for an abended operation
RELEASE	Releases the operations of an occurrence that are in HOLD status

OCL COMMAND

INSTRUCTION	DESCRIPTION
RELOP	Releases the internal successors of an operation
RELSUCC	Releases the successors of the occurrence by deleting the external dependency definition
SET	Sets a user variable
SETUPD	Sets a user variable and updates its default value in an variable table
SRSTAT	Modifies the availability status of a special resource (EQQEVPGM Also)
UNNOP	Restores an operation from NOP status
UPD	Updates the default value of a user variable in variable table
WSSTAT	Modifies the status of a workstation (EQQEVPGM Also)
WTO	Displays messages on the system console and waits for a reply

OCL SAMPLE 1 : Update variable table

OCL SAMPLE 1 : Content of table after the JCL

----- MODIFYING VARIABLES IN A TABLE -----						Row 1 of
						Scroll ==> CSR
Enter/change data in the rows below, and/or enter any of the row commands below						
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete, S - Select variable details.						
Variable table : OCL						
OWNER ID ==> TWS930						
TABLE DESCRIPTION ==> Training table						
Row	Variable	Subst.	Setup	Val	Default	
cmd	Name	Exit		req	Value	
'''	DPRE		N	N	180309	
'''	DATD		N	N	180311	
'''	LDNM		N	N	180430	
'''	NOPD		N	N	180312	
'''	YM		N	N	1803	
'''	LDCM		N	N	180331	
'''	FDCM		N	N	180301	
'''	FMM1		N	N	180430	
'''	DATEPJO		N	N	180312	
***** Bottom of data *****						

OCL SAMPLE 2 : Check application

OCL SAMPLE 2 : Check application

```
Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY T93COCL1 JOB06482 DSID 104 LINE 0 COLUMNS 02- 81
COMMAND INPUT ===> SCROLL ===> CSR
***** TOP OF DATA *****

*****
OPC CONTROL LANGUAGE (OCL) **
*****
Execution started on 11 Mar 2018 at 09:18:43.366023 **
*****
EQQCL01I ======
EQQCL00I Processing: CHKAPPL APPL(APPLI1)
EQQCL0JI Searching for occurrence APPLI1 in CP
EQQCL00I Occurrence found: APPL(APPLI1) IA(1803101000) STATUS(C)
EQQCL00I Occurrence found: APPL(APPLI1) IA(1803100932) STATUS(W)
EQQCL0KI Total number of matching occurrences: 2
READY
END
***** BOTTOM OF DATA *****
```

IWS JCL command

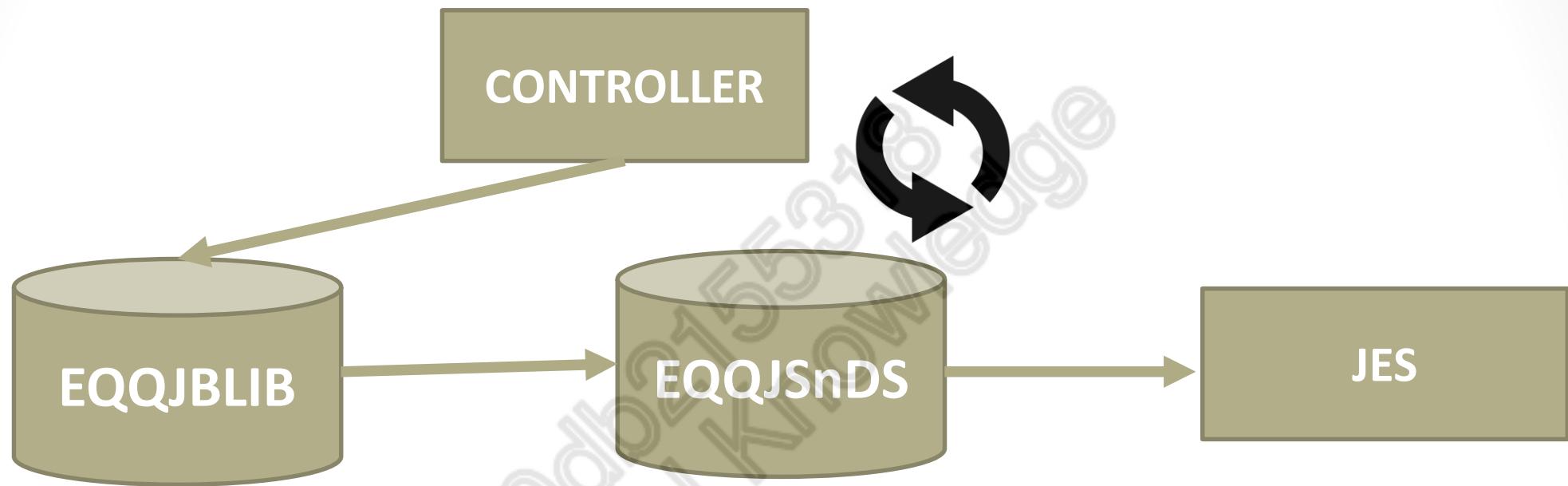
Odb2155318
Global Knowledge

JCL Command

BEFORE	///*OPC
AFTER	//*>OPC

COMMAND	DESCRIPTION
RECOVER	Recovery scenario
NOP	Job doesn't run and set to complete
SCAN	If VARSUB(SCAN) coded in PARMLIB must be coded to setup variables
SEARCH	Defines the variable tables that are searched
SETFORM	Defines the format of dynamic-format supplied variables
SETVAR	Creates a temporary variable
TABLE	Defines a variable table that will be searched
BEGIN/END	These directives, used in pairs, denote the following, depending on the value of the ACTION keyword
FETCH	This directive lets you include lines, fetched from a PDS member or supplied by an exit, in your job
COMP	A comparison expression lets you specify conditions when BEGIN and FETCH directives will be honored

JCL Processes



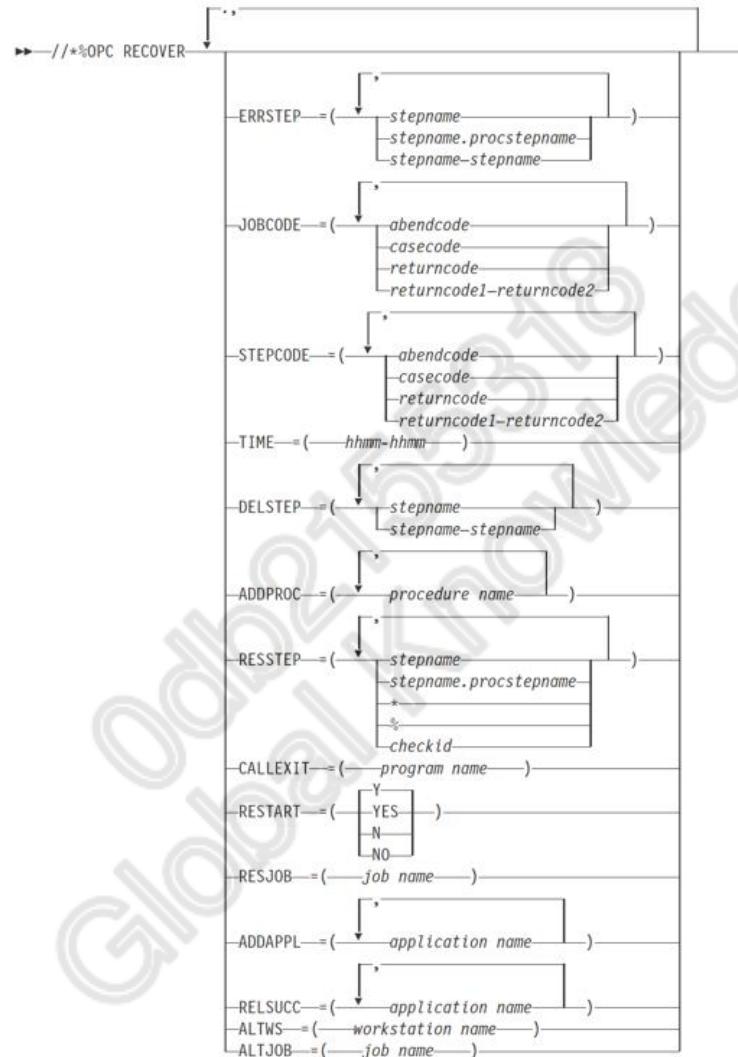
The controller write the Job into the EQQJSnDS before submitting it

The JCL Directive are modified at the SETUP/SUBMIT process

The Recovery directive are managed if the job is on error and the modification occurs only on EQQJSnDS

Using GETJCL CLIST permit to update the variables included into the JCL from EQQJSnDS

RECOVER



RECOVER

PARAMETER	TYPE	DESCRIPTION
ERRSTEP	SELECTION	Restricts the RECOVER statement to be valid only for those steps specified
JOBCODE	SELECTION	Restricts the RECOVER statement to be valid only for those job codes
STEPCODE	SELECTION	Restricts the RECOVER statement to be valid only for those stepcode
TIME	SELECTION	Restricts the RECOVER statement to be valid only in the time range
DELSTEP	REBUILD	Specifies a step or a list of steps that should be deleted
ADDPROC	REBUILD	Specifies the name, or a list of names, of JCL procedure
RESSTEP	REBUILD	Specifies the name step at which the operation should be restarted
CALLEXIT	REBUILD	Specifies the name of an exit that should be called before the restart
RESTART	ACTION	Specifies if the occurrence is to be restarted
RESJOB	ACTION	Specifies the name of the job from which the occurrence must be rerun
ADDAPPL	ACTION	Specifies an application to be added as occurrences in the current plan
RELSUCC	ACTION	Specifies the application ID of a successor occurrence
ALTWS	ACTION	Specifies the name of an alternate workstation on which to run the job
ALTJOB	ACTION	Specifies an alternate jobname to use when the job is restarted

RECOVER SAMPLE JCL

```
//%OPC RECOVER JOBCODE=S*37,ADDPROC=SPACECHG  
//%OPC RECOVER JOBCODE=(*,16-4095),RESTART=N,ADDAPPL=REORG  
//%OPC RECOVER JOBCODE=12  
//%OPC RECOVER JOBCODE=(U046,S*37),ERRSTEP=STEP01,  
//%OPC ADDPROC=(A66RECOV,SPACECHG)  
//%OPC RECOVER ERRSTEP=UPDATE,ADDAPPL=RECOV1,  
//%OPC RESTART=N,TIME=0800-1600  
//%OPC RECOVER ERRSTEP=UPDATE,ADDAPPL=RECOV1,TIME=0000-2400  
//%OPC RECOVER TIME=0000-2400  
//%OPC RECOVER ERRSTEP=SAX91,JOBCODE=S0C4,TIME=0000-2400,  
//%OPC DELSTEP=SAX80-SAX91,ADDPROC=SALRECOV
```

NOP Directive

```
///*%OPC SCAN  
///*%OPC NOP  
//TCP00000 JOB SETD,TWS930,NOTIFY=TWS930,CLASS=A,MSGCLASS=H  
//RETWT1 EXEC PGM=IEFBR14  
///*%OPC NOP
```

The JCL is not submitted

The operation is complete when all dependencies are solved.

You can code NOP before or after the JOB

SCAN Directive

The SCAN directive is honored only if the VARSUB parameter of the OPCOPTS statement is set to SCAN

If VARSUB(SCAN) /*OPC SCAN is mandatory

If VARSUB(NO) the variable are never updated

If VARSUB(YES) the variable are always updated

```
/*%OPC SCAN  
//&OJOBNAME JOB SETD,TWS930,NOTIFY=TWS930,CLASS=A,MSGCLASS=H  
//RETWT1 EXEC PGM=IEFBR14
```

The OJOBNAME variable is replace by the Jobname of operation

```
//&OJOBNAME JOB SETD,TWS930,NOTIFY=TWS930,CLASS=A,MSGCLASS=H  
/*%OPC SCAN  
//RETWT1 EXEC PGM=IEFBR14
```

The OJOBNAME variable is not replace and the JOB is in error with OJCV code.

IWS Variables

- Occurrence-related variables (34 variables) (O...)
- Operation-related variables (13 variables) (O...)
- Date-related variables (20 variables) (C...)
- Dynamic-format variables defined with SETFORM (14 variables) (C.. And O..)

IWS Variables : Occurrence-related variables

Variable	Length	Description	Date=2018/03/11 Day 78 in 2018
OADID	16	Application ID	
OADOWNER	16	Occurrence Owner	
OAUGROUP	8	Authority Group	
OCALID	16	Calendar name	
ODAY	1	1-7 = M-S	7
ODD	2	Occurrence Day in DD format	11
ODDD	3	Occurrence Day in DDD format	70
ODMY1	6	Occurrence Date in DDMMYY format	190318
ODMY2	8	Occurrence Date in DD/MM/YY format	19/03/2018
OETCRIT	44	Event triggering policy name from the ETT table	
OETEVNM	44	Complete ETT event name	
OETGGEN	8	GDG data set generation number (GnnnnVnn)	

IWS Variables : Occurrence-related variables

Variable	Length	Description	Date=2018/03/19 Day 78 in 2018
OETGROOT	35	GDG data set root. For the ETT event type R	
OETJNUM	8	Job number associated with the OETJOBN	
OETJOBN	8	The complete job name that triggered the ETT	
OETTYPE	1	Event type of the ETT table entry (J=Job, R=Resource).	
OFREEDAY	1	The occurrence date is a free day (F) or a workday (W).	F
OHH	2	Occurrence input arrival hour in HH format.	17
OHHMM	4	Occurrence input arrival hour and minute in HHMM	1714
OMM	2	Occurrence input arrival month in MM format.	03
OMMYY	4	Occurrence input arrival month and year in MMYY	0318
OWW	2	Occurrence input arrival week of the year in WW format	10
OWWD	3	Occurrence input arrival week, and day within week	107

IWS Variables : Occurrence-related variables

Variable	Length	Description	Date=2018/03/19 Day 78 in 2018
OWWLAST	1	A value, Y (yes) or N (no), that indicates whether the occurrence date is in the last week of the month.	N
OWWMONTH	1	A value between 1 and 6 that indicates the occurrence input arrival week-in-month, where each new week	2
OYM	6	Occurrence input arrival month within year in YYYYMM	201803
OYMD	8	Occurrence input arrival date in YYYYMMDD format.	20180311
OYMD1	6	Occurrence input arrival date in YYMMDD format.	180311
OYMD2	8	Occurrence input arrival date in YY/MM/DD format.	18/03/11
OYMD3	10	Occurrence input arrival date in YYYY/MM/DD format.	2018/03/11
OYY	2	Occurrence input arrival year in YY format.	18
OYYDDD	5	Occurrence input arrival date as a Julian date in YYDDD	18070
OYMM	4	Occurrence input arrival month within year in YYMM	1803
OYYYY	4	Occurrence input arrival year in YYYY format	2018

IWS Variables : operation-related variables

Variable	Length	Description	Date=2018/03/19 Day 78 in 2018
OJOBNAME	8	Operation jobname	
OLDAY	1	Operation latest start day (1-7); 1 Monday, 7 Sunday	7
OLDD	2	Operation latest start day (day in the month).	11
OLHH	2	Operation latest start hour	17
OLHHMM	4	Operation latest start in hours and minutes	1740
OLMD	4	Operation latest start time in MMDD format	0311
OLMM	2	Operation latest start month, in MM format	03
OLWK	2	Operation latest start week in WW format.	10
OLYMD	6	Operation latest start date in YYMMDD format	180311
OLYYDDD	5	Operation latest start in Julian date format (YYDDD)	18070
OOPNO	3	Operation number within the occurrence	005
OWSID	4	Workstation ID for current operation.	CPU1
OXJOBNAM	54	Extended job name at the operation level	

IWS Variables : Date-related supplied variables

Variable	Length	Description	Date=2018/03/19 Day 78 in 2018
CDAY	1	Current day of the week; 1 Mon, 7 Sun	7
CDD	2	Current day of month in DD format.	11
CDDD	3	Day number in the current year.	070
CDDMMYY	6	Current date in DDMMMYY format.	110318
CFREEDAY	1	Freeday or Workday	F
CHH	2	Current time in HH format.	17
CHHMM	4	Current hour and minute in HHMM format.	1753
		Current hour, minute, and second in	175359
CHHMMSS	6	HHMMSS format.	
		Current hour, minute, second, and	17535922
CHHMMSSX	8	hundredths of seconds in HHMMSSXX	
CMM	2	Current month in MM format.	03

IWS Variables : Date-related supplied variables

Variable	Length	Description	Date=2018/03/19 Day 78 in 2018
CMMYY	4	Current month within year in MMYY format.	0318
CWW	2	Week number in the current year.	10
CWWD	3	Current day within week in WWD format	107
CYMD	8	Current date in YYYYMMDD format.	20180311
CYY	2	Current year in YY format.	18
CYYDDD	5	Current Julian date in YYDDD format.	18070
CYYMM	4	Current month within year in YYMM format.	1803
CYYMMDD	6	Current date in YYMMDD format.	180311
CYYYY	4	Current year in YYYY format,	2018
CYYYYMM	6	Current month within year in YYYYMM	201803

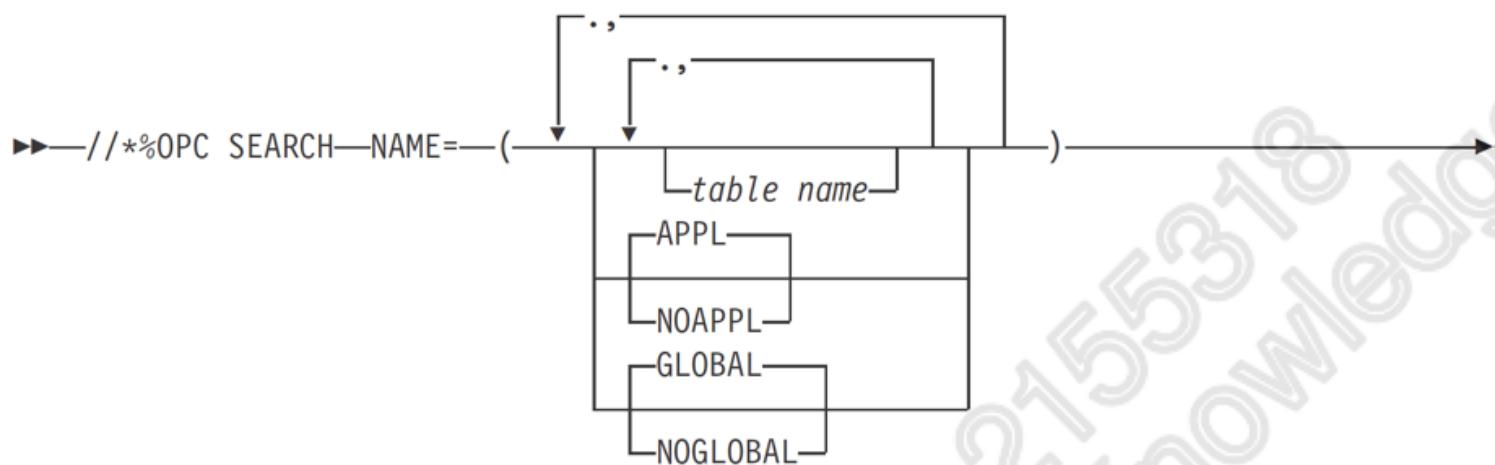
IWS Variables :Dynamic-format supplied variables

Variable	Description	Date=2018/03/19 Day 78 in 2018
CDATE	Current date.	
CTIME	Current time.	
OCDATE	Occurrence input arrival date.	
OCFRSTC	First calendar day in month of the occurrence input arrival date.	
OCFRSTW	First work day in the month of the occurrence input arrival date.	
OCFRSTWY	First work day in the year of the occurrence input arrival date.	
OCLASTC	Last calendar day in the month of the occurrence input arrival date.	

IWS Variables :Dynamic-format supplied variables

Variable	Description	Date=2018/03/19 Day 78 in 2018
OCLASTW	Last work day in the month of the occurrence input arrival date.	
OCLASTWY	Last work day in the year of the occurrence input arrival date.	
OCTIME	Occurrence input arrival time (hours and minutes).	
OPIADATE	Operation input arrival date (if blank, this takes the value of the occurrence input arrival date).	
OPIATIME	Operation input arrival time (if blank, this takes the value of the occurrence input arrival time).	
OPLSDATE	Operation latest start date.	
OPLSTIME	Operation latest start time.	

SEARCH Directive



This directive defines the variable tables that are searched when attempting to assign a variable a value

NAME(table name,..., APPL|NOAPPL, GLOBAL|NOGLOBAL) Identifies the variable tables you want searched, and in what order.

Some example :

```
/*%OPC SEARCH NAME=(&OADDID)  
/*%OPC SEARCH NAME=APPL
```

SETFORM Directive

```
►►// *%OPC SETFORM—dynamic-variable-name—=(—format—)►►
```

```
/*%OPC SETFORM CDATE=(XX,YY/MM/DDD,TT)  
/* CDATE &CDATE
```

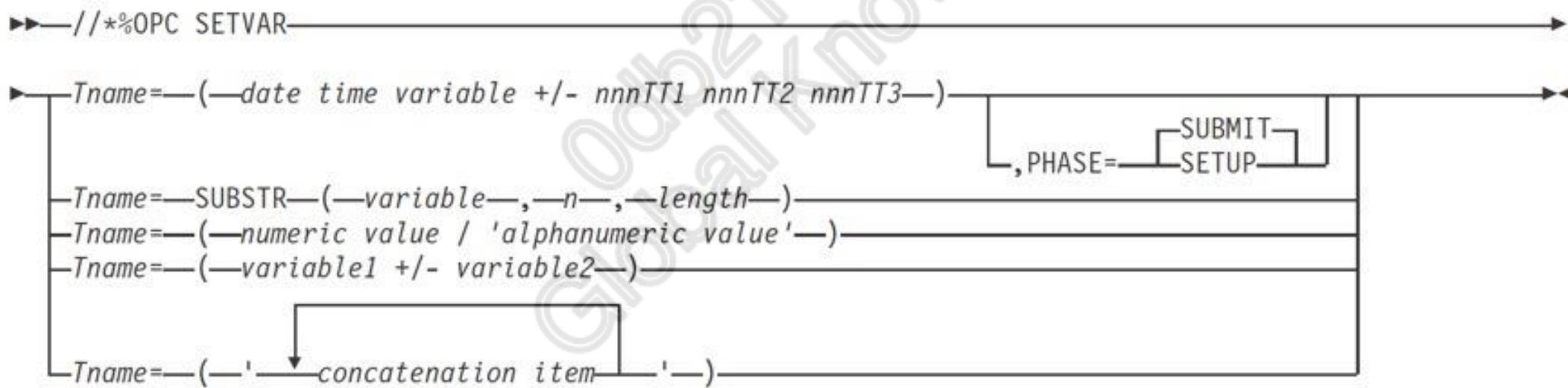
The JCL submitted is

```
/*>OPC SETFORM CDATE=(XX,YY/MM/DDD,TT  
/* CDATE      XX,18/03/070,TT
```

SETVAR Directive

This directive creates a temporary variable by using one of the following values:

- An arithmetic expression together with supplied date or time variables.
- A substring of another variable.
- The result of an arithmetic addition or subtraction.
- Concatenated strings or variables set to an alphanumeric value.



SETVAR Sample

BEFORE with Date of occurrence 11/03/2018

```
/*%OPC SETFORM OCDATE=(YYMMDD)
/*%OPC SETVAR TVAR1=(&OCDATE')
/*%OPC SETVAR TVAR2=SUBSTR(&TVAR1,3,2)
/*%OPC SETVAR TVAR3=(OCDATE + 1MO)
/*%OPC SETVAR TVAR4=SUBSTR(&TVAR3,3,2)
//RETWT1 EXEC PGM=IEFBR14
/*** TVAR1 = &TVAR1 TVAR2 = &TVAR2 TVAR3 = &TVAR3 TVAR4 = &TVAR4
```

AFTER

```
/*>OPC SETFORM OCDATE=(YYMMDD)
/*>OPC SETVAR TVAR1='180311'
/*>OPC SETVAR TVAR2=SUBSTR(180311,3,2)
/*>OPC SETVAR TVAR3=(OCDATE + 1MO)
/*>OPC SETVAR TVAR4=SUBSTR(180411,3,2)
//RETWT1 EXEC PGM=IEFBR14
/*** TVAR1=180311 TVAR2=03 TVAR3=180411 TVAR4=04
```

TABLE Directive

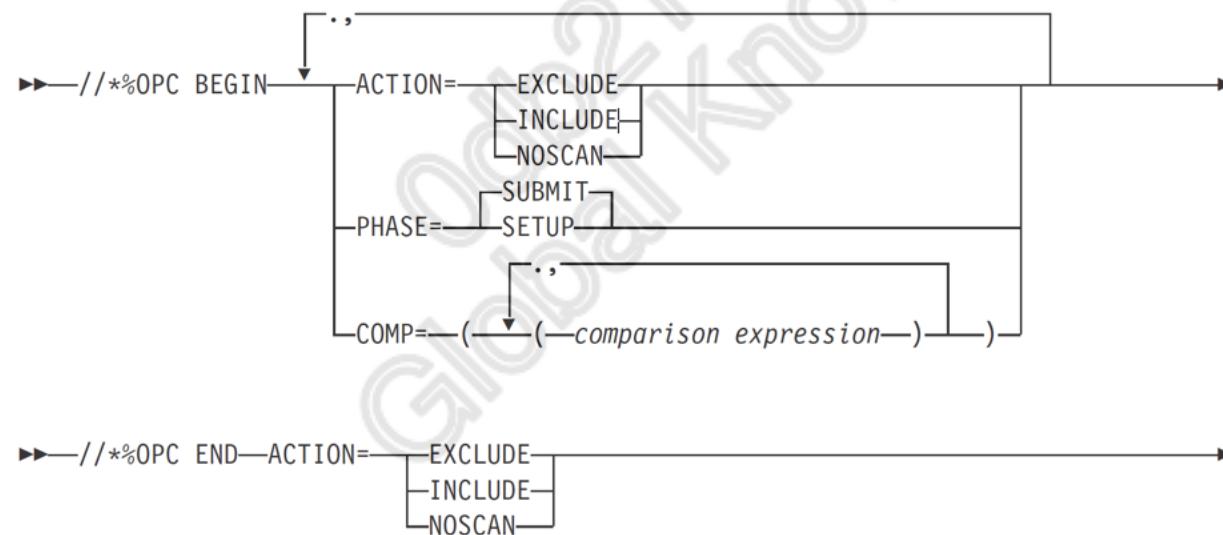
This directive defines a variable table that will be searched before the variable tables in any existing concatenation when resolving JCL variables

```
►--///*%OPC TABLE—NAME=—(table name)————→
```

BEGIN and END Directive

These directives, used in pairs, denote the following, depending on the value of the ACTION keyword:

- The start and end of variable substitution
- The start and end of the lines to be included in the tailored job
- The start and end of the lines to be excluded from the tailored job



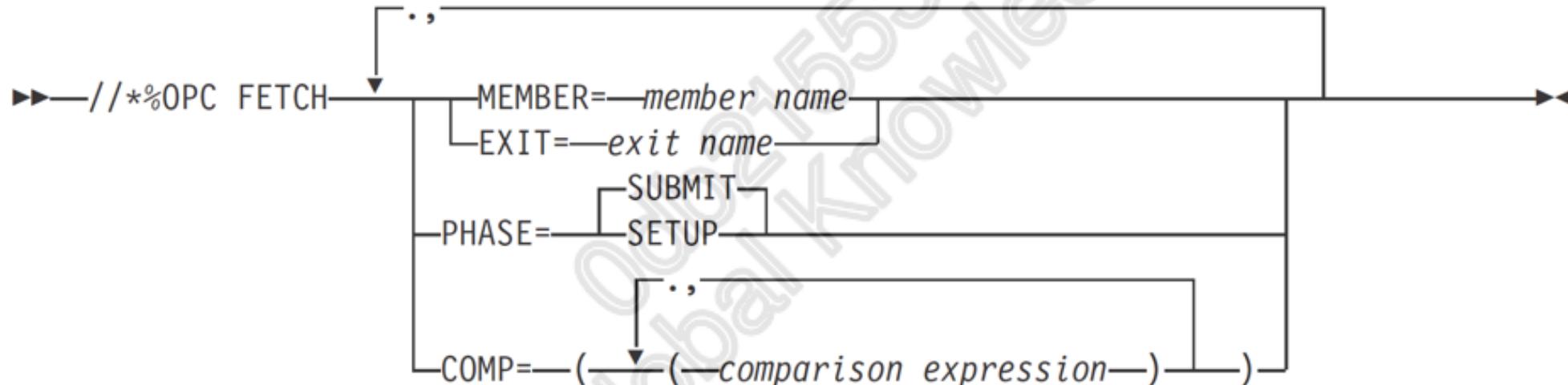
BEGIN and END Directive

```
///*%OPC SCAN
//&OJOBNAME JOB SETD,TWS930,NOTIFY=TWS930,CLASS=A,MSGCLASS=H
///*%OPC BEGIN PHASE=SUBMIT,ACTION=INCLUDE
//STEP1 EXEC PGM=IEFBR14
///*%OPC BEGIN ACTION=NOSCAN
// SET ENV=TWS930
// SET TYPE=_CNTL
//DDNAME1 DD DISP=SHR,DSN=&ENV..&TYPE
//SYSOUT DD SYSOUT=A
///*%OPC END ACTION=NOSCAN
///*%OPC END ACTION=INCLUDE
```

The variable &ENV and &TYPE are not updated by IWS

FETCH Directive

This directive lets you include lines, fetched from a partitioned data set member or supplied by an exit, in your job



FETCH Directive

The JCL before submit

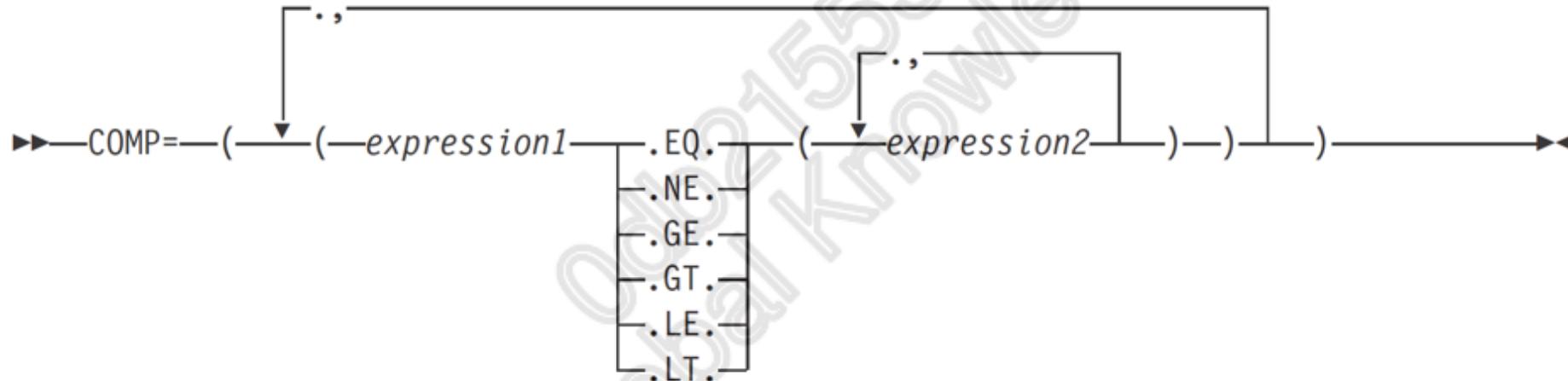
```
//*%OPC SCAN  
//&JOBNAME JOB SETD,TWS930,NOTIFY=TWS930,CLASS=A,MSGCLASS=H  
//*%OPC FETCH MEMBER=PRE
```

The JCL after submit

```
/*>OPC SCAN  
//VARFETCH JOB SETD,TWS930,NOTIFY=TWS930,CLASS=A,MSGCLASS=H  
/*>OPC FETCH MEMBER=PRE  
*****  
/* *** MEMBER PRE  
*****  
//STEP1 EXEC PGM=IEFBR14
```

COMP Directive

A comparison expression lets you specify conditions when BEGIN and FETCH directives will be honored



COMP Directive

The JCL before submit

```
/*%OPC SCAN
//&OJOBNAME JOB SETD,TWS930,NOTIFY=TWS930,CLASS=A,MSGCLASS=H
/*%OPC FETCH MEMBER=WEEK,COMP=(&CDAY..NE.(6,7))
/*%OPC FETCH MEMBER=WEEKEND,COMP=(&CDAY..EQ.(6,7))
```

The JCL after submit

```
/*>OPC SCAN
//VARCOMP JOB SETD,TWS930,NOTIFY=TWS930,CLASS=A,MSGCLASS=H
/*>OPC FETCH MEMBER=WEEK,COMP=(7.NE.(6,7))
/*>OPC FETCH MEMBER=WEEKEND,COMP=(7.EQ.(6,7))
*****
/* *** MEMBER WEEKEND
*****
//STEP1 EXEC PGM=IEFBR14
```

WAPL

Workload Automation Programming Language

Odb215518
Global Knowledge

Unit objectives : 9.3.0 WAPL

- SOE is stopped and replace by WAPL
- WAPL is supported by the LAB
- Manual of WAPL is 358 Pages

Section 1

The software changes

Odb2155318
Global Knowledge

Tools back in the bad ol' days

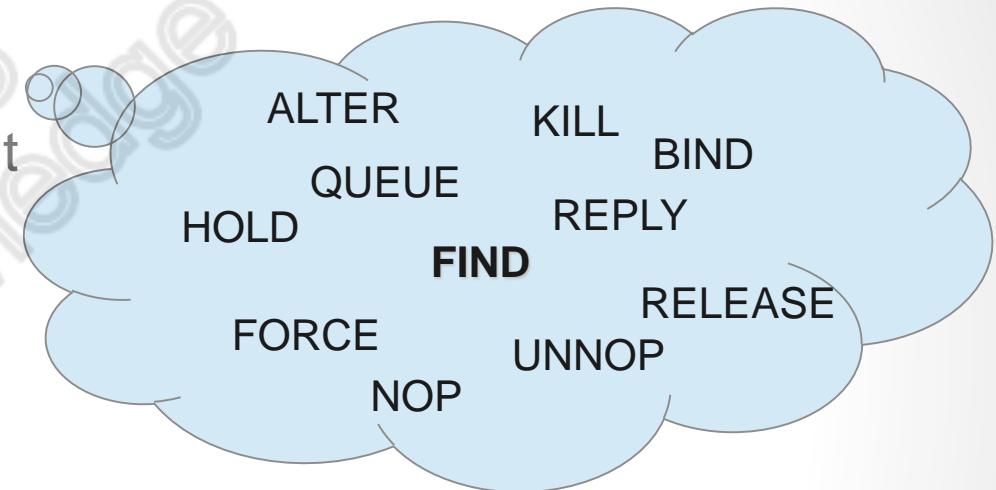
- Before 2009 your only choice was LISTSTAT (in BCIT)
 - You scheduled a job at a predefined time
 - It would issue a return code specific to that status
- From 2009 SOE had a version of LISTSTAT
 - You could choose GOOD and BAD status and only fail for bad ones
 - You still had to pick ONE job to check
- From 2011 SOE 3.1 added a new trick
 - LISTSTAT could issue SRSTAT for selected status values
 - This allowed some degree of conditional reaction to the state of your workload

What a difference a year makes

- From SOE 3.3 to WAPL 9.3
- Current Plan operation commands
 - Better search criteria to identify jobs
 - Capability to select more than one
- Object variables
 - Access to detailed information about every object in the plan
- Programming language
- Ability to make decisions within the code
- Multiple actions can be taken at a single checkpoint
- Communication tools
 - Send messages by eMail or console
 - Send commands to the console

Current Plan Operation commands

- A new set of commands to update the Current Plan
- All use the same keywords to find the jobs you want
- The FIND command is a special case
 - It finds jobs to store in a list
 - Performs no actions
 - Subsequent code can action the list



How to find a job (in the plan)

- Identification keywords
 - All the LIST CPOPCOM arguments, e.g. ADID, IA, OPNO, JOBNAME etc
 - Specify as many of these as you can for efficiency of execution
 - STATUS can be a list of values
- Filter keywords
 - RANGE allows selection within IA range from and/or to
 - COUNT says how many to select, default=1, all=0, negative numbers drop end of list
 - POSITION decides which end of the list to start counting EARLIEST or LATEST

Doing many things to many jobs

- Do you want to run multiple commands against a common set of jobs ?
 - Issuing multiple commands each with the same arguments might not work
 - One action might alter a job such that the next command won't find it
 - Beside that, why search more than once
 - This is where SAVELIST comes in
- Run the FIND command with arguments to find the job and SAVELIST
- Run all subsequent commands with USELIST as the only argument
- You can also create an OBJECT variable to hold the list

Object variables

- A new, rather complex, style of variable
 - Can be used to store contents of files
 - Can be used to store multi-dimensional representations of ANY object
 - Generated from LIST, SELECT, READ and current plan operation commands
 - Use OBJECT keyword to name the variable
 - Information then available as variables with an !@ prefix

There's even a command to help with objects

```
04/22 19.52.52 EQQI200I SHOW OBJECT(AD)
04/22 19.52.52 EQQI601A Object: @OBJ-ADKEY
04/22 19.52.52 EQQI601A Object: @OBJ-ADIA
04/22 19.52.52 EQQI601A Object: @OBJ-ADDL
04/22 19.52.52 EQQI601A Object: @OBJ-ADJCLVTAB
04/22 19.52.52 EQQI601A Object: @OBJ-ADSUFFIX
04/22 19.52.52 EQQI601A Object: @OBJ-ADID
04/22 19.52.52 EQQI601A Object: @OBJ-ADSTAT
04/22 19.52.52 EQQI601A Object: @OBJ-ADTO
04/22 19.52.52 EQQI601A Object: @OBJ-ADTYPE
04/22 19.52.52 EQQI601A Object: @OBJ-ADMONITOR
04/22 19.52.52 EQQI601A Object: @OBJ-ADFROM
04/22 19.52.52 EQQI601A Object: @OBJ-ADDESC
04/22 19.52.52 EQQI601A Object: @OBJ-ADGROUP
04/22 19.52.52 EQQI601A Object: @OBJ-ADOWNER
04/22 19.52.52 EQQI601A Object: @OBJ-ADODESC
04/22 19.52.52 EQQI601A Object: @OBJ-ADPRIOR
04/22 19.52.52 EQQI601A Object: @OBJ-ADCAL
04/22 19.52.52 EQQI601A Object: @OBJ-ADLDATE
04/22 19.52.52 EQQI601A Object: @OBJ-ADLTIME
04/22 19.52.52 EQQI601A Object: @OBJ-ADLUSER
04/22 19.52.52 EQQI601A Object: @OBJ-ADCOMVERS
04/22 19.52.52 EQQI601A Object: @OBJ-ADGROUPID
04/22 19.52.52 EQQI601A Object: @OBJ-ADLUTS
04/22 19.52.52 EQQI601A Object: @OBJ-ADDSM
04/22 19.52.52 EQQI601A Object: @OBJ-ADSM
```

```
04/22 19.52.52 EQQI601A Object: @OBJ-#ADRUN
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRSEQ
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRPER
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRVALF
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRVALT
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRUNDESC
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRURULE
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRTYPE
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRIAD
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRIAT
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRDD
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRDT
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRUNVERS
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRJVTAB
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRSHTYPE
04/22 19.52.52 EQQI601A Object: @OBJ-ADRUN-n-ADRINPOS
```

It's now a programming language

- IF/THEN/ELSE processing now available
 - Checkpoints can now react to LISTSTAT result
 - Instead of LISTSTAT the object can be interrogated and handle different outcomes
- DO/UNTIL/WHILE processing now available
 - Multiple “things” can be handled at once
- All REXX functions accessible in the code
 - Not emulated this is the actual REXX function exposed under the covers
 - Brings a richness of functionality to what can be done

Go tell it to the Developers, Ops & Managers

- TWS can finally natively communicate with the outside world
 - The new SENDMAIL command uses SMTP to send emails direct from WAPL
 - The text can include variables
 - The text can include files for variable length content
- A similar SENDMSG command can issue console messages
 - These can be used to trigger automation
- The new CONSOLE command allows commands to be issued to z/OS or JES
 - Will detect a bad command
 - But does not get any other feedback
 - Variables can be used in the commands

Section 2

Putting it to real use

Odb2155318
Global Knowledge

WAPL Examples discussed

1. LISTSTAT

- Take action on a job based on the status or RC (Success, Error, Waiting, other..)
- Using IF/THEN logic to perform multiple actions (SENDMAIL, ADD, OPSTAT, etc.)

2. FIND

- Finding and listing operation details from the Current Plan
- Setting FIND to OBJECTS to provide a lot of detail
- Using IF/THEN logic to perform multiple actions (SENDMAIL, ALTER, NOP, etc.)

3. Dynamic submission

- Using ADD function instead of traditional EVERY RUN CYCLE (FROM, EVERY, UNTIL)
- Allows easier control over repeat applications with very short intervals
- Ensures continuous processing when a job is in error. No need for operator intervention
- Less jobs in WAIT in the Current Plan

WAPL Examples cont..

4. JSSTART

- Pre-stage JCL in JS repository for processes that use the same job name
- No need for program change

5. Perform Utility functions

- Run daily backups of IWS databases
- Can be used for both unload and load

6. Global Flexibility

- The ability to code based on defined variables in the USER FIELD of multiple jobs
- Extract\Provide high level detail
- Universal generic code can keep WAPL library light
- If running batch mode, may need additional members depending on volume and/or WLM. Tailor accordingly for performance

LISTSTAT

- Want to check if a job is on time? If not, send an alert to those who need to know
- Want to know immediately when a job fails? Have an email generated and ALTER status to C
- Want to check if a job is running longer than it should? KILL the operation send email alert. Then ALTER the status to C

It all begins with the LISTSTAT:

```
LISTSTAT: LISTSTAT CPOP ADID(TESTAPP1118X) IA(!OYMD1.!OHHMM.)  
WSNAME(W705) OPNO(005)  
JOBNAME(OPER0001) POLICY(E=99,S=08,W=02,CX=00,4)
```

Examples based on single operation w/Conditional Dependency

Application : TESTAPP1118X WAPL Presentation app							
Row Cmd	Oper WS no.	Duration HH.MM.SS	Job name	Internal predecessors	Morepreds -IntExt-	No.of Conds	
NONR 001	001	00.00.01	START	-----	0 0	0	
W705 005	005	00.03.04	OPER0001	001 -----	0 0	0	
SOEB 010	010	00.00.04	TSMYWAPL	001 -----	0 0	1	
NONR 255	255	00.00.01	END	005 -----	0 0	0	

“My job needs to be active by 10am. Please let me know if it’s not running”

- Job scheduled for 10am but is being held by a Special Resource
- WAPL job to check status has set time of 10.01am

View: Full (EQQMOPLT)										Row 1 of 4			>>
Row Operation				High Level Monitoring									
cmd	Jobname	SXU	WS	No.	Application ID	Input	Arrival	Job ID	UC	HN	ST		
	START	C N	NONR	001	TESTAPP1118X	11/10/15	10.00		N	NN	YY		
	OPER0001	*XN	W705	005	TESTAPP1118X	11/10/15	10.00		YN	NN	YN		
	TSMYWAPL	W N	SOEB	010	TESTAPP1118X	11/10/15	10.01		YN	NN	YN		
	END	W N	NONR	255	TESTAPP1118X	11/10/15	10.00		YN	NN	YN		
***** end of data *****													

- At 10.01 the LISTSTAT returns a code 4 based on the POLICY
- The IF/THEN DO statement for a code 4 executes

```
IF @CMD(LISTSTAT.EQ.04) THEN DO  
  SENDMAIL FROM(<WAPL@alert.com>) TO(USER1@EMAIL.COM)  
  SUBJECT(JOB OPER0001)  
    TXT(Hello USER, it's 10am and your job is not running yet)  
END
```

Email to USER1 is generated



From: WAPL@alert.com
To: USER1@EMAIL.COM
Cc:
Subject: JOB OPER0001

Hello USER, it's 10am and your job is not running yet

“If my job fails send me an email. No further action is needed on the job”

- Job fails at 10am. WAPL job has Conditional Dependency to run when OPER0001 has a status of E

View: Full (EQQMOPLT)										Row 1 of 4			>>		
Row Operation				----- High Level Monitoring -----											
cmd	Jobname	SXU	WS	No.	Application ID	Input	Arrival	Job ID	UC	HN	ST				
	START	C	N	NONR	001	TESTAPP1118X	11/10/15	10.00					N	NN	YY
	OPER0001	E	N	W705	005	TESTAPP1118X	11/10/15	10.00	1560				YN	NN	YN
	TSMYWAPL	W	N	SOEB	010	TESTAPP1118X	11/10/15	10.00					YN	NN	YN
	END		W	N	NONR	255	TESTAPP1118X	11/10/15	10.00				YN	NN	YN
***** end of data *****															

E status triggers WAPL status job and returns a 99 based on the POLICY

```
View: Full (EQQMOPLT)                                Row 1 of 4 >>
Row Operation      ----- High Level Monitoring -----
cmd Jobname  SXU  WS    No. Application ID   Input Arrival Job ID  UC HN ST
____ START      C N  NONR 001 TESTAPP1118X     11/10/15 10.00      N NN YY
____ OPER0001   C N  W705 005 TESTAPP1118X     11/10/15 10.00  1560      N NN YN
____ TSMYWAPL  SSN  SOEB 010 TESTAPP1118X     11/10/15 10.00 JOB15509 YN NN YN
____ END        C N  NONR 255 TESTAPP1118X     11/10/15 10.00      N NN YN
***** end of data *****
```

- IF/THEN performs SENDMAIL, ALTERS E status to C

```
IF @CMD(LISTSTAT.EQ.99) THEN DO
  SENDMAIL FROM(<WAPL@alert.com>) TO(USER1@EMAIL.COM)
  SUBJECT(JOB OPER0001 FAILED)
  TXT(Hello USER, OPER0001 is in error. Error will be cleared)
  ALTER JOBNAME(OPER0001) STATUS(E) NEW_STATUS(C)
END
```

USER notified via email

From: ■ WAPL@alert.com
To: ■ USER1@EMAIL.COM
Cc:
Subject: JOB OPER0001 FAILED

Hello USER, OPER0001 is in error. Error will be cleared

- “My job runs 5 minutes. If it runs for 30 there is a big issue. Cancel the job immediately and send me an email”
- Job starts 10am. WAPL check has set time for 10.30am

```

View: Full (EQQMOPLT)                                Row 1 of 4          >>
Row Operation      ----- High Level Monitoring -----
cmd  Jobname   SXU   WS    No. Application ID  Input Arrival  Job ID  UC HN ST
-----           -----
START            C N  NONR  001 TESTAPP1118X  11/10/15 10.00      N NN YY
OPER0001         SSN  W705  005 TESTAPP1118X  11/10/15 10.00  4452  YN NN YN
TSMYWAPL        *HN  SOEB  010 TESTAPP1118X  11/10/15 10.30  YN YN YN
END              W N  NONR  255 TESTAPP1118X  11/10/15 10.00  YN NN YN
***** end of data *****

```

- LISTSTAT will return code 8 based on the POLICY
- KILL cmd issued, ALTER to C status, email sent to USER1

```
IF @CMD(LISTSTAT.EQ.08) THEN DO
  KILL ADID(TESTAPP1118) JOBNAME(OPER0001) OPNO(005) STATUS(S)
    WAIT 00.00.07
  ALTER JOBNAME(OPER0001) STATUS(E) NEW_STATUS(C)
  SENDMAIL FROM(<WAPL@alert.com>) TO(USER1@EMAIL.COM)
    SUBJECT(JOB OPER0001 Exceeded Time)
    TXT(Hello USER, OPER0001 ran for over 30 minutes.)
    TXT(The job was cancelled and cleared from the queue)
END
```

- WAPL job runs at 10.30am

```

View: Full (EQQMOPLT)                                Row 1 of 4 >>
Row Operation      ----- High Level Monitoring -----
cmd Jobname SXU WS No. Application ID Input Arrival Job ID UC HN ST
    START   C N NONR 001 TESTAPP1118X 11/10/15 10.00      N NN YY
    OPER0001 SSN W705 005 TESTAPP1118X 11/10/15 10.00 4452 YN NN YN
    TSMYWAPL SUN SOEB 010 TESTAPP1118X 11/10/15 10.30 YN YN YN
    END     W N NONR 255 TESTAPP1118X 11/10/15 10.00 YN NN YN
***** end of data *****

```

- KILL is issued

```

View: Full (EQQMOPLT)                                Row 1 of 4 >>
Row Operation      ----- High Level Monitoring -----
cmd Jobname SXU WS No. Application ID Input Arrival Job ID UC HN ST
    START   C N NONR 001 TESTAPP1118X 11/10/15 10.00      N NN YY
    OPER0001 E N W705 005 TESTAPP1118X 11/10/15 10.00 4452 YN NN YN
    TSMYWAPL SSN SOEB 010 TESTAPP1118X 11/10/15 10.30 JOB16054 YN YN YN
    END     W N NONR 255 TESTAPP1118X 11/10/15 10.00 YN NN YN
***** end of data *****

```

- ALTER to C

View: Full (EQQMOPLT)										Row 1 of 4			>>		
Row Operation				----- High Level Monitoring -----											
cmd	Jobname	SXU	WS	No.	Application ID	Input	Arrival	Job ID	UC	HN	ST				
START	C N	NONR	001	TESTAPP1118X		11/10/15	10.00		N	NN	YY				
OPER0001	C N	W705	005	TESTAPP1118X		11/10/15	10.00	4452	N	NN	YN				
TSMYWAPL	C N	SOEB	010	TESTAPP1118X		11/10/15	10.30	JOB16054	N	YN	YN				
END	C N	NONR	255	TESTAPP1118X		11/10/15	10.00		N	NN	YN				
***** end of data *****															

Email sent to USER1



From: ■ WAPL@alert.com
 To: ■ USER1@EMAIL.COM
 Cc:
 Subject: JOB OPER0001 Exceeded Time

Hello USER, OPER0001 ran for over 30 minutes.
 The job was cancelled and cleared from the queue

FIND

- I need to find all the TS* jobs in the current plan that are in W status
- I would like the list to be sent to me via email
- All I need is the job name, workstation, and status value
- Once the list is generated and emailed, you can NOP all the jobs

Create temp WORKFILE to store data and to pass to email body

Provide the TS* value in the JOBNAME field along with W status

OBJECT variables set for data needed in report

Items written to SAVELIST to be used for NOP command

Email generated with WORKFILE

NOP command issued against USELIST

```
//WORKFILE DD DISP=(NEW,DELETE),DSN=&&WORK,  
//      RECFM=FB,LRECL=80  
//SYSIN  DD DATA,DLM=##  
//SYSIN  DD *  
OPTIONS TRACKERS(TWCP.*.TWTP) MSGLEVEL(5) MAILSERVER(ISO.COM)  
VARSUB SCAN  
FINDIT: FIND JOBNAME(TS*) STATUS(W) DATE(151110)  
WSNAME(ISOP)  
OBJECT(JOB) COUNT(0) SAVELIST(TSWLIST)  
IF @CMD(FINDIT.EQ.0) THEN DO  
  VARSET MAX = WORDS("!@JOBS-@FILTER")  
  DO IJOB = 1 TO !MAX  
    VARSET OJOB = WORD("!@JOBS-@FILTER", "!JOB.")  
    VARSET JOB  = @V(@JOBS!OJOB.-CPOPJBN)  
    VARSET WS   = @V(@JOBS!OJOB.-CPOWSN)  
    VARSET ST   = @V(@JOBS!OJOB.-CPOPST)  
    WRITE WORKFILE "!WS.  !JOB.  !ST."  
  END  
  CLOSE WORKFILE  
  IF @V(ST) = "W" THEN DO  
    SENDMAIL FROM(WAPL@alert.com) TO(USER1@EMAIL.COM)  
      SUBJECT(TS* jobs in W status)  
      TXT(The following TS* jobs are in W status. All jobs will be NOP)  
      TXT()  
      TEXTDD(WORKFILE)  
      NOP USELIST(TSWLIST)  
  END
```

- FIND is obtaining the details from the CP

```

EQQI200I FINDIT: FIND JOBNAME(TS*) STATUS(W) DATE(151116)
EQQI200I WSNAME(ISOP) OBJECT(JOBS) COUNT(0)
EQQI161A SELECTED: COESTBLH#002 1511160330 ISOP_6 TSTKBKUP W
EQQI161A SELECTED: TSACMTST#001 1511160800 ISOP_3 TSACMTST W
EQQI161A SELECTED: TSCATBKC#001 1511160800 ISOP_3 TSCATBKC W
EQQI161A SELECTED: TSCATBKC#001 1511160800 ISOP_6 TSCATMSC W
EQQI161A SELECTED: TSCATBKG#001 1511160800 ISOP_3 TSCATBKG W
EQQI161A SELECTED: TSCATBKG#001 1511160800 ISOP_6 TSCATUNI W
EQQI161A SELECTED: TSCATBKG#001 1511160800 ISOP_9 TSCATMSG W
EQQI161A SELECTED: TSCATBKN#001 1511160800 ISOP_3 TSCATBKN W
EQQI161A SELECTED: TSCATBKN#001 1511160800 ISOP_6 TSCATMSN W
EQQI161A SELECTED: TSCTTRIG#002 1511160620 ISOP_6 TSCATD59 W
EQQI161A SELECTED: TSCTTRIG#002 1511160620 ISOP_9 TSCATE32 W
EQQI161A SELECTED: TSCTTRIG#002 1511160620 ISOP_12 TSCATE57 W
EQQI161A SELECTED: TSCTTRIG#002 1511160620 ISOP_15 TSMFCATR W
EQQI161A SELECTED: TSDYVMA1#001 1511160500 ISOP_3 TSDYVMA1 W
EQQI161A SELECTED: TSDYVMA2#001 1511160500 ISOP_3 TSDYVMA2 W
EQQI161A SELECTED: TSDYVMP1#001 1511160500 ISOP_3 TSDYVMP1 W
EQQI161A SELECTED: TSDYVMP2#001 1511160500 ISOP_3 TSDYVMP2 W
EQQI161A SELECTED: TSDYVMT1#001 1511160510 ISOP_3 TSDYVMT1 W
EQQI161A SELECTED: TSDYVMT2#001 1511160515 ISOP_3 TSDYVMT2 W
EQQI161A SELECTED: TSDYVMU1#001 1511160500 ISOP_3 TSDYVMU1 W
EQQI161A SELECTED: TSDYVPA1#007 1511160515 ISOP_3 TSDYVPA1 W
EQQI161A SELECTED: TSDYVPA2#007 1511160515 ISOP_3 TSDYVPA2 W
EQQI161A SELECTED: TSDYVPP1#007 1511160515 ISOP_3 TSDYVPP1 W
EQQI161A SELECTED: TSDYVPP2#007 1511160515 ISOP_3 TSDYVPP2 W

```

USER1 generated email with TS* jobs in W status and workstation

From: WAPL@alert.com
To: USER1@EMAIL.COM
Cc:
Subject: TS* jobs in W status

The following TS* jobs are in W status. All jobs will be NOP

ISOP	TSTKBKUP	W
ISOP	TSACMTST	W
ISOP	TSCATBKC	W
ISOP	TSCATMSC	W
ISOP	TSCATBKG	W
ISOP	TSCATUNI	W
ISOP	TSCATMSG	W
ISOP	TSCATBKN	W
ISOP	TSCATMSN	W
ISOP	TSCATD59	W
ISOP	TSCATE32	W
ISOP	TSCATE57	W
ISOP	TSMFCATR	W
ISOP	TSDYVMA1	W
ISOP	TSDYVMA2	W
ISOP	TSDYVMP1	W
ISOP	TSDYVMP2	W
ISOP	TSDYVMT1	W
ISOP	TSDYVMT2	W
ISOP	TSDYVMU1	W
ISOP	TSDYVPA1	W
ISOP	TSDYVPA2	W
ISOP	TSDYVPP1	W



WAPL@alert.com

Dynamic Submission

- Instead of using the EVERY option when creating a RUN CYCLE, use WAPL ADD command with FROM, EVERY, and UNTIL
- Using ADD in conjunction with LISTSTAT creates a self managing application. Jobs that end in error can have action taken so there is no downtime
- Allows for easier maintenance if process needs to be held or changed
- Less jobs in waiting in the CP throughout the day

- Run Cycle has IA Time of 13.10, then WAPL takes care of the intervals

```
//SYSIN DD *
ADD (!OẠDID.) FROM(0030) UNTIL(2200) EVERY(0010)
/*
```

- First run completes, WAPL dynamically adds next interval 10 minutes later

View: Full (EQQMOPLT)										Row 1 of 8			>>
Row	Operation	High Level Monitoring								UC	HN	ST	
		Cmd	Jobname	SXU	WS	No.	Application ID	Input Arrival	Job ID				
	START	C	N	NONR	001	TESTAPP1118X	11/10/15 13.10		N	NN	YY		
	OPER0001	C	N	W705	005	TESTAPP1118X	11/10/15 13.10	4280	N	NN	YN		
	TSMYWAPL	C	N	SOEB	010	TESTAPP1118X	11/10/15 13.10	JOB22738	N	NN	YN		
	END	C	N	NONR	255	TESTAPP1118X	11/10/15 13.10		N	NN	YN		
	START	AHN		NONR	001	TESTAPP1118X	11/10/15 13.20		N	YN	YY		
	OPER0001	W	N	W705	005	TESTAPP1118X	11/10/15 13.20		N	NN	YN		
	TSMYWAPL	W	N	SOEB	010	TESTAPP1118X	11/10/15 13.20		N	NN	YN		
	END	W	N	NONR	255	TESTAPP1118X	11/10/15 13.20		N	NN	YN		
***** end of data *****													

JSSTART

Users submit report requests via online interface where multiple queries can be made. Each query generates JCL in a single member and ETT sends the job to create the report to current plan. This occurs for each query.

To prevent performance issues for online users the jobs have a Special Resource that only allows them to run at night. Without the ability to prestage the JCL with JSSTART, all the jobs that are triggered into the queue will all have the same JCL from the latest query.

- Make sure EQQJSPDS is indicated and that the MEMBER exists

```
//RUNPIF EXEC EQQYXJCL,          /* SOE JCL PROCEDURE */
//      SUBSYS=&SSNM.           /* GLOBAL TABLE    */
//EQQJSPDS DD DISP=SHR,DSN=ISOCP.PROD.TIVTWS.TWCP.JOBLIB
//SYSIN  DD DATA,DLM=XX
OPTIONS ACTION(REPLACE) MSGLEVEL(5) TRACE(0)
JSSTART ADID(TESTAPP1118X) JOBNAME(OPER0001) STATUS(W) OPNO(015)
IA(&OYMD1.&OHHMM.) MEMBER(OPER0002)
XX
```

- Make sure the JSSTART runs as soon as the application is triggered

Row cmd	Operation ws	no.	text	Jobname	PS	Duration HH.MM.SS	Opt S T	Dep S/P	Res S R1 R2	Stat N Cu
'''	NONR	001		START	1	00.00.01	Y Y		0 0	A
'''	SOEB	005		TSMYWAPL	1	00.02.45	Y N	C	0 0	W
'''	ISOP	010		OPER0001	1	00.00.04	Y N		0 0	W
'''	NONR	255		END	1	00.00.01	Y N		0 0	W
***** Bottom of data *****										

View: Full (EQQMOPLT)									Row 1 of 4			>>
----- High Level Monitoring -----												
cmd	Jobname	SXU	WS	No.	Application ID	Input	Arrival	Job ID	UC	HN	ST	
START		C	N	NONR	001 TESTAPP1118X		11/10/15	10.00		N	NN	YY
TSMYWAPL		C	N	SOEB	005 TESTAPP1118X		11/10/15	10.00	JOB19735	N	NN	YN
OPER0001	RXN			ISOP	010 TESTAPP1118X		11/10/15	10.00		N	NN	YN
END		C	N	NONR	255 TESTAPP1118X		11/10/15	10.00		N	NN	YN

***** end of data *****

- JSSTART runs as soon as the application is triggered and inserts the JCL from defined member in EQQJSPDS. OPER0001 is waiting for Special Resource to run at night but with prestaged JCL

```

Application : TESTAPP1118X          WAPL Presentation app
Operation   : ISOP 10
Status of operation : Ready
Jobname    : OPER0001                JCL last updated by: TWTP

*****
000001 //OPER0002 JOB (1975,74,3,I,01975-1812-T19500-XX).
000002 //           SCHEDULING,CLASS=7,MSGCLASS=H,
000003 //           USER=L741975
000004 /*JOBPARM SYSAFF=IS00
000005 /**
000006 //STEP3      EXEC PGM=EQQRETWM,PARM='WT01,RC00'
000007 /*
000008 //

***** Bottom of Data *****

```

HOUSEKEEPING DATABASES

- Use WAPL to backup IWS databases. Puts output in batch loader format for quick reload in the event databases are corrupted or for migration. Use the same proc to load with output file as SYSIN
- WS, ETT, SR, CL, PER, AD, OI, JCLV

```
//OUTDATA DD SYSOUT=*,LRECL=4096
//OUTBL  DD DSN=ISOCP.PROD.TIVTWS.TWCP.SOE.BKUP***+1),
//        UNIT=SYSDA,DISP=(,CATLG),
//        SPACE=(CYL,(100,25),RLSE),
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
/*
//SYSIN  DD *
OPTIONS STRIP(Y) SHOWDFLT(N)
LIST WS WSNAME(*) SELECT(Y)
LIST ETT ETTNAME(*) SELECT(Y)
LIST JCLV JCLVTAB(*) SELECT(Y)
```

Global Flexibility

- The ability to read in variables defined in each individual USER FIELD gives WAPL an extremely high degree of customization. At the same time it allows for less overhead in respect to defined WAPL jobs in your library.
- Creating a handful of “Master JCL” members can be tailored for different scenarios to be used across multiple applications\operations
- The more variables set in the WAPL job, the more USER FIELDS can be coded to substitute.
- Monitor performance and adjust accordingly

```
/*
//SYSIN DD *
OPTIONS TRACKERS(TWCA.*.TWTA) MSGLEVEL(5) MAILSERVER(ISO.COM)
VARSUB SCAN
VARSET APP#CHCK USRF(ADID_TO_CHECK)
VARSET JOB#CHCK USRF(JOBN_TO_CHECK)
VARSET JOB#OPNO USRF(OPNO_TO_CHECK)
VARSET JOB#WSID USRF(WSID_TO_CHECK)
VARSET EMAILIST USRF(ADDR_TO_CHECK)
VARSET JOBDesc USRF(JOB_DESC)
VARSET SERVER USRF(SERVER_NAME)
VARSET OFFSET USRF(TIME_OFFSET)
VARSET SCRIPT USRF(SCRIPT_PATH)
SHOW USRF
```

- Items in yellow represent variables to be passed to the subsequent code.
- Items in white represent the USER FIELD NAME to substitute the data in the UFVALUE to the variable

- **USER FIELDS** tailored with detail specific to each operation

Row	User Field Name	User Field Value	1	2	3	4	5
cmd			-----	-----	-----	-----	-----
1	TIME_OFFSET	60					
2	ADID_TO_CHECK	SMXADDNC#007					
3	JOBN_TO_CHECK	SMXADDNC					
4	JOB_DESC	Add New Customer Script					
5	SERVER_NAME	SNSMTXBA01					
6	SCRIPT_PATH	\DrivingDNA\AAA\AddNewCustomer-3.1_3.0.bat					
7	OPNO_TO_CHECK	003					
8	WSID_TO_CHECK	W702					
9	ADDR_TO_CHECK	i70530@iso.com,i71103@iso.com					
***** Bottom of data *****							

LISTSTAT w/VARSUB

```
LISTSTAT: LISTSTAT CPOP ADID(!APP#CHCK.) IA(!OYMD1.!OHHMM.)
          WSNAME(!JOB#WSID.) OPNO(!JOB#OPNO.)
          JOBNAME(!JOB#CHCK.) POLICY(E=99,S=08,C=00,4)
```

- Template code based on LISTSTAT code 99
- Send email notification, provide detailed email body, change status to C, and dynamically add same application into cp with specified OFFSET IA Time.

```
IF @CMD(LISTSTAT.EQ.99) THEN DO
  SENDMAIL FROM(<WAPL@alert.com>) TO(!EMAILIST.,!EMAILST1.,
  !EMAILST2.)
  SUBJECT(!JOB#CHCK. FAILED and is running on another interval)
  TXT(This is an automated email informing that !JOB#CHCK. )
  TXT('failed. This job runs on server !SERVER. and executes the)
  TXT('the following script)
  TXT( )
  TXT(!SCRIPT.)
  TXT( )
  TXT(Description: !JOBDESC. )
  TXT( )
ALTER ADID(!OADID.) WSNAME(!JOB#WSID.) JOBNAME(!JOB#CHCK.)
  STATUS(E) NEW_STATUS(C)
ADD ADID(!OADID.) IA(+!OFFSET.)
```