IBM Tivoli Workload Scheduler for z/OS

Diagnosis Guide and Reference

Version 9 Release 2

SC32-1261-09

IBM Tivoli Workload Scheduler for z/OS

Diagnosis Guide and Reference

Version 9 Release 2

SC32-1261-09

Note

Before using this information and the product it supports, read the information in “Notices” on page 483.

This edition applies to version 9, release 2, modification level 0 SPE of Tivoli Workload Scheduler for z/OS (program number 5698-T08) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 5. Data areas 83

Figures

Tables

About this publication

Tivoli Workload Scheduler for z/OS*Diagnosis Guide and Reference* contains information you need to diagnose and correct possible problems in using IBM® Tivoli® Workload Scheduler for z/OS®. This publication also describes the structure of an Tivoli Workload Scheduler for z/OS address space and describes the major functions of the subtasks.

This publication addresses these tasks:

- Determining whether the problem was caused by Tivoli Workload Scheduler for z/OS.
- If the problem is in Tivoli Workload Scheduler for z/OS, developing a list of keywords that describe it. This list of keywords will be used by the IBM Support Center to search a database to determine if your problem has previously been reported.
- Isolating failing program code so that a solution can be found.

The term *scheduler*, when used in this publication, refers to Tivoli Workload Scheduler for z/OS. The term DB2®, when used in this publication, refers to DATABASE 2 and DB2 Universal Database™.

The term *z/OS* is used in this publication to mean z/OS and OS/390® operating systems. Where the term *OS/390* appears, the related information applies only to OS/390 operating systems.

What is new in this release

Learn what is new in this release.

For information about the new or changed functions in this release, see *Overview*, section *Summary of enhancements*.

For information about the APARs that this release addresses, see the Tivoli Workload Scheduler for z/OS Program Directory and the Dynamic Workload Console Release Notes at http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27041033.

Who should read this publication

This publication is intended for system programmers who work with Tivoli Workload Scheduler for z/OS and for IBM users responsible for Tivoli Workload Scheduler for z/OS service.

Publications

The Tivoli Workload Automation product is supported by a set of publications.

For a list of publications in the Tivoli Workload Automation product library, see *Publications* under *Reference* in the product documentation.

For a list of terms used in the Tivoli Workload Automation product, see *Glossary* under *Reference* in the product documentation.

Using LookAt to look up message explanations

Look up explanations for most of the IBM messages you encounter, as well as for some system abends (an abnormal end of a task) and codes.

LookAt is an online facility that lets you look up explanations for most of the IBM messages you encounter, as well as for some system abends (an abnormal end of a task) and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM®, VSE/ESA, and Clusters for AIX® and Linux:

- The Internet. You can access IBM message explanations directly from the LookAt website at http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/.
- Your z/OS TSO/E host system. You can install code on your z/OS system to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX System Services running OMVS).
- Your Microsoft Windows workstation. You can install code to access IBM message explanations on the *IBM Online Library z/OS Software Products Collection Kit* (SK3T-4270), using LookAt from a Microsoft Windows DOS command line.
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt website.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your *IBM Online Library z/OS Software Products Collection Kit* (SK3T-4270), or from the LookAt website (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully.

With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For full information with respect to the Dynamic Workload Console, see the Accessibility Appendix in the *IBM Tivoli Workload Scheduler User's Guide and Reference*.

Tivoli technical training

Tivoli provides technical training.

For Tivoli technical training information, refer to the following IBM Tivoli Education website:

http://www.ibm.com/software/tivoli/education

Support information

IBM provides several ways for you to obtain support when you encounter a problem.

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see the appendix on support information in *Tivoli Workload Scheduler: Troubleshooting Guide*.

Conventions used in this publication

Conventions used in this publication.

The publication uses several typeface conventions for special terms and actions. Technical changes to the text are indicated by a vertical line to the left of the change. These conventions have the following meanings:

Information type	**Style convention**	**Example**
Commands	All capital letters	CREATE
References in the text to fields on panels	All capital letters	QUANTITY
File and directory names, input you should type in panel fields	Monospace	`MYAPPLICATION`
First time new term introduced, publication titles	Italics	*Application*

Part 1. Diagnosis guide

This part describes how you classify and describe problems that you encounter when working with Tivoli Workload Scheduler for z/OS systems.

Chapter 1. Introduction

This book is intended to help you classify and describe problems in Tivoli Workload Scheduler for z/OS systems.

To identify an error, you must first gather information related to the problem, such as ABEND codes and dumps. You can then determine whether the problem is in Tivoli Workload Scheduler for z/OS and, if it is, this book helps you to classify and describe the problem. The external symptoms of several problems are described to help you identify which problem type to investigate. Each problem type requires a different procedure when you describe the problem. Use these procedures to build a string of keywords and to obtain documentation relevant to the problem. This combination of a keyword string and associated documentation helps you to describe the problem accurately to IBM Software Support.

Use the Tivoli Workload Scheduler for z/OS problem description sheet in Appendix C, "Problem description sheet," on page 481 as a problem log. You can copy this form and use it to record all the related facts about the problem.

Using keywords to describe a problem

A *keyword* is a word or abbreviation that describes a single aspect of a program failure to IBM Customer Support. You use keywords to describe all aspects of a problem, from the Tivoli Workload Scheduler for z/OS component ID to the area of failure (see Chapter 2, "Initial problem analysis," on page 9). You then use the problem analysis procedures to build a keyword string (see Chapter 3, "Problem analysis procedures," on page 13).

For example, if your program failure is abnormal termination of a task, the keyword is ABEND. Other keywords are also formed to describe particular aspects of the abnormal termination, such as the name of the module where the abend occurred. These keywords are then combined to form a keyword string:

```
5697-WSZ01 ABENDnnn EQQyyyyy REGn
```

where:

5698-A17
The Tivoli Workload Scheduler for z/OS component ID

ABEND
The problem type

nnn The ABEND code (for example, 0C4)

EQQ*yyyyy*
The module containing the ABEND (for example, EQQMAJOR)

REG*n* The general-purpose register containing invalid information

Searching the software-support database

To determine if the problem has been noted before, you can use the keyword string that you create to search the software-support database. If a problem similar to yours is described in the database, a solution is probably available. To widen or narrow the database search, you can vary the keyword string you develop.

If you have the Information/Access IBM licensed program, you can use the keyword string to search the RETAIN® database for solutions to problems similar to yours. IBM Level 1 service personnel can help you develop the keyword string and search the database for a similar problem. If the RETAIN database is not available to you, the IBM Support Center will help you solve your problem.

What is IBM Software Support?

IBM Software Support helps you resolve problems with IBM products, and to ensure that you can make the best use of your IBM computing systems. IBM Software Support is available to all licensed users of IBM licensed programs.

The role of each member of IBM's support staff is outlined in the following sections of this chapter.

Working with the IBM Support Center

To get assistance, you should phone your local IBM Support Center. Before you contact the IBM Support Center, however, try to ensure that the problem belongs with the Support Center. Do not worry if you cannot be sure that the problem is caused by Tivoli Workload Scheduler for z/OS itself. How sure you are will depend on the complexity of your installation, the experience and skill levels of your systems staff, and the symptoms that you have been experiencing.

In practice, many errors reported to the Support Center turn out to be user errors, errors that cannot be reproduced, or errors that should be dealt with by other parts of IBM Service, such as Hardware Customer Engineering or Systems Engineering. This indicates just how difficult it can be to determine the precise cause of a problem. User errors are mainly caused by mistakes in setting up or using systems.

Supplying information to the IBM Support Center

Your first contact with the IBM Support Center will be with the call receipt operator, who will take initial details and put your problem on a queue. You will subsequently be contacted by a Support Center representative who will investigate your problem further.

The Support Center needs to know as much as possible about your problem, so have the information ready before making your first call. It is a good idea to enter the information on a problem description sheet, such as the one shown in Appendix C, “Problem description sheet,” on page 481.

Here are two advantages of using a problem description sheet:

- You will be communicating with the IBM Support Center by telephone. With all your findings before you on a sheet of paper, you will be better prepared to respond to the questions that you might be asked.
- You can maintain your own in-house tracking system to record and document all problems. This information can then be used for planning, organizing, communicating, and establishing priorities for controlling and resolving these problems.

When you contact the Support Center, you will need to give the operator the name of your organization and your access code or customer number. Your access code or customer number is a unique code authorizing you to use IBM Software Services. You must provide this code each time you contact the Support Center.

Using this information, the operator will access your customer profile, which contains your address, relevant contact names, telephone numbers, and details of the IBM products at your installation.

The Support Center operator will ask you if this is a new problem or a call on an existing one. If it is new, you will be assigned a unique incident number. A problem management record (PMR) will be opened on the RETAIN system where all activity associated with your problem will be recorded. The problem will remain *open* until it is resolved. Make a note of the incident number on your own problem reporting sheet. The Support Center will expect you to quote the incident number in all future calls connected with this problem.

If the problem is new to you, the operator will ask you for the source of the problem within your system software—that is, the program that seems to be the cause of the problem. As you are reading this book, it is likely that you have already identified Tivoli Workload Scheduler for z/OS as the problem source. You will also have to give the Tivoli Workload Scheduler for z/OS version, release, and maintenance level.

You will need to give a severity level for the problem. Severity levels can be 1, 2, or 3, and they have the following meanings:

Severity level 1
Indicates that you are unable to use Tivoli Workload Scheduler for z/OS, resulting in a critical condition that needs immediate attention

Severity level 2
Indicates that you are able to use Tivoli Workload Scheduler for z/OS, but that operation is severely restricted

Severity level 3
Indicates that you are able to use Tivoli Workload Scheduler for z/OS with limited functions, but the problem is not critical to your overall operation.

When deciding the severity of the problem, take care neither to understate it nor to overstate it. The Support Center procedures depend on the severity level so that the most suitable use can be made of their skills and resources. Your problem will normally be dealt with immediately if it is severity level 1.

Finally, the call receipt operator will ask you for a brief description of the problem and might prompt you for keywords associated with the problem. The primary keywords are ABEND, ABENDU, DOC, INCORROUT, LOOP, MSG, PERFM, and WAIT, corresponding exactly to the problem classification types used later in this book. Strings containing other keywords are also useful. These are not predefined and might include such items as a message or message number, an abend code, any parameters known to be associated with the problem, or, for example, the name of a feature of Tivoli Workload Scheduler for z/OS. The keywords will subsequently be used as search arguments on the RETAIN database to see if your problem is a known one that has already been the subject of an authorized program analysis report (APAR).

You will not be asked for any more information at this stage. However, you should keep all the information relevant to the problem, including logs, dumps, and traces.

How your problem is processed

How your problem is subsequently processed depends on its nature. The representative who handles the problem will give you guidance on what is required from you.

Details of your call are passed, using the RETAIN problem management system, to the associated support group. Because your problem is one associated with Tivoli Workload Scheduler for z/OS, it is put on a Tivoli Workload Scheduler for z/OS queue. The problems are dealt with in order of severity level.

At first, a Level 1 representative will use the keywords that you have provided to search the RETAIN database. If your problem is found to be one already known to IBM, and a fix has been devised for it, a program temporary fix (PTF) can quickly be dispatched to you. If the RETAIN search is unsuccessful, the problem will be passed to a Level 2 representative, who will ask you for more information about your problem over the telephone.

Let the Level 2 representative know if any of the following events occurred before the problem appeared:
- Changes in level of z/OS or licensed programs
- PTFs applied
- Additional features used
- Application programs changed
- Unusual operator action.

You might be asked to give values from a formatted dump or trace table. You might also be asked to carry out some special activity (for example, to set a trap or to use trace with a certain selectivity) and then to report on the results.

It might be necessary to have several follow-up telephone calls, depending on the complexity of the symptoms and your system environment. In every case, the actions taken by you and the Support Center are entered in the PMR. The Level 2 representative can then be acquainted with the full history of the problem before any follow-up call.

If the Level 2 investigation shows that the problem is already known and a fix has been developed, the fix will be sent to you. If the problem is new, an APAR might be submitted. This will be dealt with by the Tivoli Workload Scheduler for z/OS change team.

Working with the change team

You use APARs to communicate a new problem with a program, in this case Tivoli Workload Scheduler for z/OS, to the change team for the program. When the change team solves the problem, they can, at your request, supply you with an APAR fix, enabling you to get your system running again. Finally, a PTF is produced to replace the module in error, and the APAR is closed.

The APAR process

The first step in the APAR process is that a Level 2 representative will enter your APAR into the RETAIN system. The APAR text will contain a description of your problem. If you have found a means of bypassing the problem, details of this will be entered as well. Your name is also entered so that the change team knows who to contact if they need to ask anything about the APAR documentation and supporting material (for simplicity, referred to as documentation in the rest of this book).

When the APAR has been entered, you will be given an APAR number. You must write this number on all the documentation you submit to the change team. This number will always be associated with the APAR and its resolution. If a code change is required, this number will be associated with the APAR fix as well.

The next stage in the APAR process involves providing the change team with the necessary documentation. Here is a summary of the things you must do:

1. You must collect the documentation that is required for the APAR. You will be given guidance by the Level 2 representative on precisely what you must send. The documentation that is required will vary, depending on the problem area (see "Collecting the documentation for the APAR").
2. You must package all the documentation and send it to the change team. The procedure for this is given in "Sending the documentation to the change team" on page 8.
3. You must apply the PTF resulting from the APAR, possibly after testing the fix on your system. This is described in "Obtaining and applying the fix" on page 8.

Collecting the documentation for the APAR

As a general rule, the documentation you must submit for an APAR includes all the material you need to carry out problem determination. Some of the documentation is common to all Tivoli Workload Scheduler for z/OS problems, and some is specific to particular problems.

Make sure the problem you have described can be seen in the documentation you send. If the problem has ambiguous symptoms, you must reveal the sequence of events leading up to the failure. Tracing is valuable in this respect, but you might be able to provide details that the trace cannot give. You are encouraged to annotate your documentation, if your annotation is legible and if it does not cover up vital information. You can highlight data in any hard copy you send, using transparent highlighting markers. You can also write notes in the margins, preferably using a red pen so that the notes are not overlooked.

Note: If you send too little documentation or if it is unreadable, the change team will have to return the APAR marked 'insufficient documentation'. Therefore, you should prepare your documentation carefully, and send everything that is relevant to the problem.

Here is a list of the documentation you might be asked to submit for an APAR (in this case, documentation also refers to supporting material). However, these are only guidelines; you must find out from your Level 2 representative precisely what documentation you must send for your specific problem.

- Details of the JCL and messages from the jobs that were run when Tivoli Workload Scheduler for z/OS was installed.
- The Tivoli Workload Scheduler for z/OS files that are involved, from both before and after the error. You must supply the contents of the files, as well as their attributes (such as size and block size), and a list of the catalog entries for the VSAM files.
- A storage dump of program load modules.
- A printout of the z/OS SYSLOG message-log file.
- A printout of the Tivoli Workload Scheduler for z/OS message log (EQQMLOG) for each address space.
- A printout of Tivoli Workload Scheduler for z/OS diagnostic files for both the dialog user and the address space. The files are defined by ddname EQQDUMP.

- Information about the route used through the dialog (panel IDs) and the data entered on each panel. A printout of the dialog panels involved with input data is also useful.
- A listing of the job-tracking logs.
- ISPF diagnostic material.
- A description of the Tivoli Workload Scheduler for z/OS configuration.
- A description of the Tivoli Workload Scheduler for z/OS maintenance level—PTFs and APARs applied. System Modification Program/Extended (SMP/E) provides this information.
- Details about your z/OS and JES levels.
- Details of any user modifications.
- The kind of interface the customer uses (Dynamic Workload Console, PIF).

You can use the Tivoli Workload Scheduler for z/OS APAR tape function to collect all important Tivoli Workload Scheduler for z/OS data sets. You might need to modify the JCL that is generated by the APAR tape function so that all event data sets are collected.

Sending the documentation to the change team

The best way to submit documentation to the Support Center is via a communication link. If this is not possible, you can send the documentation in an APAR box, which you can obtain from your local IBM branch office. APAR boxes are clearly marked as such, and they have a panel where you can write tracking information, such as the APAR number.

Place all your documentation and notes in one or more APAR boxes. If you need to use more than one box, make sure that each box is clearly marked. If you include any magnetic tapes, write this clearly on the outside of the box. This will lessen the chance that the tapes will knowingly be stored in magnetic fields strong enough to damage the data. You will also need a mailing label with the address of the Tivoli Workload Scheduler for z/OS change team on it.

When the change team receives the package, they will note this in your APAR record on the RETAIN system. The team will then investigate the problem and occasionally ask you to send more documentation.

Obtaining and applying the fix

When the problem is solved, a code will be entered on RETAIN to close the APAR, and you might be provided with an APAR fix. You can ask your Support Center about the progress of your APAR at any time, particularly if it is a problem of high severity. When the change team has found a fix for your problem, they might want you to test it on your system. If they do ask you to test the fix, you are normally given 2 weeks to do the test and to provide them with feedback. However, you can ask for an extension if you are unable to complete the testing in that time.

When the team is confident that the fix is satisfactory, the APAR is closed. You will receive notification when this happens.

If the solution involves a change to code in a Tivoli Workload Scheduler for z/OS module and no APAR has yet been raised, you might be supplied with a ZAP or a USERMOD. The change will be distributed later as a PTF. If you want a PTF to resolve a specific problem, you can order it explicitly by its PTF number through the IBM Support Center.

Chapter 2. Initial problem analysis

This chapter defines problems that might occur in Tivoli Workload Scheduler for z/OS. The external symptoms of each problem are described to help you or an IBM representative to classify your problem. After you identify the problem type, turn to Chapter 3, "Problem analysis procedures," on page 13 to document the problem further and to build a keyword string. The keyword string will be used by the IBM Support Center to search a database to determine if your problem has been previously reported. You can use a copy of the Tivoli Workload Scheduler for z/OS problem description sheet in Appendix C, "Problem description sheet," on page 481 to record all the information related to the problem.

Steps in classifying the problem

When you are classifying the problem, you should go through the following steps:

1. Select the related problem-type keyword by matching one or more of the symptoms described in this chapter with the symptoms of your Tivoli Workload Scheduler for z/OS problem.
2. Turn to the corresponding procedure in Chapter 3, "Problem analysis procedures," on page 13, and gather the relevant information for your specific problem.
3. Provide the information that is needed for all problems, as described under "Information needed for all problems" on page 24. Continue building the keyword string.

If you are unable to match your problem to any of those described here, you should still compile the information needed for all problems (item 3 above).

Problem-type keywords

The problem-type keywords are used to identify the failure that occurred. The following are the keywords and the problem types they identify:

ABEND
Abnormal termination with a system abend code

ABENDU
Abnormal termination with a user abend code

DOC Documentation

INCORROUT
Incorrect output

LOOP Loop

MSG Message

PERFM
Performance

WAIT Wait

ABEND

Choose the **ABEND** keyword when the Tivoli Workload Scheduler for z/OS program comes to an abnormal end with a system abend code. You should also use ABEND when any program that services Tivoli Workload Scheduler for z/OS (for example, VTAM®) terminates it, and one of the following symptoms appears:

- An abend message at an operator console. The abend message contains the abend code and is found in the system console log.
- A dump is created in a dump data set.

If a message was issued, use the MSG keyword to document it. See "Abnormal termination (ABEND or ABENDU) procedure" on page 13 to describe the problem.

ABENDU

Choose the **ABENDU** keyword when the Tivoli Workload Scheduler for z/OS program comes to an abnormal end with a user abend code, and the explanation of the abend code states that it is a program error. Also choose this keyword when a user abend (which is not supposed to signify a program error) occurs when it should not occur, according to the explanation. If a message was issued, use the MSG keyword to document it.

For a description of the problem, see "Abnormal termination (ABEND or ABENDU) procedure" on page 13.

DOC

Choose the **DOC** keyword when one or more of the following symptoms appear:

- There is incomplete or inaccurate information in an Tivoli Workload Scheduler for z/OS publication.
- The published description of Tivoli Workload Scheduler for z/OS does not agree with its actual operation.

See "Documentation (DOC) procedure" on page 15 to describe the problem.

Note: Report a documentation problem only when it meets one of the criteria listed.

INCORROUT

Choose the **INCORROUT** keyword when one or more of these symptoms appear:

- You received unexpected output, and the problem does not appear to be a loop.
- The output appears to be incorrect or incomplete.
- The output is formatted incorrectly.
- The output comes from damaged files or from files that are not set up or updated correctly.

Turn to "Incorrect output (INCORROUT) procedure" on page 16 to describe the problem.

LOOP

Choose the **LOOP** keyword when one or more of the following symptoms exist:

- Part of the program, other than a message, is repeating itself.
- An Tivoli Workload Scheduler for z/OS command has not completed after an expected period of time, and the processor usage is at higher-than-normal levels.
- The processor is used at higher-than-normal levels, a workstation operator experiences terminal lockout, or there is a high channel activity to an Tivoli Workload Scheduler for z/OS database.

Turn to "LOOP" to describe the problem.

MSG

Choose the **MSG** keyword to specify a message failure. You use this keyword when an Tivoli Workload Scheduler for z/OS problem causes an Tivoli Workload Scheduler for z/OS error message. The message might appear at the system console or in the Tivoli Workload Scheduler for z/OS message log, or both. If you use the end-to-end feature, the messages also appear in the STDLIST subdirectory of the work directory (defined by the WRKDIR keyword of the TOPOLOGY statement).

The messages issued by Tivoli Workload Scheduler for z/OS appear in the following formats:

EQQ*FnnnC*
EQQ*FFnnC*
EQQ*nnnnC*

The message is followed by the message text. The variable components represent:

F or FF
The Tivoli Workload Scheduler for z/OS component that issued the message

nn, nnn, or nnnn
The message number

C A severity code of I (information), W (warning), or E (error).

Tivoli Workload Scheduler for z/OS message-number examples

EQQW002E
EQQSU21E
EQQ0370W

The message log of the end-to-end server can list also Tivoli Workload Scheduler messages. They have the following format:

```
AWS<component_code><numeric_identifier><severity>
```

See the Tivoli Workload Scheduler *Administration and Troubleshooting Guide* for message reference.

If the message of the log file contained in the STDLIST directory does not have the EQQ or AWS prefixes, your problem is probably not associated with Tivoli Workload Scheduler for z/OS, and you should not use the MSG keyword.

Turn to “Message (MSG) procedure” on page 16 to describe the problem.

Note: In cases such as the following, you might want to use **INCORROUT** in addition to the MSG keyword:

- A message describes conditions that do not apply to the actual program operation.
- A message text has not appeared.
- A message text appears to contain incorrect data.
- Tivoli Workload Scheduler for z/OS has printed a message that starts with EQQ and the message is not documented, or is not documented correctly, in *Messages and Codes*.
- A message is issued under conditions that should not have resulted in it being issued.

PERFM

Choose the **PERFM** keyword when one or more of the following symptoms appear:

- Tivoli Workload Scheduler for z/OS event processing or commands, including commands entered from a terminal in session with Tivoli Workload Scheduler for z/OS, take an excessive amount of time to complete.
- Tivoli Workload Scheduler for z/OS performance characteristics do not meet explicitly stated expectations. Describe the actual and expected performances and the explicit source of the performance expectation.

Turn to “Performance (PERFM) procedure” on page 17 to describe the problem.

WAIT

Choose the **WAIT** keyword when one or more of the following symptoms appear:

- The Tivoli Workload Scheduler for z/OS program, or any program that services this program, has suspended activity while waiting for a condition to be satisfied without issuing a message to indicate why it is waiting.
- The console operator cannot enter commands or otherwise communicate with Tivoli Workload Scheduler for z/OS, and Tivoli Workload Scheduler for z/OS does not appear to be in a loop.
- If the last instruction executed was not a WAIT SVC instruction (X'0A01'), you might not be in a true wait state. You should then analyze the problem further to determine whether you are in a loop or executing code as expected.

See “WAIT” to describe the problem.

Chapter 3. Problem analysis procedures

This chapter details the procedures that you use to further describe a problem. First, you gather the information for the specific problem type. Then you provide the information that is common to all problems (see "Information needed for all problems" on page 24). You should compile this information even if you are unable to identify the specific problem type.

Information needed for a specific problem type

When you have chosen a problem-type keyword, see Chapter 2, "Initial problem analysis," on page 9, collect problem documentation and create a keyword string to describe the problem. To do this, gather the information for the specific problem (the associated keyword is in parentheses).

- System or user abnormal-termination procedure (ABEND or ABENDU)
- Documentation procedure (DOC)
- Incorrect output procedure (INCORROUT)
- Loop procedure (LOOP)
- Message procedure (MSG)
- Performance procedure (PERFM)
- Wait procedure (WAIT)
- Procedure to resolve event-reapplying problems at controller startup

After collecting the documentation using a specific procedure, go to "Information needed for all problems" on page 24 to complete the problem description. You can use the Tivoli Workload Scheduler for z/OS problem description sheet, see Appendix C, "Problem description sheet," on page 481, to record all the information related to the problem.

Abnormal termination (ABEND or ABENDU) procedure

A malfunction in the system can cause an abnormal termination (abend). Abend categories are:

- User abend
- System abend
- Program check abend.

User abends originate in the application program. Tivoli Workload Scheduler for z/OS abend codes are documented in Appendix A, "Abend codes," on page 475 and also in *Messages and Codes*

System abends can occur, for example, when a system supervisor-call instruction (SVC) is issued in a program with an incorrect event-control-block (ECB) address. Refer to *z/OS Codes* and *z/OS Messages*

Program check abends are hardware-detected error conditions, such as a store to an invalid address or an attempt to execute an invalid instruction.

To correct the problem, you need to know which part of the code the problem is in. For any abend, you need to determine which module failed. For example, was the failure in:

- An Tivoli Workload Scheduler for z/OS module?

- User-written code, an exit, or an Tivoli Workload Scheduler for z/OS program interface (PIF) application?
- Another IBM licensed program running in the Tivoli Workload Scheduler for z/OS address space?
- A PIF application running in another address space?
- The Dynamic Workload Console?

If your Tivoli Workload Scheduler for z/OS problem is an abnormal termination, you should:

1. Determine the name of the active Tivoli Workload Scheduler for z/OS task. The active task can be determined only if you have a dump data set. You can use the SYSMDUMP dump data set and format the output using interactive problem control system (IPCS).
2. Check the name of the failing load module. If the first 3 characters are EQQ, an Tivoli Workload Scheduler for z/OS module has abended.
3. Locate and record the abend code.
4. Locate and record the name of the failing module and the offset of the failing instruction in the current CSECT of that module. The address of the failing instruction is given by the program status word (PSW) at entry to the abend information.

   Locate the failing instruction in the dump. The base register will normally be register 11 or 12. Determine the values of these registers by locating the `Registers at entry to ABEND` area in the dump. Either register 11 or register 12 will have a value that is less than the address of the failing instruction but within X'1000' of this address. This register is the base register, which you use to find the beginning of the failing CSECT in the dump. At offset 5 in the failing CSECT, you will find a module identification that contains the name of the failing CSECT and the compilation date of this CSECT.
5. Record the name and compilation date of the failing CSECT. Calculate and record the offset of the failing instruction in the failing CSECT.
6. Locate the save-area trace information in the dump. You should be able to find a line containing `AT EP entry-point-name` in the save-area trace. The entry-point-name is the module identification of the failing CSECT. The first word on the same line is the name of the failing load module.
7. Record the name of the failing load module.
8. Compile the required information described in “Information needed for all problems” on page 24.

Tivoli Workload Scheduler for z/OS abend codes are listed in Appendix A, “Abend codes,” on page 475.

Note: This procedure does not apply to the end-to-end server address space when the failing module is a USS process.

The diagnostic file (EQQDUMP)

When Tivoli Workload Scheduler for z/OS internal validity checking discovers error conditions within the network communication function, debugging information is written to the Tivoli Workload Scheduler for z/OS diagnostic file (defined by ddname EQQDUMP). For serious error conditions, Tivoli Workload Scheduler for z/OS abends with user code 3999 as well.

The diagnostic information consists of message EQQ0000T, which gives the name of the module in error and the reason for the error, in two 8-byte character strings. Tivoli Workload Scheduler for z/OS also writes a formatted version of the trace table to the diagnostic file. In most situations, Tivoli Workload Scheduler for z/OS will also “snap” the data that it considers to be in error.

Trace information:
Tivoli Workload Scheduler for z/OS maintains an internal trace to make it possible to see the order that its modules have been invoked in prior to an abend. The trace is wraparound with an end mark after the last trace entry added. Each entry consists of two 8-byte character fields: the module name field and the reason field. The end mark consists of a string of 16 asterisks (X'5C'). For most abnormal terminations, a trace table is written in the diagnostic file (EQQDUMP). These trace entries are intended to be used by IBM staff when they are diagnosing Tivoli Workload Scheduler for z/OS problems.

A trace entry with reason PROLOG is added on entry to the module. Similarly, an entry with EPILOG is added at exit from the module. When trace entries are added for other reasons, the reason is provided in the reason field. When the dump is online, in machine readable format, the trace table can normally be found by scanning the dump and searching for the strings `PROLOG` or `EPILOG`. There are several trace tables in the dump; one for each Tivoli Workload Scheduler for z/OS task.

Alternatively, the address of the trace table, and its end mark, can be found in the Tivoli Workload Scheduler for z/OS BIB block. This block is found via register 13 in a dump.

- If the dump was produced by a user abend (3900 or 3999), the address of the BIB is found by looking at offset X'4' of the area that register 13 points to. This offset contains a pointer to another area where the address of the BIB is found at offset X'48'. The BIB block is easy to identify because its first characters hold the string `BIB`.
- If the dump was produced by any other abend, register 13 points directly to an area where the address of the BIB is found at offset X'48'.

The address of the trace table is found at offset X'440' in the BIB, and the address of the end mark of the trace is found at offset X'448'.

Documentation (DOC) procedure

If your problem is related to Tivoli Workload Scheduler for z/OS documentation, perform the following steps:

1. Identify the order number of the manual that contains the inaccurate information. The order number appears on the back cover of the manual in the form *xxxx-xxxx-yy*, where *yy* is a 2-digit revision code. Be sure to include the revision code if it appears.

   Technical newsletters (TNLs) are issued for some manuals. TNLs consist of updated pages for a manual. If you are reporting inaccurate information contained in a TNL, identify the TNL number. You will find this number, in the form *xxxx-xxxx,* at the top of any page in the TNL.
2. Indicate the page or pages in the document that contain the incorrect or incomplete information.
3. Prepare a description of the problem that was caused by the incorrect or incomplete information.
4. Gather the required documentation described in “Information needed for all problems” on page 24.

Incorrect output (INCORROUT) procedure

If your problem concerns incorrect output, perform the following steps:

1. Specify which output is incorrect.
2. Specify how the output differs from what was expected:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the output incorrect?

   Note: Sometimes problems with other licensed programs or an exit can generate incorrect output.
3. Compile the required documentation described in "Information needed for all problems" on page 24.

LOOP procedure

If your problem type is LOOP, perform the following steps:

1. Use the Tivoli Workload Scheduler for z/OS message log or system console log to help you identify what happened just before the program loop occurred.
2. Obtain a dump, using the z/OS DUMP command. If the dump options do not include RGN and GRSQ, then both options should be specified as SDATA values on the DUMP command.
3. Document instruction addresses from within the loop, if possible.
4. Provide a description of the situation leading up to the problem.
5. Gather the required documentation described in "Information needed for all problems" on page 24.

Message (MSG) procedure

If your Tivoli Workload Scheduler for z/OS problem type is MSG, perform the following steps:

1. Look up the message in *Messages and Codes* for an explanation. This manual includes information on what action Tivoli Workload Scheduler takes and what action the operator should take in response to a message. If you plan to report the problem, gather the documentation before you take action.
2. Copy the message identifier and the message text. The IBM Support Center representative needs the exact message text.
3. Supplement the MSG keyword with the message identifier. You use the supplemented keyword in your keyword string when searching the software support database. The keyword format can be one of the following:

   MSGEQQ*FnnnC*
   MSGEQQ*FFnnC*
   MSGEQQ*nnnnC*.

   The variable components represent:

   F or FF
   The Tivoli Workload Scheduler for z/OS component that issued the message

   nn, nnn, or nnnn
   The message number, from 000 to 999

   C A severity code of I (information), W (warning), or E (error).

   Message-number examples
   EQQW002E
   EQQSU21E

EQQ0370W

4. Compile the required documentation described in "Information needed for all problems" on page 24.

Performance (PERFM) procedure

If your problem concerns performance, perform the following steps:

1. Check that all the parameters have been set as described in the section about how tuning the controller and tracker of *Customization and Tuning*.
2. Document the actual performance, the expected performance, and the source of information for the expected performance. If a document is the source, note the order number and page number of the document.
3. Document the information about your operating environment, such as:
   - The number of active initiators, number of TSO users, and number of Tivoli Workload Scheduler for z/OS users connected.
   - Other licensed programs that are active in your environment.
   - Any user modifications to the program. Exits, REXX programs, and command lists can affect performance. You should consider whether the user-installed code, REXX programs, or CLISTs are contributing to the problem.
4. If you are using the end-to-end feature, document the network topology.
5. Document any modifications to your system. Performance problems can be related to various system limitations. Your market division representative might be able to identify possible causes of a performance problem.
6. Compile the required documentation described in "Information needed for all problems" on page 24.
7. Obtain a dump, using the z/OS DUMP command.

WAIT procedure

If your problem type is WAIT, perform the following steps:

1. Research the activity before the system activity was suspended, identifying any operations that are in the wait state.
2. Specify any messages that were sent to the Tivoli Workload Scheduler for z/OS message log or to the system console.
3. Obtain a dump by using the z/OS DUMP command. If the dump options do not include RGN and GRSQ, then specify both options as SDATA values on the DUMP command. Format the trace table contained in the dump, and determine which task issued the WAIT.

   Alternatively, determine the module in which the WAIT occurred by locating the address of the last instruction that ran. This instruction should be a WAIT SVC (X'0A01'), otherwise perform further analysis to determine if the program is in a loop or if the code is running as expected:
   a. Record the name and the compilation date of the module.
   b. Record the offset into the module.
   c. Provide a description of the situation leading up to the problem.
4. Gather the required documentation as described in "Information needed for all problems" on page 24.

Procedure to resolve event-reapplying problems at controller startup

When the controller starts and a current plan already exists, you must ensure that no JT events were lost while the controller was stopped. JTOPTS CURRPLAN is the statement used by the controller to determine from which current plan to start. You can set it to one of the following values:

- Set CURRPLAN(NEW) to start from the latest valid new current plan (NCP) and apply all the events logged in EQQJTARC and in the current EQQJT*xx*.

  - OR -
- Set CURRPLAN(CURRENT) to start from the inactive current plan and apply the events logged in EQQJT*xx* with a date and time later than the inactive current plan backup.

Note: For detailed information about the JTOPTS statement, see *Customization and Tuning*.

However, if you set the CURRPLAN(CURRENT) option and the controller finds some inconsistencies, the option is not applied and you are requested to set CURRPLAN(NEW). The restarting process then runs based on the checkpoint data set EQQCKPT, which contains information about the current status of Tivoli Workload Scheduler for z/OS.

If problems occur during the restart, you can understand what went wrong by analyzing the initial values stored in the current plan VSAM headers, checkpoint data set, and job-tracking files. This data, however, might have changed or been lost during the restarting process and be difficult to reproduce. To prevent this problem, Tivoli Workload Scheduler for z/OS logs the following messages and provides you with the following modify command:

- Message EQQN140I, to show the checkpoint data set content at controller startup and stop.
- Message EQQN141I, to show the checkpoint data set content when the restart processing begins.
- Modify command `/F` *`subsys`*`,DCKPT` to show the checkpoint data set header.

The following scenario describes how these messages and command help you to understand what happened when the controller restarts.

Understanding the restarting process

The following scenario describes how you can use EQQMLOG and modify command `/F` *`subsys`*`,DCKPT` to understand what happens when the controller restarts. Suppose that you:

1. Start the controller after a REFRESH function. EQQMLOG stores message EQQN140I to show the checkpoint data set header at startup. The value for PLAN EXISTING is N, meaning that a current plan does not exist (because the restart occurred after a REFRESH).

   Note: In this scenario JTOPTS is set to CURRPLAN(CURRENT).

```
...
03/20 17.16.22 EQQZ015I INIT STATEMENT:         CURRPLAN(CURRENT)
....
03/20 17.16.22 EQQN140I CKPT FILE AT START TIME
03/20 17.16.22 EQQN140I --------------------------------------------
03/20 17.16.22 EQQN140I CKPT FILE JUST INITIALIZED: N
```

```
03/20 17.16.22 EQQN140I BACKUP IN PROGRESS        : N
03/20 17.16.22 EQQN140I INACTIVE PLAN DATA SET    : EQQCP2DS
03/20 17.16.22 EQQN140I CURRENT JT NUMBER         : 05
03/20 17.16.22 EQQN140I LAST ARCHIVED JT NUMBER   : 04
03/20 17.16.22 EQQN140I PLAN EXISTING             : N
03/20 17.16.22 EQQN140I ACTIONS ON NCP            : NO ACTIONS NEEDED
03/20 17.16.22 EQQN140I ---------------------------------------------
```

Because a current plan does not exist, the following message EQQN141I does not provide information that is useful for the controller restarting decision.

```
03/20 17.16.24 EQQN141I CKPT DATA AFTER RESTART DECISION
03/20 17.16.24 EQQN141I -----------------------------------
03/20 17.16.24 EQQN141I INACTIVE PLAN DATA SET      : EQQCP2DS
03/20 17.16.24 EQQN141I OCP  LAST BACKUP TIMESTAMP  :
03/20 17.16.24 EQQN141I USED LAST BACKUP TIMESTAMP  :
03/20 17.16.24 EQQN141I EVENTS FOUND                : N
03/20 17.16.24 EQQN141I JTARC MAY BE TRUNCATED      : N
03/20 17.16.24 EQQN141I SYMPHONY CURRENT RUN NUMBER : 0000
03/20 17.16.24 EQQN141I ----------------------------------
```

Message EQQN105W provides you with even more detailed information:

```
03/20 17.16.24 EQQN105W NO VALID CURRENT PLAN EXISTS. CURRENT PLAN VSAM
               I/O IS NOT POSSIBLE
```

Message EQQN012I informs you that, as soon as you create a plan, all job-tracking events will be stored in file EQQJT05 (this is the JT number 05 shown in the previous message EQQN140I):

```
03/20 17.16.24 EQQN012I JOB TRACKING EVENTS ARE NOW BEING LOGGED ON FILE
               EQQJT05
```

To look at the entire content of the checkpoint data set, enter the following modify command:

```
/F subsys,DCKPT
```

```
EQQZ024I LAST CKPT HEADER GET/PUT
EQQZ024I     START ADDRESS: 0007FF68                 TIME: 130320 1716488
EQQZ024I     000000  F0F0F0F0 F0F0F0F0 F0F00000 00000000  *0000000000...*
EQQZ024I     000010  00004040 40404040 40404040 C5D8D8C3  *..             E*
EQQZ024I     000020  D7F1C4E2 C5D8D8C3 D7F2C4E2 F0F5F0F4  *P1DSEQQCP2DS0*
EQQZ024I     000030  F0F5D500 00000000 00000000 D5D5D5C5  *05N.........N*
EQQZ024I     000040  D8D8D1E2 F1C4E2D5 F1F0C8E6 E2F9F1F0  *QQJS1DSN10HWS*
EQQZ024I     000050  F040D500 00000000 404040D6 00640000  *0 N.....   O.*
EQQZ024I     000060  00000000 00000000                    *........        *
EQQZ048I A MODIFY COMMAND HAS BEEN PROCESSED. MODIFY CWSG,DCKPT
```

Note: The layout is mapped by the OPC - OPCSTAT record. For detailed information, see "OPC - OPCSTAT record" on page 369.

2. Create a new current plan by running a DP batch. EQQMLOG records that a new current plan (NCP) is created, named EQQCP2DS. The inactive plan is EQQCP1DS, the current job-tracking file is EQQJT01, and EQQJT05 was archived in EQQJTARC.

```
03/20 17.29.27 EQQN121I A DAILY PLANNING ACTIVITY WAS STARTED
03/20 17.29.57 EQQN120I DP BATCH NCP CREATION PHASE ENDED
03/20 17.29.57 EQQN116I A NEW NCP HAS BEEN CREATED
...
03/20 17.29.58 EQQN056I A CURRENT PLAN COPY PROCESS HAS STARTED
03/20 17.29.58 EQQN057I A CURRENT PLAN DATA SET WAS SUCCESSFULLY COPIED:
               FROMDD=EQQNCPDS, TODD=EQQCP1DS
...
03/20 17.29.59 EQQN051I A CURRENT PLAN BACKUP PROCESS HAS STARTED.
               TRIGGER WAS: DP END
...
```

```
03/20 17.30.00 EQQN012I JOB TRACKING EVENTS ARE NOW BEING LOGGED ON FILE
               EQQJT01
...
03/20 17.30.04 EQQN056I A CURRENT PLAN COPY PROCESS HAS STARTED
03/20 17.30.05 EQQN057I A CURRENT PLAN DATA SET WAS SUCCESSFULLY COPIED:
               FROMDD=EQQCP1DS, TODD=EQQCP2DS

03/20 17.30.05 EQQN090I THE JOB TRACKING LOG DATA SET DEFINED BY DDNAME
               EQQJT05 HAS BEEN COPIED TO THE JOB TRACKING LOG ARCHIVE
               DATA SET
```

3. Stop the controller by using the /P *subsys* command. EQQMLOG shows the following messages:

```
03/20 18.09.08 EQQZ000I A STOP COMMAND HAS BEEN RECEIVED
...
03/20 18.09.14 EQQN051I A CURRENT PLAN BACKUP PROCESS HAS STARTED.
               TRIGGER WAS: NMM END
```

   Before closing, the controller creates a current plan backup. EQQMLOG shows you the checkpoint data set information recorded *before* the CP backup, as follows:

```
03/20 18.09.14 EQQN140I CKPT FILE AT NMM ENDING: BACKUP
03/20 18.09.14 EQQN140I --------------------------------------------
03/20 18.09.14 EQQN140I CKPT FILE JUST INITIALIZED: N
03/20 18.09.14 EQQN140I BACKUP IN PROGRESS        : N
03/20 18.09.14 EQQN140I INACTIVE PLAN DATA SET    : EQQCP1DS
03/20 18.09.14 EQQN140I CURRENT JT NUMBER         : 01
03/20 18.09.14 EQQN140I LAST ARCHIVED JT NUMBER   : 05
03/20 18.09.14 EQQN140I PLAN EXISTING             : Y
03/20 18.09.14 EQQN140I ACTIONS ON NCP            : NO ACTIONS NEEDED
03/20 18.09.14 EQQN140I --------------------------------------------

03/20 18.09.19 EQQN056I A CURRENT PLAN COPY PROCESS HAS STARTED
03/20 18.09.19 EQQN057I A CURRENT PLAN DATA SET WAS SUCCESSFULLY COPIED:
               FROMDD=EQQCP2DS, TODD=EQQCP1DS
03/20 18.09.19 EQQN090I THE JOB TRACKING LOG DATA SET DEFINED BY DDNAME
               EQQJT01 HAS BEEN COPIED TO THE JOB TRACKING LOG ARCHIVE
               DATA SET
```

   After the CP backup completes, the checkpoint data set information is updated. EQQMLOG shows that the JT data set was archived, the inactive plan is EQQCP2DS, the current JT data set has become EQQJT02, and the latest archived JT data set is EQQJT01. For more detailed information about the archiving process, see “Job-tracking-log archiver (JL)” on page 68.

```
03/20 18.09.19 EQQN090I COPIED TO THE JOB TRACKING LOG ARCHIVE DATA SET
03/20 18.09.19 EQQN140I CKPT FILE AT NMM ENDING: LOG ARCHIVE
03/20 18.09.19 EQQN140I --------------------------------------------
03/20 18.09.19 EQQN140I CKPT FILE JUST INITIALIZED: N
03/20 18.09.19 EQQN140I BACKUP IN PROGRESS        : N
03/20 18.09.19 EQQN140I INACTIVE PLAN DATA SET    : EQQCP2DS
03/20 18.09.19 EQQN140I CURRENT JT NUMBER         : 02
03/20 18.09.19 EQQN140I LAST ARCHIVED JT NUMBER   : 01
03/20 18.09.19 EQQN140I PLAN EXISTING             : Y
03/20 18.09.19 EQQN140I ACTIONS ON NCP            : NO ACTIONS NEEDED
03/20 18.09.19 EQQN140I --------------------------------------------
```

4. Restart the controller. Message EQQN140I shows exactly the same information that was recorded in the latest run.

```
03/20 19.53.55 EQQZ015I INIT STATEMENT:          CURRPLAN(CURRENT)
...
03/20 19.53.55 EQQN140I CKPT FILE AT START TIME
03/20 19.53.55 EQQN140I --------------------------------------------
03/20 19.53.55 EQQN140I CKPT FILE JUST INITIALIZED: N
03/20 19.53.55 EQQN140I BACKUP IN PROGRESS        : N
03/20 19.53.55 EQQN140I INACTIVE PLAN DATA SET    : EQQCP2DS
03/20 19.53.55 EQQN140I CURRENT JT NUMBER         : 02
```

```
03/20 19.53.55 EQQN140I LAST ARCHIVED JT NUMBER   : 01
03/20 19.53.55 EQQN140I PLAN EXISTING             : Y
03/20 19.53.55 EQQN140I ACTIONS ON NCP            : NO ACTIONS NEEDED
03/20 19.53.55 EQQN140I ---------------------------------------------
...
03/20 19.53.56 EQQN141I CKPT DATA AFTER RESTART DECISION
03/20 19.53.56 EQQN141I ---------------------------------------------
03/20 19.53.56 EQQN141I INACTIVE PLAN DATA SET      : EQQCP2DS
03/20 19.53.56 EQQN141I OCP  LAST BACKUP TIMESTAMP  : 130320/17091869
03/20 19.53.56 EQQN141I USED LAST BACKUP TIMESTAMP  : 130320/17091869
03/20 19.53.56 EQQN141I EVENTS FOUND                : N
03/20 19.53.56 EQQN141I JTARC MAY BE TRUNCATED      : N
03/20 19.53.56 EQQN141I SYMPHONY CURRENT RUN NUMBER : 0000
03/20 19.53.56 EQQN141I ---------------------------------------------
...
```

Suppose that the controller runs one day and performs all the jobs scheduled in the CP and then you run a CP backup. EQQMLOG shows that the inactive plan is now EQQCP1DS and the latest archived JT data set is EQQJT02.

```
03/21 15.48.50 EQQN051I A CURRENT PLAN BACKUP PROCESS HAS STARTED.
               TRIGGER WAS: BACKUP CMD
03/21 15.48.50 EQQN012I JOB TRACKING EVENTS ARE NOW BEING LOGGED ON FILE
               EQQJT03
03/21 15.48.53 EQQN056I A CURRENT PLAN COPY PROCESS HAS STARTED
03/21 15.48.53 EQQN057I A CURRENT PLAN DATA SET WAS SUCCESSFULLY COPIED:
               FROMDD=EQQCP1DS, TODD=EQQCP2DS
03/21 15.48.54 EQQN090I THE JOB TRACKING LOG DATA SET DEFINED BY DDNAME
               EQQJT02 HAS BEEN COPIED TO THE JOB TRACKING LOG ARCHIVE
               DATA SET
```

You can verify this information also by issuing the modify command `/F` *`subsys`*`,DCKPT`:

```
EQQZ024I LAST CKPT HEADER GET/PUT
EQQZ024I     START ADDRESS: 0007FF68                TIME: 130321 1552432
EQQZ024I     000000  F0F0F0F0 F0F0F0F0 F0F00000 00000000  *0000000000...
EQQZ024I     000010  0000F4F1 F0F3F2F0 F2F4F0F0 C5D8D8C3  *..4103202400E
EQQZ024I     000020  D7F2C4E2 C5D8D8C3 D7F1C4E2 F0F3F0F2  *P2DSEQQCP1DS0
EQQZ024I     000030  F0F5D500 00000000 00000000 E8D5D5C5  *05N.........Y
EQQZ024I     000040  D8D8D1E2 F1C4E2D5 F1F0C8E6 E2F9F1F0  *QQJS1DSN10HWS
EQQZ024I     000050  F040D500 00000000 404040D6 00640000  *0 N.....   O.
EQQZ024I     000060  00000000 00000000                    *........
EQQZ048I A MODIFY COMMAND HAS BEEN PROCESSED. MODIFY CWSG,DCKPT
```

5. As you continue to work, you add new occurrences to the plan. You then force the controller to immediately stop with a CANCEL function.
6. When you restart the controller, EQQMLOG shows you that the inactive plan is EQQCP1DS and the latest archived JT data set is EQQJT02:

```
03/21 15.55.46 EQQN140I CKPT FILE AT START TIME
03/21 15.55.46 EQQN140I ---------------------------------------------
03/21 15.55.46 EQQN140I CKPT FILE JUST INITIALIZED: N
03/21 15.55.46 EQQN140I BACKUP IN PROGRESS        : N
03/21 15.55.46 EQQN140I INACTIVE PLAN DATA SET    : EQQCP1DS
03/21 15.55.46 EQQN140I CURRENT JT NUMBER         : 03
03/21 15.55.46 EQQN140I LAST ARCHIVED JT NUMBER   : 02
03/21 15.55.46 EQQN140I PLAN EXISTING             : Y
03/21 15.55.46 EQQN140I ACTIONS ON NCP            : NO ACTIONS NEEDED
03/21 15.55.46 EQQN140I ---------------------------------------------
```

The controller uses the information shown in message EQQN141I to decide how to restore a correct current plan. The inactive plan EQQCP1DS is used as the start plan and the events stored in EQQJT03 after the backup date and time 130321/13485048 are applied. The following messages are recorded to show that the inactive plan EQQCP1DS is copied to EQQCP2DS:

```
03/21 15.55.49 EQQN056I A CURRENT PLAN COPY PROCESS HAS STARTED
03/21 15.55.49 EQQN057I A CURRENT PLAN DATA SET WAS SUCCESSFULLY COPIED:
               FROMDD=EQQCP1DS, TODD=EQQCP2DS
```

The following messages show that EQQJT03 is used to update the plan before archiving it, EQQJT04 becomes the current JT data set, and EQQCP1DS becomes the current CP after the CP backup completes. You are also shown that the catch-up process identified and applied all the events required to restore a correct plan:

```
03/21 15.55.50 EQQN058I THE SCHEDULER WILL RECOVER THE CP DATA SET FROM
               THE JTLOGS
03/21 15.55.50 EQQN058I PLEASE BE PATIENT. THIS WILL TAKE A WHILE
03/21 15.55.50 EQQN014I THE JOB TRACKING LOG DATA SET, DDNAME EQQJT03 ,
               HAS BEEN USED TO UPDATE THE CURRENT PLAN.
               2 EVENTS HAVE BEEN APPLIED TO DD-NAME EQQCP2DS
03/21 15.55.50 EQQN051I A CURRENT PLAN BACKUP PROCESS HAS STARTED.
               TRIGGER WAS: CATCH UP
03/21 15.55.53 EQQN056I A CURRENT PLAN COPY PROCESS HAS STARTED
03/21 15.55.54 EQQN057I A CURRENT PLAN DATA SET WAS SUCCESSFULLY COPIED:
               FROMDD=EQQCP2DS, TODD=EQQCP1DS
03/21 15.55.54 EQQN012I JOB TRACKING EVENTS ARE NOW BEING LOGGED ON FILE
               EQQJT04
03/21 15.55.54 EQQN013I JOB TRACKING IS NOW ACTIVE AND CURRENT PLAN
               DD-NAME IS EQQCP1DS
```

Description of messages EQQN140 and EQQN141

Messages EQQN140 and EQQN141 are informational. EQQN140 is issued at controller startup and stop, and shows the following information:

```
CKPT FILE AT: &phase
---------------------------------------------
CKPT FILE JUST INITIALIZED: &NEWCKPT
BACKUP IN PROGRESS        : &BACKUP
INACTIVE PLAN DATA SET    : &OLDCP
CURRENT JT NUMBER         : &CURRJT
LAST ARCHIVED JT NUMBER   : &LASTJT
PLAN EXISTING             : &VALCP
ACTIONS ON NCP            : &NCPSTAT
---------------------------------------------
```

Where:

CKPT FILE AT
It can be one of the following values:

START TIME
Controller has just started.

NMM ENDING: BACKUP
Controller is stopping and the current plan is being backed up.

NMM ENDING: LOG ARCHIVE
Controller is stopping, the current plan is backed up, and the latest JT data set is archived.

CKPT FILE JUST INITIALIZED
It can be one Y or N. Y means that the checkpoint data set has just been initialized with the default values. This might occur the first time that the controller starts, when the checkpoint data set is empty.

BACKUP IN PROGRESS
It can be Y or N. Y means that when the controller was stopped, a backup plan was in progress, therefore the controller could not complete switching the CP and JT data set.

INACTIVE PLAN DATA SET
The DD name of the inactive plan (the old current plan).

CURRENT JT NUMBER
The name of the current JT data set.

LAST ARCHIVED JT NUMBER
The name of the latest archived JT data set.

PLAN EXISTING
It can be Y or N. Y means that a current plan exists.

ACTION ON NCP
The status related to the creation of CP. Possible values are:

- **NCP TO BE USED AS CURRENT PLAN**
  The new current plan is to be used by the controller as the current plan.

- **NCP UNDER CREATION, DP BATCH IS WORKING ON IT**
  A DP batch command is running to create a new plan.

- **NO ACTIONS NEEDED**
  No action to be done on the NCP.

EQQN141 is issued after the restarting decision was taken and shows the following information:

```
CKPT DATA &phase
---------------------------------------------
INACTIVE PLAN DATA SET      : &OLDCP
OCP  LAST BACKUP TIMESTAMP  : &DTCP
USED LAST BACKUP TIMESTAMP  : &DTUS
EVENTS FOUND                : &EVENT
JTARC MAY BE TRUNCATED      : &JTARCT
SYMPHONY CURRENT RUN NUMBER : &SYMPHN
---------------------------------------------
```

Where:

CKPT DATA
It always has the value AFTER RESTART DECISION.

INACTIVE PLAN DATA SET
The DD name of the inactive plan (the old current plan).

OCP LAST BACKUP TIMESTAMP
Date and time of the latest backup in the inactive CP data set.

INACTIVE PLAN DATA SET
The DD name of the inactive plan (the old current plan).

USED LAST BACKUP TIMESTAMP
Date and time that the latest backup was actually used. Normally it is the same as OCP LAST BACKUP TIMESTAMP. However, if a backup was in progress at stop time and the backup is not reliable, the controller considers the date and time of the first event in the current JT data set as the latest backup. This date is used to decide which events are to be applied.

EVENTS FOUND
Events to be applied are found in the JT data set.

JTARC MAY BE TRUNCATED
It can be Y or N. Y means that the latest archived JT data set was not complete and that data might have been lost.

JTARC MAY BE TRUNCATED
If a Symphony plan exists, it indicates its current number.

Information needed for all problems

Even when you are unable to identify a problem type, you should gather the following information for any problem you have. Begin your initial problem analysis by examining the contents of the message log data set. Record relevant information on a copy of the Tivoli Workload Scheduler for z/OS problem description sheet provided in Appendix B.

1. Obtain a copy of the Tivoli Workload Scheduler for z/OS message log. This is a sequential data set defined by the EQQMLOG ddname.
2. Record the Tivoli Workload Scheduler for z/OS component ID: 5697-WSZ01. The component ID should be the *first* keyword in the string preceding the problem type and other modifier keywords.
3. Record the maintenance level for all operating environments, particularly those for z/OS, JES, ISPF, and RACF®.
4. Document any additional program temporary fixes (PTFs) or APARs that have been applied to your level of Tivoli Workload Scheduler for z/OS.
5. If the problem is within the network communication function, obtain copies of the Tivoli Workload Scheduler for z/OS EQQDUMP file.
6. Obtain copies of the Tivoli Workload Scheduler for z/OS diagnostic files defined to the user address space and to the subsystem address space by SYSMDUMP.
7. Obtain a copy of the system log.
8. If the problem involves end-to-end scheduling, obtain a copy of the work directory in the following way:
   a. Identify the work directory. To do this, read in the configuration parameter the value of the WRKDIR keyword of the TOPOLOGY statement.
   b. Create a tar file of the work directory. To do this:
      1) Logon on a USS shell (using OMVS or telnet).
      2) Assure that you are running with uid 0 (use the *id* command to see your uid). If you are authorized to the BPX.SUPERUSER RACF facility class, issue the *su* command to obtain uid 0.
      3) Create the tar file by entering this command:

         ```
         tar cvzf tarfilename workdirectory
         ```
   c. Collect TWSIN and TWSOU.
   d. Collect the topology information in the PARMLIB library (DD name EQQPARM).
9. Reconstruct the sequence of events leading to the problem. Include any commands entered just before the problem occurred.

   Write down the exact events that lead to the problem:
   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?
   e. Can you re-create the problem?
10. Specify any unique information about the problem or about your system:

   a. Indicate any other applications that were running when the problem occurred.
   b. Describe how Tivoli Workload Scheduler for z/OS was started.
   c. Describe all user modifications to active Tivoli Workload Scheduler for z/OS programs.

If more information is needed, an IBM Support Center representative will guide you concerning any additional diagnostic traces that you can run.

For information about diagnosing problems with the APPC/z/OS subtask, see "APPC/z/OS (PP)" on page 57.

Information to collect for OCL problems

Information to collect when you encounter OCL problems

If you receive an error message related to OCL, gather the following information, for any problem that you have.

1. Reconstruct the sequence of events leading up to the problem. Include any commands entered just before the problem occurred.

   Write down the exact events that lead to the problem:
   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?
   e. Can you re-create the problem?
2. Obtain a copy of the OCL job log.
3. Collect a SLIP dump for the message.
4. Collect the PIF trace.
5. Indicate any other applications that were running when the problem occurred.

Information to collect for restart and cleanup problems

Information to collect when you encounter restart and clean up problems

If you receive an error message related to restart and cleanup problems, gather the following information:

1. Reconstruct the sequence of events leading up o the problem. Include any commands entered just before the problem occurred. Write down the exact events that lead to the problem:
   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?
   e. Can you re-create the problem?
2. Obtain a copy of the job log of the first job you submitted.
3. Obtain a copy of the restart and cleanup job log or of the stand-alone cleanup job log.
4. Obtain a copy of the controller MLOG.
5. Obtain a copy of the data store MLOG.
6. Obtain a copy of the unstructured data files.
7. Indicate any other applications that were running when the problem occurred.

Information to collect for critical path problems

Information to collect when you encounter critical path problems

If you receive an error message related to critical path problems, gather the following information:

1. When possible, reconstruct the sequence of events that caused the problem. Include any commands entered just before the problem occurred.
   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?
2. Before running the daily plan batch, collect an APAR tape including the new current plan and a console dump including data spaces as shown below:

```
DUMP COMM=(reason for taking dump)
R xx,JOBNAME=(ZZZZ),CONT
R xx,DSPNAME=('ZZZZ'.*),CONT
R xx,SDATA=(COUPLE,ALLNUC,LPA,LSQA,PSA,RGN,SQA,TRT,
CSA,GRSQ,XESDATA,WLM),END
```

   where:

   xx Specify the replay number ID.

   ZZZZ Specify the name of the controller.

Information to collect for JCL editing problems

Information to collect when you encounter JCL editing problems

If you receive an error message that is related to JCL editing, gather the following information, for any problem that you have.

1. Reconstruct the sequence of events that caused the problem. Include any commands that you entered just before the problem occurred.

   Write down the exact events that lead to the problem:
   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?
   e. Can you re-create the problem?
2. Obtain a copy of the controller MLOG.
3. Obtain a copy of the server MLOG.
4. Obtain a copy of the Joblib data set.

Information to collect for migration problems

Information to collect when you encounter migration problems

If you receive an error message that is related to migration, gather the following information, for any problem that you have.

1. Reconstruct the sequence of events that caused the problem. Include any commands that you entered just before the problem occurred.

   Write down the exact events that lead to the problem:
   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?

   e. Can you re-create the problem?
2. Collect the JCL used to run the migration.
3. Collect the input data sets.
4. Collect the output data sets, if generated.
5. Collect the EQQDUMP data set.
6. Obtain a copy of the controller MLOG.

Information to collect for application description problems

Information to collect when you encounter application description problems

If you receive an error message related to JCL, gather the following information, for any problem that you have.

1. Reconstruct the sequence of events leading up to the problem. Include any commands entered just before the problem occurred.

   Write down the exact events that lead to the problem:
   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?
   e. Can you re-create the problem?
2. Obtain a copy of the Application Description and Work Station (WS) cluster files.
3. Obtain a copy of the controller and server MLOGs.
4. Obtain a copy of the DUMP data set, if available.

Information to collect for monitoring problems

Information to collect when you encounter monitoring problems

If you receive an error message related to JCL, gather the following information, for any problem that you have.

1. Reconstruct the sequence of events that caused the problem. Include any commands that you entered just before the problem occurred.

   Write down the exact events that lead to the problem:
   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?
   e. Can you re-create the problem?
2. Obtain a copy of the monitoring data set.
3. Obtain a copy of the controller MLOGs.
4. Obtain a copy of the system log.
5. Obtain a copy of the DUMP data set, if available.

Information to collect for daily planning problems

Information to collect when you encounter daily planning problems

If you receive an error message related to daily planning, gather the following information, for any problem that you have.

1. Reconstruct the sequence of events that caused the problem. Include any commands that you entered just before the problem occurred.

Write down the exact events that lead to the problem:

  a. What was the first indication of the problem?
  b. What were you trying to do?
  c. What should have happened?
  d. What did happen?
  e. Can you re-create the problem?

2. Before running the daily plan batch, collect an APAR tape including the new current plan and a console dump including data spaces as shown below:

```
DUMP COMM=(reason for taking dump)
R xx,JOBNAME=(ZZZZ),CONT
R xx,DSPNAME=('ZZZZ'.*),CONT
R xx,SDATA=(COUPLE,ALLNUC,LPA,LSQA,PSA,RGN,SQA,TRT,
CSA,GRSQ,XESDATA,WLM),END
```

where:

xx Specify the replay number ID.

ZZZZ Specify the name of the controller.

Information to collect for data store problems

Information to collect when you encounter daily planning problems

If you receive an error message related to data store, gather the following information, for any problem that you have.

1. Reconstruct the sequence of events that caused the problem. Include any commands that you entered just before the problem occurred.

   Write down the exact events that lead to the problem:

   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?
   e. Can you re-create the problem?

2. Obtain the original joblog,
3. Obtain the restart and cleanup joblog or the stand-alone cleanup joblog.
4. Obtain a copy of the controller MLOGs, the data store MLOG.
5. Obtain a copy of the UDF and SDF files.
6. Obtain a copy of the EQQDUMP, if available.

Information to collect for general service problems

Information to collect when you encounter general service problems

If you receive an error message related to data store, gather the following information, for any problem that you have.

1. Reconstruct the sequence of events that caused the problem. Include any commands that you entered just before the problem occurred.

   Write down the exact events that lead to the problem:

   a. What was the first indication of the problem?
   b. What were you trying to do?
   c. What should have happened?
   d. What did happen?
   e. Can you re-create the problem?

2. Obtain a copy of the controller MLOGs.

3. Collect the dump.
4. Obtain a copy of the system log.

Part 2. Reference

This part describes the major components, data areas, and DB2 tables.

Chapter 4. Functional description

This chapter describes the major components of Tivoli Workload Scheduler for z/OS and how they fit together. These components are described here:
- Dialogs
- Description databases
- Data sets for planning, tracking, and recovery
- Batch programs
- Subsystem
- Task structure
- Subtasks.

Tivoli Workload Scheduler for z/OS dialogs

Most users of Tivoli Workload Scheduler for z/OS functions use interactive system productivity facility (ISPF) panels to communicate with the product. These panels are referred to as *Tivoli Workload Scheduler for z/OS dialogs*. Each dialog user is also a z/OS TSO user executing in a separate address space, or in a different z/OS system.

The dialogs frequently require support from the Tivoli Workload Scheduler for z/OS subsystem. Tivoli Workload Scheduler for z/OS is a started task that must be active to run the dialogs. The z/OS subsystem interface (SSI) is used to pass requests from the user to Tivoli Workload Scheduler for z/OS. To process the request, this subsystem interface invokes a routine residing in common storage. If the dialogs are executed in a different z/OS system, the requests will be sent to a Tivoli Workload Scheduler for z/OS server on an APPC or TCP/IP session. The server will then forward the requests to the Tivoli Workload Scheduler for z/OS subsystem via the z/OS SSI.

Before performing any function requested by a user, the Tivoli Workload Scheduler for z/OS dialog function uses the system authorization facility (SAF) interface to pass the request to RACF (or a functionally equivalent security package). RACF verifies that the user is authorized to access the resources needed to run the current function. The z/OS router service calls RACF to perform this authority check if RACF is installed and active in the z/OS system. A typical request for service from aTivoli Workload Scheduler for z/OS dialog to the Tivoli Workload Scheduler for z/OS subsystem is to access a set of database records (for example, application description records). The databases, as well as the planning data sets, are maintained and controlled by the started task.

A summary of the system components and interfaces used by Tivoli Workload Scheduler for z/OS dialogs is provided in Figure 1 on page 34.

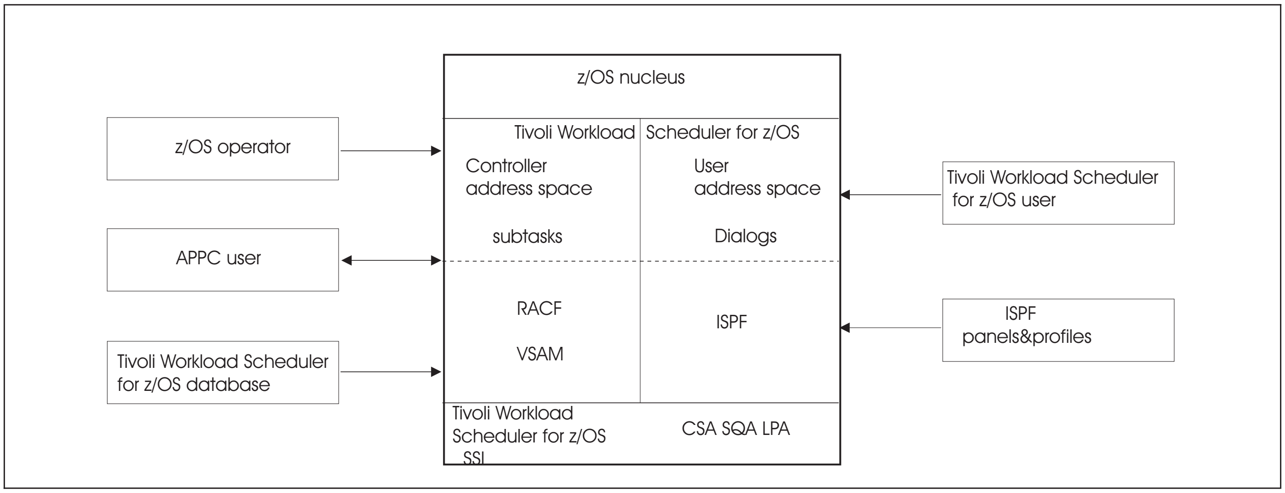

z/OS nucleus
Tivoli Workload Scheduler for z/OS
Controller address space
subtasks
User address space
Dialogs
RACF
VSAM
ISPF
Tivoli Workload Scheduler for z/OS SSI
CSA SQA LPA
z/OS operator
APPC user
Tivoli Workload Scheduler for z/OS database
Tivoli Workload Scheduler for z/OS user
ISPF panels&profiles

Figure 1. Tivoli Workload Scheduler for z/OS dialog structure

ISPF tables

The Tivoli Workload Scheduler for z/OS dialogs use a number of resident ISPF tables:

*pref***ACMDS**
Is the ISPF command table. The prefix *pref* is the name of the current ISPF application. You can specify this as a value on the NEWAPPL keyword when starting the application.

*pref***AXMDL**
Is used to hold information about the dynamic ISPF data tables used in the dialogs (that is, the data tables that are created and deleted during the dialog processing). Typical information includes the table sort order, which resides here from one usage of the dialog to the next. The prefix *pref* is the name of the current ISPF application. You can specify this as a value on the NEWAPPL keyword when starting the application.

EQQELDEF
Contains the ended-in-error-list default layouts.

EQQEVERT
Contains definitions of ended-in-error-list layout variables.

EQQLUDEF
Contains the list of subsystem names and associated LU names that the Tivoli Workload Scheduler for z/OS dialogs user can connect to.

EQQRLDEF
Contains definitions of the ready list layouts. The layouts in this table cannot be modified or deleted in the WORKSTATION COMMUNICATION dialog.

EQQRLOUT
Contains definitions of the ready list layouts. The layouts in this table can be modified and deleted in the WORKSTATION COMMUNICATION dialog. New ready-list layouts created by a user are held in this table.

EQQXVART
Contains definitions of the ISPF data tables used in the dialogs.

Tivoli Workload Scheduler for z/OS data

Tivoli Workload Scheduler for z/OS requires data about your installation environment and the work that is to be processed. It keeps this information in description databases. Tivoli Workload Scheduler for z/OS also maintains a number of other data sets for planning, tracking, and recovery purposes. Another set of data is that managed by the Data Store.Tivoli Workload Scheduler for z/OS data resources are described in this section.

Description databases

The basic data that Tivoli Workload Scheduler for z/OS uses comes from the description databases. These are listed in Table 1.

Table 1. Tivoli Workload Scheduler for z/OS description databases

Database	Physical VSAM data set	DD name
Application description	AD	EQQADDS
Calendar	WS	EQQWSDS
Job descriptions	AD	EQQADDS
Operator instruction	OI	EQQOIDS
Period	WS	EQQWSDS
Special resource description	RD	EQQRDDS
Side information	SI	EQQSIDS
Variable table	AD	EQQADDS
Workstation description	WS	EQQWSDS

Application Description Database

The application description (AD) database contains application descriptions, which consist of:

- A general part
- A part for each run cycle (except for members of an application group)
- A part for each operation (except for group definitions).

Calendar database

The calendar (CAL) database contains data about workdays and free days for work processing.

Job descriptions database

The job descriptions database contains special applications containing a single processing operation.

Operator instruction database

The operator instruction (OI) database contains operator instructions, each of which corresponds to an operation in the AD database.

Period database

The period database contains data used to define run cycles for applications.

Special resources database

The special resources database contains details about individual elements in your installation that you want to identify for scheduling purposes. For example, data sets or tape drives.

Side information database

The side information database contains event-triggered-tracking (ETT) criteria, which are used to add occurrences to the current plan based on triggering events, and configuration information.

Variable table database

The variable table database contains JCL variable definitions, which are grouped into tables. You can associate a variable table with a period, in the period database.

Workstation description database

The workstation description database contains information about the different workstations at the installation.

Data sets used for planning, tracking, and recovery

Tivoli Workload Scheduler for z/OS maintains two plans, the *long-term plan* and the *current plan*. The event-tracking functions collect status information about the work in progress, store this information in event data sets, and amend the current plan to reflect the status of operations. These planning and event data sets, as well as the data sets that are used for recovery, are described here.

When the end-to-end scheduling with fault tolerance capabilities feature is installed, Tivoli Workload Scheduler for z/OS maintains the following additional data sets:

- Input events data set (EQQTWSIN)
- Output events data set (EQQTWSOU)
- Script library (EQQSCLIB)
- Current® plan backup (EQQSCPDS)
- Centralized script data set (EQQTWSCS)

See “Tasks and data sets used for end-to-end scheduling with fault tolerance capabilities” on page 41 for details.

Every Tivoli Workload Scheduler for z/OS address space that uses the Restart and Clean up feature requires the allocation of a local VSAM repository for the structured information related to each job run.

Table 2. Tivoli Workload Scheduler for z/OS planning, tracking, and recovery resources

Resource	Physical data sets	DD name
Checkpoint data set	Checkpoint data set	EQQCKPT

Table 2. Tivoli Workload Scheduler for z/OS planning, tracking, and recovery resources (continued)

Resource	Physical data sets	DD name
Current plan	• Primary current plan • Alternate current plan • New current plan • New current plan extension • Current plan extension • Current plan backup for the creation of Symphony® • Primary extended data • Alternate extended data • New extended data	• EQQCP1DS • EQQCP2DS • EQQNCPDS • EQQNCXDS • EQQCXDS • EQQSCPDS • EQQXD1DS • EQQXD2DS • EQQNXDDS
Centralized script data set for end-to-end scheduling with fault tolerance capabilities	Centralized script	EQQTWSCS
Event data sets for end-to-end scheduling with fault tolerance capabilities	• Input events • Output events	• EQQTWSIN • EQQTWSOU
Script library for end-to-end scheduling with fault tolerance capabilities	Script library	EQQSCLIB
Event data sets	• Event log • Event data set (01–16) for an event reader	• EQQEVDS • EQQEVD*xx*
JCL repository	• Primary JCL repository • Alternate JCL repository	• EQQJS1DS • EQQJS2DS
Job library	Partitioned data set for JCL	EQQJBLIB
Job tracking	• Job-tracking log (maximum 99) • JT archive data set	• EQQJT*xx* • EQQJTARC
Long-term plan	Long-term plan	EQQLTDS
Restart and clean up	• Primary index • Secondary index • Structured data files	• EQQPKI*xx* • EQQSKI*xx* • EQQSDF*xx*
Side information	Side information	EQQSIDS

Checkpoint (EQQCKPT)

The checkpoint data set contains information about which Tivoli Workload Scheduler for z/OS data sets are currently active; for example, in the case of the current plan, the primary or alternate current-plan data set. It also contains record-position information for all event data sets and the Symphony run number.

Current plan (EQQCPnDS)

The current plan (CP) is created from the LTP, AD, RD, and WS databases. The CP consists of VSAM files created by any of the following daily planning batch programs:

- Plan next period. This creates the initial CP and thereafter extends the CP by modifying it with new information from LTP, AD, RD, WS, and the old CP (if available).
- Replan current period. This reorganizes the CP with new information from the databases.

The CP is also updated dynamically by the job-tracking functions, the MODIFY CURRENT PLAN dialog, the program interface (PIF), the application programming interface (API), Dynamic Workload Console, OCL, BCIT, and by events created from RODM notifications.

Tivoli Workload Scheduler for z/OS uses these VSAM data sets when creating or maintaining the current plan:

- The primary current-plan data set, DD name EQQCP1DS.
- The alternate current-plan data set, DD name EQQCP2DS.
- The new current plan (NCP) data set, DD name EQQNCPDS.
- The new current plan extension (NCX) data set, DD name EQQNCXDS.
- The current plan extension (CX) data set, DD name EQQCXDS.
- The current plan backup (SCP) for the creation of the Symphony file, DD name EQQSCPDS.
- The primary "extended data" data set EQQXD1DS
- The alternate "extended data" data set EQQXD2DS
- The new "extended data" data set EQQNXDDS

The "extended data" data sets are used in the same ways as the current plan VSAM files.

When the CP is created, extended, or modified, it is held in the new current-plan data sets, NCP and NCX. The CP turnover process copies the NCP into one of CP1 or CP2 (whichever is inactive), and then this data set becomes the *active* CP. The NCX is copied to the CX data set, and the CX data set is then copied to a data space managed by the controller.

During the creation of the current plan, the SCP data set is used as a CP backup copy for the production of the Symphony file.

The primary and alternate CP data sets (CP1 and CP2), are used in a flip-flop manner; that is, Tivoli Workload Scheduler for z/OS copies the active CP to the inactive data set, and then uses this newly copied data set as the active CP. The active data set is called the CP logical file.

Updates to the CX file are made in the data space. During the current plan backup process, the data space is refreshed to DASD.

For more information, see “Current plan turnover” on page 70.

Centralized script data set for end-to-end scheduling with fault tolerance capabilities (EQQTWSCS)

Tivoli Workload Scheduler for z/OS uses the centralized script data set for end-to-end scheduling with fault tolerance capabilities to temporarily store a script when it is downloaded from the JOBLIB data set to the agent for its submission.

Input and output events for end-to-end scheduling with fault tolerance capabilities (EQQTWSIN and EQQTWSOU)

These data sets are required by every IBM Tivoli Workload Scheduler for z/OS address space that uses end-to-end scheduling with fault tolerance capabilities.

They record the descriptions of events related with operations running on fault-tolerant workstations and are used by both the end-to-end enabler task and the translator process in the scheduler's server.

The layouts used to map these files are DCLEVE and DCLEX0. See the corresponding sections for more information.

Script library for end-to-end scheduling with fault tolerance capabilities (EQQSCLIB)

This script library data set includes members containing the job definitions for operations running on fault-tolerant workstations. It is required in the controller if you want to use end-to-end scheduling with fault tolerance capabilities. It is similar to the JOBLIB with the difference that it describes the Tivoli Workload Scheduler jobs. For details about the JOBREC, RECOVERY, and VARSUB statements, refer to *Scheduling End-to-end with Fault Tolerance Capabilities*.

Event (EQQEVDS and EQQEVDnn)

Each Tivoli Workload Scheduler for z/OS system has an event data set containing records that describe the events that have occurred in the processing of work on that system. The controlling system collects these events in an event log.

The event data set is also used to checkpoint submit actions. The submit subtask uses the first record in the event data set for submit checkpointing.

JCL repository (EQQJS1DS and EQQJS2DS)

Tivoli Workload Scheduler for z/OS maintains its own copy of JCL in the JCL repository data set for every job that it submits in the current plan. Tivoli Workload Scheduler for z/OS uses a primary and alternate data set for the JCL repository, EQQJS1DS and EQQJS2DS. It reorganizes the JCL repository data set that is in use by copying it to the alternate data set and then switching over to use the newly copied data set. The value you specify on the MAXJSFILE keyword defines if the JCL repository should be automatically copied and how often the automatic copy process should occur. You can also request, or schedule, backups with the BACKUP command (see *Managing the Workload* for more information).

JCL is stored on the JS data set for each job that has been submitted in the current plan. When an occurrence is completed, Tivoli Workload Scheduler for z/OS locates all records in the JS data set for that occurrence and flags them as ready for deletion. At the same time, Tivoli Workload Scheduler for z/OS deletes from the JS file all occurrences of the same application, which were previously flagged as ready for deletion. Because JCL records remain on the JS data set until the next occurrence of the same application is set to complete, the JS file should be large enough to hold at least one occurrence of each application that Tivoli Workload Scheduler for z/OS runs. If you delete an occurrence, the JCL is flagged as ready for deletion and is actually deleted when the next occurrence of that application is set to complete.

If you run an application only once, the JCL remains on the JS file indefinitely. If you run many such applications, or run many applications only occasionally, or have changed operation or application names, you can use PIF to remove these unwanted records from the JS file. Two samples, EQQPIFDJ and EQQPIFJD, are provided in the sample library, which you can use to list records in, or delete records from, the JS file.

The space released in the JS file cannot be reused until the file is compressed. The released space is therefore not available until the copy process is performed.

Job library data set (EQQJBLIB)

The job library data set contains the JCL for the jobs and started tasks that IBM Tivoli Workload Scheduler for z/OS will submit. It is required by a controller. If you already have a job library that you use for IBM Tivoli Workload Scheduler for z/OS purposes, specify this data set on the EQQJBLIB statement. If not, allocate one before you start the controller.

Job-tracking (EQQJTxx and EQQJTARC)

The job-tracking log contains data about all updates to the current plan. These are used during recovery to restore the current plan. The job-tracking log also contains audit trail records, which detail accesses to resources.

The job-tracking logs are linked to the new current plan data sets (EQQNCPDS). You can define a maximum of 99 job-tracking logs, with DD name EQQJT*xx*, where *xx* is the sequence number of the JT log. The job-tracking log is switched every time a current-plan copy is performed. see "Current plan turnover" on page 70.

The job-tracking archive data set is defined by the DD name EQQJTARC, and is used by the job-tracking-log archiver subtask as a repository for JT log records associated with each new current plan. The JT archive data set is copied to the tracklog (EQQTROUT) data set by the daily plan batch programs. The NM subtask empties the JT archive data set when a daily plan batch job has created a new current plan, and NCP takeover is completed successfully.

For more information about the job-tracking-log archiver subtask, see "Job-tracking-log archiver (JL)" on page 68.

Long-term plan (EQQLTDS)

The long-term plan (LTP) is created from the AD, calendar, and period databases. The LTP data set is a VSAM file that is created and updated in any of the following ways:

- Create long-term plan. This creates the initial LTP.
- Extend long-term plan. This extends the LTP by appending new information to it. Depending on the value that you specify for the LTPDEPRES keyword of the BATCHOPTS initialization statement, extending the LTP will either leave the information in the existing LTP period unchanged or modify the old part of the plan.
- Modify or replan the long-term plan. This reorganizes the entire LTP, modifying it with new information from the AD, calendar, and period databases.

Restart and clean up data sets (EQQPKIxx, EQQSKIxx, and EQQSDFxx)

Every IBM Tivoli Workload Scheduler for z/OS address space that uses the Restart and Clean Up feature requires the allocation of a local VSAM repository for the structured information related to each job run. These data sets have the same structure as the data store VSAM files and can be allocated by running the EQQPCS07 sample. Keep in mind that every IBM Tivoli Workload Scheduler for z/OS requires the allocation of a unique local VSAM repository.

Side information data set (EQQSIDS)

The side information data set contains event-triggered tracking criteria and JES2 NJE node names. Tivoli Workload Scheduler for z/OS uses the same data set for database functions and current plan functions.

Tasks and data sets used for end-to-end scheduling with fault tolerance capabilities

Figure 2 shows the processes and the data sets used by the controller and by the server for end-to-end scheduling with fault tolerance capabilities.

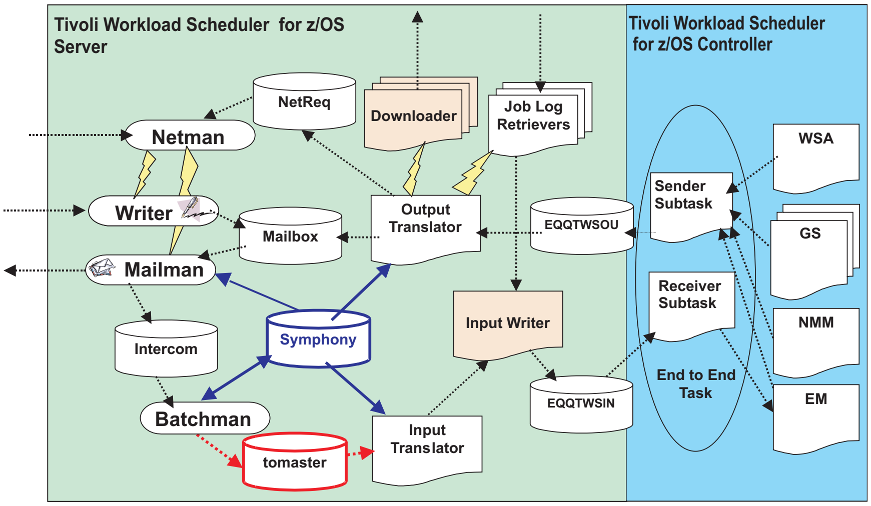


Figure 2. Tivoli Workload Scheduler for z/OS data sets and files for end-to-end scheduling with fault tolerance capabilities

The server address space hosts the tasks and the data sets that function as the intermediaries between the controller and the Tivoli Workload Scheduler domain manager at the other end of the network. In many cases these tasks and data sets are a replica of the Tivoli Workload Scheduler tasks and data sets.

The processes for end-to-end scheduling with fault tolerance capabilities are of two types, according to where they run:

- Controller processes. They are:
  - Sender
  - Receiver
- End-to-end server. They are:
  - Netman
  - Writer
  - Mailman
  - Batchman
  - Job log retriever
  - Output translator
  - Input translator
  - Input writer
  - Downloader

Netman

Replicates the Tivoli Workload Scheduler process. It starts at system startup. It monitors the NetReq.msg queue and the Tivoli Workload Scheduler TCP/IP port (usually 31111). When it receives a request, it starts the Writer or Mailman processes. The request to start or stop Mailman will come from the Output Translator via the NetReq.msg queue. The request to start or stop Writer will come from Mailman on the Tivoli Workload Scheduler domain manager via the TCP/IP port.

Writer Replicates the Tivoli Workload Scheduler process. It is started by Netman on request from the Mailman of the connected Tivoli Workload Scheduler domain manager. Writer has the task of writing the events that it receives from the remote Mailman in Mailbox.msg.

Mailman

Replicates the Tivoli Workload Scheduler process. Its main tasks are:

- Routing events. It reads the events stored in the Mailbox.msg queue and sends them either to the controller, writing them in Intercom.msg, or to the remote Writer on the Tivoli Workload Scheduler domain manager.
- Establishing the connection with the domain manager by calling the remote Netman to start Writer.
- Sending the Symphony to the other Tivoli Workload Scheduler nodes when a new Symphony is created.

If defined in the topology, Mailman can start other Mailman processes, called Mailman servers, to manage communication with specific workstations in the MASTERDM domain.

Batchman

Updates the Symphony file and resolves dependencies at master level. It replicates the functionality of Tivoli Workload Scheduler Batchman to a limited extent (submission has been removed).

Job Log Retriever

Receives from each distributed agent the log of a job run on the agent. After the Job Log Retriever has received the log, it sizes the log according to Tivoli Workload Scheduler for z/OS specifications, translates it from UTF-8 to the EBCDIC codepage, and sends it to the Input Writer. The retrieval of a job log is a lengthy operation and users may request several logs at the same time. For this reason, a subtask is started for each job log retrieval. The subtasks are temporary and terminate after the logs are sent to the Input Writer.

Output Translator

Receives the events in Tivoli Workload Scheduler for z/OS format from the outbound queue and elaborates them to activate the correct Tivoli Workload Scheduler function. It also translates event names from the EBCDIC codepage to UTF-8.

The Output Translator interacts with three different components depending on the type of the event:

- Starts a Job Log Retriever thread if the event is to retrieve the log of a job from a Tivoli Workload Scheduler distributed agent.
- Starts a Downloader thread if the event is to download the script.
- Enqueues an event in NetReq.msg if the event is to start or stop Mailman.
- Enqueues events in Mailbox.msg for the other events that are sent to update the Symphony file on the distributed agents (i.e. events for a job that has changed status, events for manual changes on jobs or workstations, or events to link/unlink workstations).
- Switches the Symphony files.

The Output Translator uses the Tivoli Workload Scheduler libraries to create the events and to enqueue them on the event files.

Input Translator
Translates the events read from the tomaster.msg to the Tivoli Workload Scheduler for z/OS format, including UTF-8 to EBCDIC translation, and writes them in the inbound queue.

Sender Task
Is a subtask of the end-to-end scheduling with fault tolerance capabilities task. It receives events from all the controller tasks that make changes in the control program and enqueues the events in the outbound queue. The events are received via SSI that is the usual method the Tivoli Workload Scheduler for z/OS tasks use to exchange events. The tasks that change the control program and that send events to the Sender Task are:

- General service (GS)
- Normal mode manager (NMM)
- Event manager (EM)
- Workstation analyzer (WSA)

The NMM sends events to the Sender Task during the daily planning for synchronization purposes.

Receiver Task
Is a subtask of the end-to-end scheduling with fault tolerance capabilities task. It receives events from the inbound queue and enqueues them to the Event Manager. The events have already been filtered and elaborated by the Input Translator.

Input Writer
Receives the input from Job Log Retriever and Input Translator and writes it in the inbound queue (EQQTWSIN).

Downloader
Sends scripts to remote agents. A thread is started for every script that must be downloaded and ends after the script has been downloaded.

Tivoli Workload Scheduler for z/OS uses the following data sets for end-to-end scheduling with fault tolerance capabilities:

- Data sets on z/OS:

  EQQTWSIN
  Sequential data set used to queue events sent by the server to the controller (inbound queue).

  EQQTWSOU
  Sequential data set used to queue events sent by the controller to the server (outbound queue).

  EQQSCLIB
  Partitioned data set used as a repository for the definitions of the jobs running on distributed agents. This data set is not shown in Figure 2 on page 41.

  EQQSCPDS
  VSAM data set containing a copy of the current plan used by the daily plan batch programs to create the Symphony file. This data set is not shown in Figure 2 on page 41.

  EQQTWSCS
  Sequential data set used to temporarily store a script when it is downloaded from the JOBLIB data set to the agent for its submission. This data set is not shown in Figure 2 on page 41.

- Data sets on USS. The following filets are in the work directory which is defined through the TOPOLOGY statement.

  Symphony
  File containing the active copy of the plan used by the distributed agents. The daily planning batch job creates the Symnew file. The Translator renames Symphony in Symold and Symnew in Symphony. The Translator creates also the Sinfonia file that contains the copy of the Symphony file used by the distributed agents.

  Sinfonia
  File containing the distribution copy of the plan used by the distributed agents.

  NetReq
  File used to queue requests for the Netman process.

  Mailbox
  File used to queue events sent to the Mailman process.

  Intercom
  File used to queue events sent to the Batchman process.

  Tomaster
  File used to queue events sent to the Translator process.

Data sets used for data store

The Data Store SYSOUT database consists of VSAM and unstructured data files and of primary and secondary indexes. They are described in the following table:

Table 3. The data store data sets

Database	Physical data sets	DD name
Structured data files	Data file (01-99)	EQQSDFxx
Primary index	One primary index for each Data Store	EQQPKIxx
Unstructured data files	Data file (01-99)	EQQUDFxx
Secondary index	One secondary index for each Data Store	EQQSKIxx

Structured data files

The structured data files contain joblog SYSOUTs in a form based on the parsing of the three components of the job log, the JESJCL, the JESYSMSG and the JESMSGLG, especially the first two. User SYSOUTS are excluded from the structuring mode. These files are required.

Unstructured data files

The unstructured data files contain joblog SYSOUTs in a flat form, as provided by the JES spool. You can check the SYSOUT with the BROWSE JOBLOG function. Note that the unstructured data file can store, if requested, also the user SYSOUTs. The activation of the unstructured data files is optional, depending on appropriate data store parameters.

Primary index

This contains an entry for each SYSOUT whose data is stored in one of the data files.

Secondary index

The secondary index is a variable-length key-sequenced data set (KSDS). Because it can be a single record, that corresponds to a specific secondary-key value, it can trace many primary keys.

Tivoli Workload Scheduler for z/OS batch programs

The controller includes a set of batch programs for tasks that are not suitable for online processing. The batch programs are submitted from the dialog, using predefined JCL. Before submission, the JCL is modified from the Tivoli Workload Scheduler for z/OS dialog according to user specifications. Some of the batch programs use services provided by the Tivoli Workload Scheduler for z/OS address space.

Tivoli Workload Scheduler for z/OS includes the following batch-program load modules that are called by EQQBATCH:

EQQADCOP
Calculate and print the run dates of an application

EQQADDEP
Produce a cross-reference listing between applications and their external predecessors and successors

EQQADMUP
Mass update the application descriptions

EQQADPRT
Print detailed application descriptions

EQQAXR00
Produce a cross-reference listing for applications

EQQCLPRP
Print the periods

EQQCLPRT
Print the calendars

EQQDNTOP
Produce (extend or create) a new current plan

EQQDOTOP
Print the current plan statistics

EQQDPCOP
Produce a copy of the AD, LTP, RD, SI, and WS VSAM data sets that can be used to produce a trial current plan

EQQDRTOP
Replan all values in the “current” current plan

EQQDSTOP
Renews the Symphony file

EQQDTTOP
Produce a trial current plan

EQQEVPGM
Report events

EQQICTOP
VSAM data set migration program

EQQJVPRT
Print JCL variables

EQQLTCRE
Create the long-term plan

EQQLTMOA
Modify all occurrences in the long-term plan

EQQLTMOO
Modify one application in the long-term plan

EQQLTPRT
Print the long-term plan

EQQLTTRY
Produce a trial long-term plan

EQQOIBAT
Print operator instructions

EQQOIBLK
Mass update operator instructions

EQQPDLF
Purge a data lookaside facility (DLF) object

EQQPURGE
Process DLF purge JCL

EQQWSPRT
Print the workstation descriptions

EQQYLTOP
Create application descriptions and operator instructions

EQQYTOPX
The program interface top module.

Tivoli Workload Scheduler for z/OS also includes the cataloged procedure **EQQRCSIM**, which is used for simulating return codes with step-level restart.

For more information about the batch programs, see *Managing the Workload*.

The Tivoli Workload Scheduler for z/OS system

This section describes the major subcomponents of the Tivoli Workload Scheduler for z/OS subsystem and their functions.

Subsystem early code - EQQINITL

Each z/OS subsystem builds a subsystem vector table (SSVT) control block that defines what functions are supported by that subsystem. The Tivoli Workload Scheduler for z/OS SSVT is built by a separate module, EQQINITL, which is invoked during the master scheduler initialization at IPL. EQQINITL does the following:

- Sets the UPSS bit in the (SSVT) so that Tivoli Workload Scheduler for z/OS can be started under JES
- Builds common storage area (CSA) control blocks to queue events until Tivoli Workload Scheduler for z/OS is started
- Builds a subsystem extension (SSX) block and chains it to the SSVT

- Builds an SSVT and chains it to the subsystem communication vector table (SSCVT).

The SSVT that EQQINITL builds specifies that Tivoli Workload Scheduler for z/OS supports two subsystem calls:

- A request for service from a Tivoli Workload Scheduler for z/OS dialog (type 67)
- A notification about a tracking event (type 68).

The code that is required to support these two subsystem calls is contained in a separate load module, EQQSSCMJ. This module is loaded in ECSA by EQQINITL processing at IPL. It can be temporarily replaced when the Tivoli Workload Scheduler for z/OS address space is started using the SSCMNAME keyword of OPCOPTS, see *Customization and Tuning*.

Tivoli Workload Scheduler for z/OS interface component

The Tivoli Workload Scheduler for z/OS subsystem interface component consists of:

- The Tivoli Workload Scheduler for z/OS SSVT. The SSVT is built by the EQQINITL module during the master scheduler initialization and is located in key zero, CSA storage (subpool 241).
- The modules that implement the passing of a request from the caller address space to the Tivoli Workload Scheduler for z/OS address space. These routines are located in the EQQSSCMJ module.

An SSVT can support up to 256 function calls. The Tivoli Workload Scheduler for z/OS SSVT supports two function calls. The first call is issued by Tivoli Workload Scheduler for z/OS dialogs; the second is used by event-tracking exits.

z/OS and JES exits supplied with Tivoli Workload Scheduler for z/OS request service by directing the SSI request to the MSTR subsystem. These function calls cause the master subsystem to invoke all defined subsystems.

Requesting Tivoli Workload Scheduler for z/OS service

Requests for Tivoli Workload Scheduler for z/OS subsystem services from these sources are passed to the product using the z/OS subsystem interface:

- Event-tracking routines. These are versions of standard z/OS and JES exits that are supplied by Tivoli Workload Scheduler for z/OS and that invoke services by issuing the EQQEXIT macro. The EQQEXIT macro generates code to invoke Tivoli Workload Scheduler for z/OS services correctly.
- TSO users running Tivoli Workload Scheduler for z/OS dialogs in an ISPF environment. The dialogs invoke Tivoli Workload Scheduler for z/OS services by calling the TSO service facility, which in turn calls the EQQMINOL module. The EQQMINOL module is APF-authorized and is able to use the z/OS subsystem interface to request service from Tivoli Workload Scheduler for z/OS.
- The Tivoli Workload Scheduler for z/OS server handles requests from remote dialog users or PIF applications, or from the GUI. The requests are forwarded to the subsystem via the EQQMINOL module, that uses the z/OS subsystem interface.
- The program interface, the batch loader, and the event generation program (EQQEVPGM).

Requests for Tivoli Workload Scheduler for z/OS subsystem services can also be made through the application programming interface (API).

As a result of calling the z/OS subsystem interface, the module EQQSSCMD is invoked. This module is loaded during the master scheduler initialization and resides in the link pack area (LPA) or CSA, depending on whether it is loaded from SYS1.LPALIB or SYS1.LINKLIB. The EQQSSCMD routine provides the interface to the Tivoli Workload Scheduler for z/OS address space.

If the service requester is an event-tracking routine, the EQQSSCMD module is called once for each Tivoli Workload Scheduler for z/OS subsystem that is defined. If the service requester is a Tivoli Workload Scheduler for z/OS dialog, the EQQSSCMD module is called only for one specific The subsystem determined by the subsystem name that the TSO user has defined in the dialog initialization panel.

If EQQSSCMD is called for a job-tracking event, an exit event record is built and added to a Tivoli Workload Scheduler for z/OS queue that is serviced by the event writer (EW) subtask. (This subtask is described in "Event writer (EW)" on page 63. The job-tracking caller does not wait for service from the event writer. Instead, the job-tracking caller returns after updating the EW queue.

If EQQSSCMD is called for a dialog service request, an entry is added to a Tivoli Workload Scheduler for z/OS queue that is serviced by the general service (GS) subtask. (This subtask is described in "General service (GS)" on page 66.) If a GS subtask is active, the request is dequeued and processed. The dialog user caller waits for service from the GS subtask. When the service is performed, the GS subtask posts the waiting dialog. The EQQSSCMD module then returns to EQQMINOL, which in turn returns to the Tivoli Workload Scheduler for z/OS dialog.

Subsystem address space - EQQMAJOR

Tivoli Workload Scheduler for z/OS is a z/OS started task that is implemented by the EQQMAJOR load module. To run successfully, this load module must reside in an APF-authorized library. It is link-edited with the AC(1) and AMODE(31) attributes.

EQQMAJOR performs the following tasks:

- Address space initialization:
  - Set up ESTAE recovery.
  - Establish an address-space-level and TCB-level resource manager.
  - Process initialization file.
  - Open all required files.
  - Update the SSX with Tivoli Workload Scheduler for z/OS TCB and ASCB addresses to indicate (to Tivoli Workload Scheduler for z/OS-subsystem-interface routines) that the product is active.
- Address space termination:
  - Clear Tivoli Workload Scheduler for z/OS TCB and ASCB addresses from the SSX to indicate (to Tivoli Workload Scheduler for z/OS-subsystem-interface routines) that the product is not active.
  - Close all open files.
  - Cancel ESTAE recovery.
- Address space recovery:
  - Create diagnostic information.
  - Determine the retry routine to be used.
  - Return to RTM via SETRP retry.

The subsystem job-step task (EQQMAJOR) builds the Tivoli Workload Scheduler for z/OS environment. EQQMAJOR invokes the subsystem task dispatcher, EQQZMAIN. The majority of Tivoli Workload Scheduler for z/OS functions are performed by subtasks to EQQZMAIN. EQQZMAIN attaches these subtasks to run the various subfunctions of the subsystem. The initialization parameters that are passed to Tivoli Workload Scheduler for z/OS (from the parameter library, EQQPARM) determine which subtasks will be activated.

Subsystem task dispatcher - EQQZMAIN

After Tivoli Workload Scheduler for z/OS is started and initialization is complete, the subsystem task waits for more work. EQQZMAIN, the subsystem task dispatcher, then performs one of the following:
- Attaches subtasks (if required)
- Requests subtasks to terminate (if Tivoli Workload Scheduler for z/OS is stopping)
- Returns to caller (if no events are pending)
- Invokes the wait routine.

Tivoli Workload Scheduler for z/OS monitors its subtasks by referring to a table that has one entry for each possible subtask. Each entry in this table contains:
- The name of the task
- The name of the load module to be given control
- The TCB address of the subtask
- The ECB that each Tivoli Workload Scheduler for z/OS will post when the subtask is to terminate
- Status information.

All subtasks are expected to terminate only when their stop-ECBs are posted. If a subtask has ended unexpectedly, Tivoli Workload Scheduler for z/OS checks the corresponding entry in the subtask table to determine if the task should be reattached. If so, the task is restarted.

If Tivoli Workload Scheduler for z/OS is requested to stop, all the active subtasks are posted to stop processing. The product then waits for the subtasks to end.

Message-writer routine

Tivoli Workload Scheduler for z/OS passes the address of a Tivoli Workload Scheduler for z/OS communication area (the MCA block) to each subtask it attaches. The MCA contains data shared by subtasks and the addresses of common subroutines.

One of these common routines is the message-writer routine. All Tivoli Workload Scheduler for z/OS subtasks use this routine to issue messages to the operator and to the Tivoli Workload Scheduler for z/OS message-log file.

Server address space - EQQSERVR

The Tivoli Workload Scheduler for z/OS server is a z/OS started task that is implemented by the EQQSERVR load module. To run successfully, this load module must reside in an APF–authorized library. It is link–edited with the AC(1) and AMODE(31) attributes.

The server must run on the same z/OS system where the served controller is started.

EQQSERVR performs the following tasks:

- Address space initialization:
  - Set up ESTAE recovery
  - Process initialization file
  - Attach the main server task
- Address space termination:
  - Detach the main server task
  - Cancel ESTAE recovery
- Address space recovery:
  - Create diagnostic information
  - Determine the retry routine to be used
  - Return to RTM via SETRP retry

The server job–step task (EQQSERVR) attaches the main server task (EQQPHTOP), and waits for operator commands. All the server processing is performed by the EQQPHTOP and its subtasks.

Main server task - EQQPHTOP

This subtask is attached by EQQSERVR when initialization is complete. The major tasks performed by EQQPHTOP are:

For APPC:

- Initialize to APPC as scheduler
- Wait for one of the following events to occur:
  - An allocation request is received from APPC for a new connection. EQQPHTOP attaches an EQQPHAPP subtask to handle this connection.
  - The stop ECB is posted. In this case, EQQPHTOP will propagate the termination to all the subtasks by posting their stop ECB, will terminate itself as an APPC scheduler, and will return to the caller.

For TCP/IP:

- Initialize to TCP/IP as server
- Respond when a request is received for a new TCP/IP connection (from ISPF, Dynamic Workload Console, or PIF). EQQPHTOP calls the EQQPTTOP subtask to handle this connection. EQQPTTOP is linked to the EQQZTSER process, which includes a main thread that listens for incoming requests and in response to every request for connection opens a new thread.

For end-to-end scheduling with fault tolerance capabilities:

- Start the EQQPWAPP task to communicate with the distributed network, if the TPLGYPARM keyword was specified among the server parameters in the SERVOPTS statement,

The EQQPHTOP task ends when all the subtasks ended.

Data store address space – EQQFARCH

The Tivoli Workload Scheduler for z/OS data store is a z/OS started task that is implemented by the EQQFARCH load module. To run successfully, this load module must reside in an APF-authorized library. It is link-edited with the AC(1) and AMODE(31) attributes.

EQQFARCH is the top module of the Data Store Started Task Control. It basically loads and calls the EQQFARMJ module, that performs the following tasks:

- Initialization of Data Store parameters

- Initialization, synchronization, and termination of all Data Store subtasks. In particular for the synchronization function, EQQFARMJ builds an event table to manage wait/post requests which synchronize the different subtask activity.

Refer to “Tivoli Workload Scheduler for z/OS data store task structure” on page 56 and “Tivoli Workload Scheduler for z/OS data store subtasks” on page 80 for more information about the subtask functions and their interaction.

Tivoli Workload Scheduler for z/OS task structure

The task dispatcher, EQQZMAIN, waits for one of the following events to occur:

- A stop or a modify command entered by the z/OS operator. If an operator stop command is received, all the subtasksare posted to terminate.
- A message created by a subtask. When messages are created by Tivoli Workload Scheduler for z/OS subtasks, they are formatted and written to the message log data set.
- Subtask termination.

The subsystem job-step task ends when all subtaskshave ended.

Figure 3 on page 52 shows the tasks that can be started as subtasks by the job-step task when Tivoli Workload Scheduler for z/OS is installed.

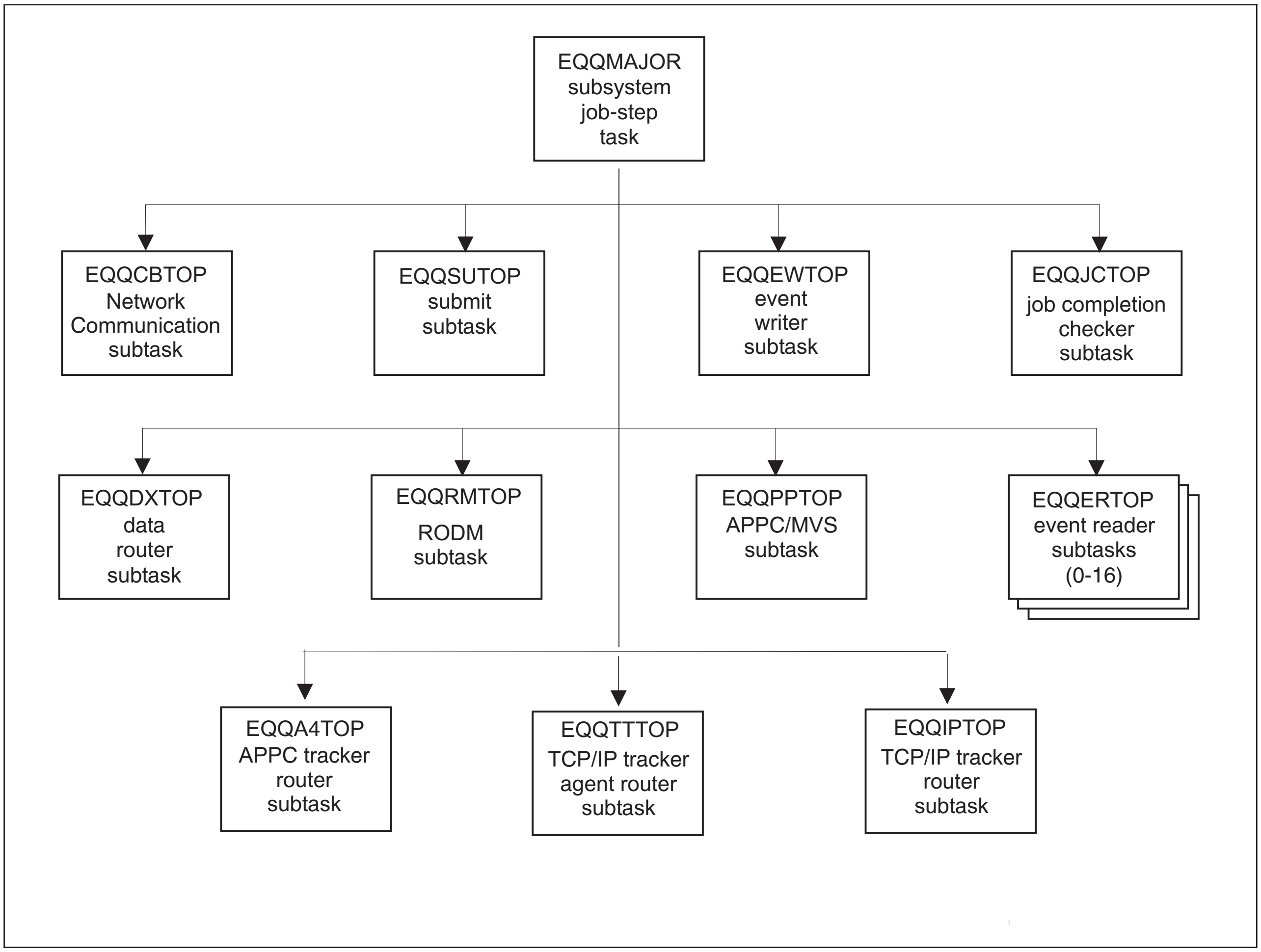

EQQMAJOR subsystem job-step task
EQQCBTOP Network Communication subtask
EQQSUTOP submit subtask
EQQEWTOP event writer subtask
EQQJCTOP job completion checker subtask
EQQDXTOP data router subtask
EQQRMTOP RODM subtask
EQQPPTOP APPC/MVS subtask
EQQERTOP event reader subtasks (0-16)
EQQA4TOP APPC tracker router subtask
EQQTTTOP TCP/IP tracker agent router subtask
EQQIPTOP TCP/IP tracker router subtask

Figure 3. Tivoli Workload Scheduler for z/OS subsystem tasks

If the controller is installed on the system, the normal-mode-manager (NM) subtask will be attached. The NM subtask will attach the tasks that reference the current plan as subtasks. These tasks are shown in Figure 4 on page 53.

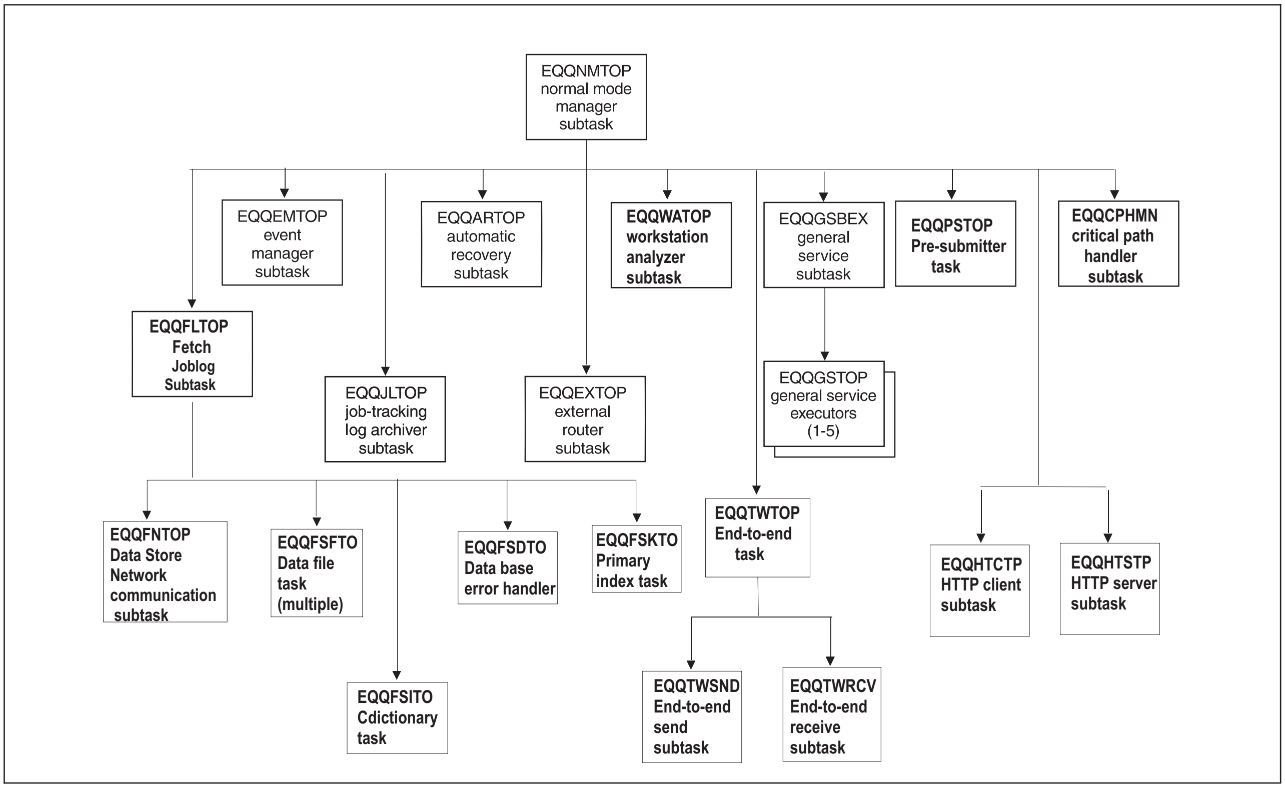

EQQNMTOP normal mode manager subtask
EQQEMTOP event manager subtask
EQQARTOP automatic recovery subtask
EQQWATOP workstation analyzer subtask
EQQGSBEX general service subtask
EQQPSTOP Pre-submitter task
EQQCPHMN critical path handler subtask
EQQFLTOP Fetch Joblog Subtask
EQQJLTOP job-tracking log archiver subtask
EQQEXTOP external router subtask
EQQGSTOP general service executors (1-5)
EQQFNTOP Data Store Network communication subtask
EQQFSFTO Data file task (multiple)
EQQFSDTO Data base error handler
EQQFSKTO Primary index task
EQQTWTOP End-to-end task
EQQHTCTP HTTP client subtask
EQQHTSTP HTTP server subtask
EQQFSITO Cdictionary task
EQQTWSND End-to-end send subtask
EQQTWRCV End-to-end receive subtask

Figure 4. controller subtasks

Tivoli Workload Scheduler for z/OS server task structure

The main server task, EQQPHTOP, is attached by main task EQQSERVR. If you are using the APPC communication protocol, for each new conversation a new instance of EQQPHAPP subtask is attached. If you are using the TCP/IP communication protocol, for each new connection a new thread is started by process EQQZTSER (which depends on task EQQPTTOP). The EQQPWAPP subtask is attached for end-to-end scheduling with fault tolerance capabilities. It triggers the server processes for end-to-end scheduling. Figure 5 on page 54 shows the task structure of the Tivoli Workload Scheduler for z/OS server address space.

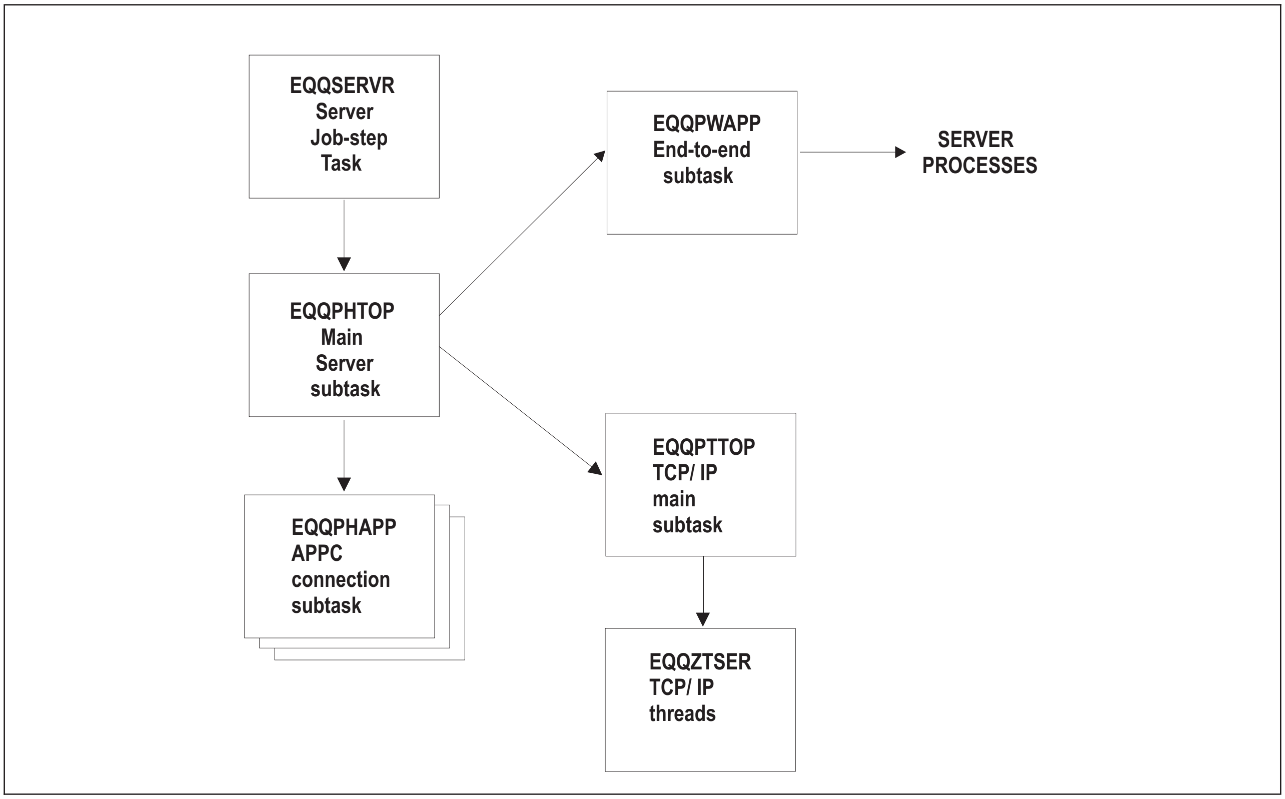


Figure 5. Tivoli Workload Scheduler for z/OS server tasks

Figure 6 on page 55 provides a graphical summary of how process EQQZTSER works. The main thread runs a Listener thread, which starts a new communication thread for each incoming TCP/IP connection request it receives. A request for connection can come from any of the available user interfaces: PIF, OCL, BCIT, ISPF dialogs, or Dynamic Workload Console. Note that:

- Any communication thread that stays idle for more than five minutes is automatically closed the end of this time.
- The maximum number of threads that can be open at any moment is given by your z/OS system configuration.

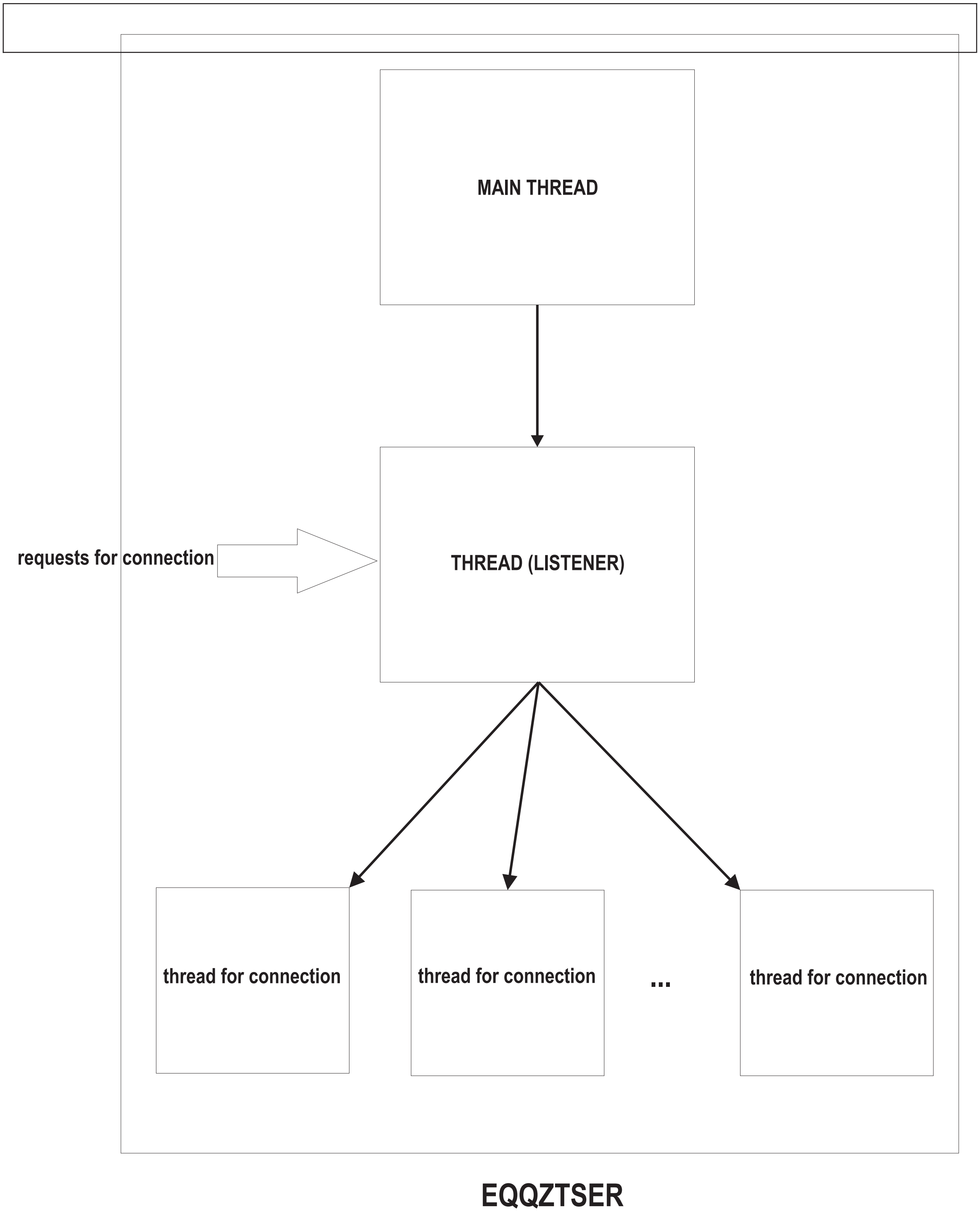


Figure 6. Internals of the EQQZTSER process.

Figure 7 on page 56 shows the end-to-end server processes started by the EQQPWAPP subtask.

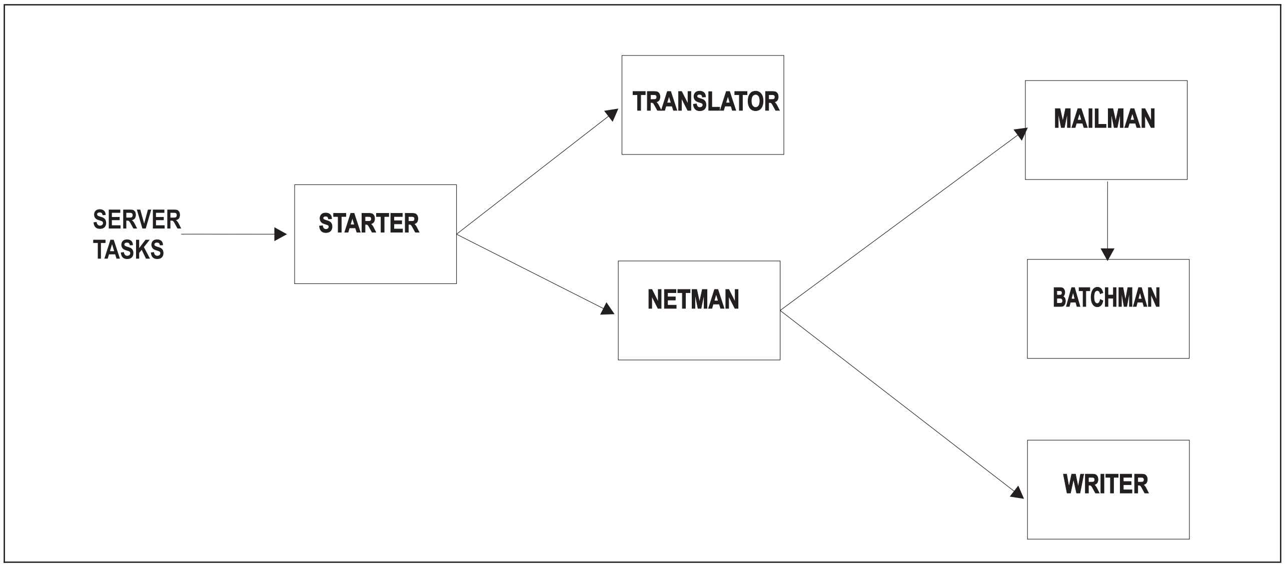


Figure 7. Tivoli Workload Scheduler for z/OS server processes for end-to-end scheduling with fault tolerance capabilities

Tivoli Workload Scheduler for z/OS data store task structure

When EQQFARCH is started, EQQFARMJ initializes and activates the main subtasks and initializes the database. Depending on the number of data files referenced in the startup JCL, different data file subtasks are also activated. Figure 8 on page 57 shows the task structure of the Tivoli Workload Scheduler for z/OS Data Store address space.

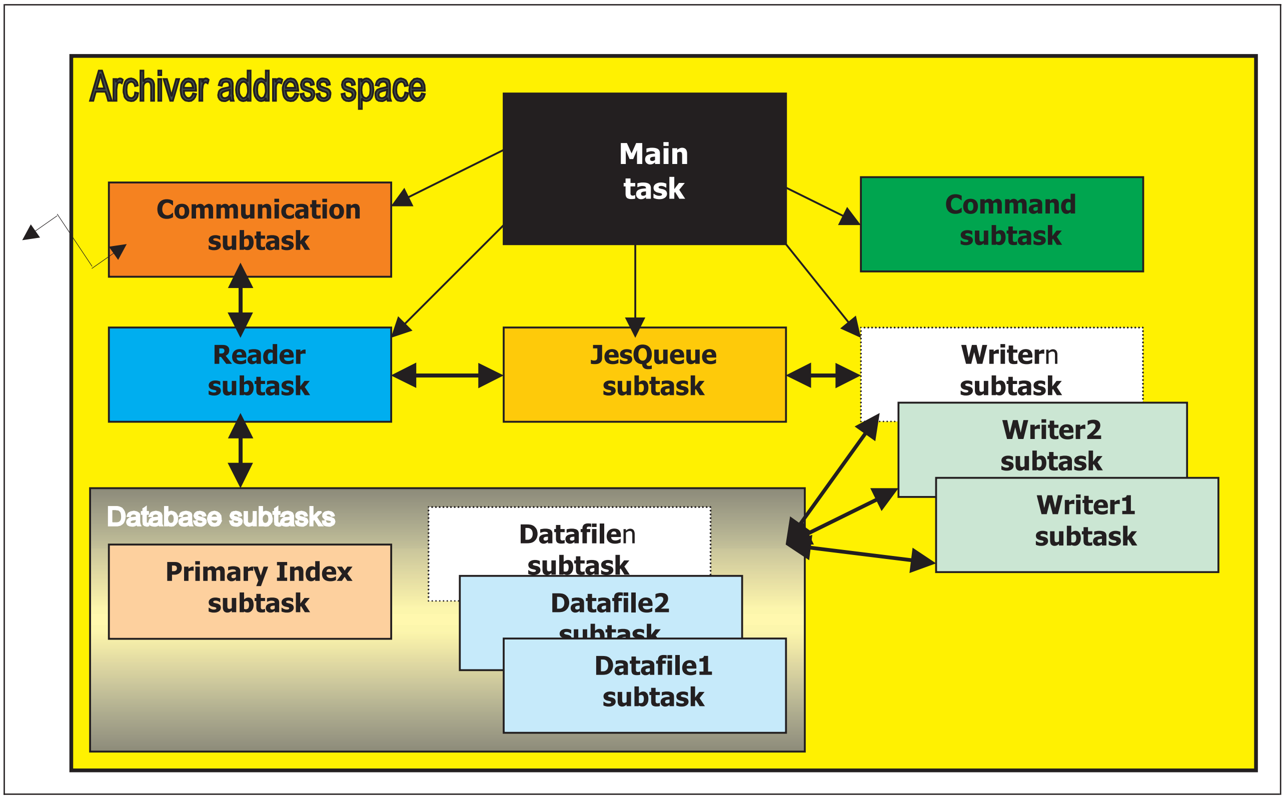


Figure 8. Tivoli Workload Scheduler for z/OS data store address space structure

Tivoli Workload Scheduler for z/OS subtasks

This section describes the tasks that can be started as subtasks to the Tivoli Workload Scheduler for z/OS subsystem.

APPC/z/OS (PP)

The APPC/z/OS (PP) subtask lets you connect to programs running on any Systems Application Architecture® (SAA) platform, and any other platforms that conform to CPI-C protocol, which support Advanced Program-to-Program Communications (APPC). APPC is the term used to refer to an implementation of Systems Network Architecture (SNA) for logical unit (LU) type 6.2.

The application programming interface (API) is provided to enable your application transaction program (ATP) to establish a *conversation* with the Tivoli Workload Scheduler for z/OS subsystem. The API uses a subset of Common Programming Interface for Communications (CPI-C) calls, send and receive buffer mappings, and a call-sequence protocol. Your programs access the Tivoli Workload Scheduler for z/OS address space data by providing calls to the APPC API. You can regard the set of APPC verbs as a programming language that you can write conversations in. The Tivoli Workload Scheduler for z/OS API lets your own programs:

- Extract information about the current plan (GET request)
- Update or add current-plan operations (PUT request)
- Delete operations in the current plan (DEL request)
- Report events to Tivoli Workload Scheduler for z/OS (CREATE request).

For more information about the API, refer to *Driving Tivoli Workload Scheduler for z/OS*. For more information about CPI-C, see *Common Programming Interface: Communications Reference* in the Systems Application Architecture (SAA) library.

Tivoli Workload Scheduler for z/OS is an APPC transaction scheduler; the scheduler name is the same as the subsystem name. By associating the scheduler with a logical unit in the APPCPM*nn* member of SYS1.PARMLIB, all allocate requests directed to this LU are passed to the subsystem. Tivoli Workload Scheduler for z/OS then schedules a partner transaction program (TP) in its own address space. The originating TP issues a request and waits for the complete reply, before issuing a new request or deallocating the conversation.

Tivoli Workload Scheduler for z/OS recognizes these TP names:

EQQTRK
Supplied by trackers that communicate with the controller through APPC

EQQAPI
Supplied by user programs (ATPs) that communicate with Tivoli Workload Scheduler for z/OS through the API.

The APPC/z/OS subtask is attached by the subsystem task. The subtask creates a new queue and table with one entry for each conversation. This entry contains data received from APPC about this conversation and parameter list areas, including the event control block (ECB), used for all asynchronous requests.

APPC/z/OS and Tivoli Workload Scheduler for z/OS interface in this way:

- The queue server contains a *work-to-do* loop, which waits on an ECB list. Initially, this ECB list contains a stop-ECB and a queue-ECB. The Tivoli Workload Scheduler for z/OS tracker attaches and detaches APPC/z/OS.
- Inbound allocation messages are placed on the APPC queue (APPQ) by the APPC cross-system coupling facility (XCF) message exit. When this message is detected on the queue, a new entry is added to the conversation table, and the ECB is added to the ECB list.
- The subtask interfaces with the system security product through SAF RACROUTE calls, using the security token and local LU name. If the user does not have read or update access, the request is rejected immediately and the conversation deallocated with `security_invalid`. Tivoli Workload Scheduler for z/OS checks security for every request to ensure that authority has not been changed.
- Tivoli Workload Scheduler for z/OS places inbound requests GET, PUT, or DEL, on the general services (GS) queue, or calls the subsystem interface (SSI) for CREATE requests. Tivoli Workload Scheduler for z/OS issues receive requests until state_received=send, before placing a request block on the GS queue or calling the SSI. All other entries in the table are checked for posted ECBs and the program then returns to a WAIT state.
- For GET, PUT, or DEL requests, GS places the requested data on the APPC queue (APPQ). In case of error, GS generates an error return code and reason code.
- APPC builds an outbound APP buffer. When all requested data is sent, the program returns to a WAIT state. Processing continues until the subtask is stopped or the conversation ends (state_received=deallocate_normal). The entry in the table is deleted and the ECB list is rebuilt. This also happens if severe errors occurring during the conversation.

test

If APPC fails, Tivoli Workload Scheduler for z/OS deallocates the conversation and deletes the entry in the table. You are informed of this situation, and should mask this event from your end-user by initiating a new *allocate* before sending a new request. If Tivoli Workload Scheduler for z/OS is stopped, all conversations are deallocated.

Tivoli Workload Scheduler for z/OS supports many conversations. The table is retained across APPC/z/OS subtask sessions. Requests outstanding at the time APPC/z/OS is closed are deallocated.

Automatic recovery (AR)

The automatic recovery (AR) subtask handles automatic recovery requests. A z/OS job, a started-task procedure, or a job using the centralized script can contain recovery statements that specify an error situation and the recovery actions to be performed if the job or procedure fails during processing.

The AR subtask scans the failing jobs or started tasks for a recovery statement that matches the particular error. If a matching recovery statement is found, the requested recovery actions are performed. When a recovery statement matches an error condition it is changed by Tivoli Workload Scheduler for z/OS to a JCL comment statement.

RECOVER statements are defined in the job input as z/OS statements. The statement is changed when the operation is submitted to a valid comment statement for the target operating system. The image of the job input in the JOBLIB is not altered.

The following input is needed for the AR subtask:

- The job-name-table record of the current plan. When the job or procedure is run, this record saves error information, the completion codes of each step executed, and the job completion code.
- The ended-in-error record of the current plan. When a job or started-task procedure ends in error and an ended-in-error entry is created, an indicator is set in this entry. This indicator shows that the JCL or procedure should be checked for automatic recovery statements. The indicator can also be set at a later time from the HANDLING OPERATIONS ENDED IN ERROR panel in the MODIFY CURRENT PLAN dialog. The AR subtask resets the indicator when you select the entry for processing.
- The JCL repository (JS) data set. The AR subtask gets the JCL and automatic recovery statements for the failed job from this file.
- An event control block (ECB) in the normal-mode-manager control block NMMARECB. When this ECB is posted, AR investigates the ended-in-error records of the current plan for automatic recovery requests.
- Options of the AROPTS initialization statement. Using these options, you can restrict when an automatic recovery should be performed.
- The case code macro EQQCASEC.
- A list of case code definitions created by EQQCASEC.

AR invokes other components of the Tivoli Workload Scheduler for z/OS subsystem. The logical-file-handler component is called to accomplish the VSAM input/output operations that are required. The modify-current-plan function of the Tivoli Workload Scheduler for z/OS subsystem is called to perform the actions specified in the recovery statement.

Critical path handler (CPH)

The critical path handler (CPH) subtask updates the critical job table, which is a table of storage control blocks, containing an entry for each critical job and each predecessor to a critical job. The scheduler allocates the critical job table in a data space, as it does for the special resources control blocks.

Only the CPH subtask can update the critical job table, using a queue driven mechanism. Any task that requests to update the critical job table adds an element to the queue named CPHQ. The CPH subtask serves this queue, by using an internal timer to wake up periodically and check the critical job table data: the process recalculates a critical path every time a predecessor of the target operation starts delaying for one of the following reasons:

- It is late, meaning that it did not start within its latest start time.
- It is long running, meaning that it is running longer than its estimated duration.
- It ends with an error.

If the critical path changed, the scheduler updates the current plan.

Data router (DX)

Tivoli Workload Scheduler for z/OS subtasks communicate with each other within the system and among systems. The data router (DX) subtask routes data from the components that send data to those that receive data. The following Tivoli Workload Scheduler for z/OSsubtasks send data for processing to other subtasks via the DX subtask:

- Event reader subtask (sends events)
- Event writer subtask (sends events)
- Workstation analyzer subtask (sends JCL, commands, release requests, and WTO messages)
- Network communication function (CB) subtask.

The following Tivoli Workload Scheduler for z/OS subtasks receive data from the DX subtask:

- Event manager subtask (processes events)
- Submit subtask (submits JCL)
- Network communication function (CB) subtask.

When a sending component transmits data to a receiving component, the sending component puts the data on a router queue for processing by the DX subtask. The subtask can send the data to a component on the same Tivoli Workload Scheduler for z/OS system as the sending component or to a component on another Tivoli Workload Scheduler for z/OS system.

The DX subtask passes the data directly to the receiving component when the sending and receiving components reside on the same system. When the data is sent to another Tivoli Workload Scheduler for z/OS system, the DX subtask transmits the data to the DX subtask on that system. The receiving DX subtask then passes that data to the receiving component on that system. One of three methods is used to transmit data between systems:

- Shared DASD

- Cross-system coupling facility (XCF) communication links
- SNA communication links.
- TCP/IP communication link

Sometimes other components are involved in routing data between Tivoli Workload Scheduler for z/OS systems:

- When you are using SNA communication links, the network communication function (CB) performs the transmission.
- When data is transmitted via shared DASD, the DX subtask on the transmitting system writes the data to a submit/release data set. The event writer subtask on the receiving system reads the data and passes it to the router queue. The DX subtask processes the data and then passes it to the correct component for processing.
- When an operation is ready to be started on a workstation that specifies a user-defined destination, the DX subtask queues the request to the external router (EX) subtask. The EX subtask calls the operation-initiation exit, EQQUX009, to handle communications with the user-defined destination.

Event manager (EM)

The event manager (EM) subtask processes job-tracking, and user-created events, and updates the current plan accordingly. Event records are passed from a number of event writers via event reader subtasks, the network communication function, and XCF, and are passed to the EM queue via the data router subtask. The EM subtask calls a queue handler routine to retrieve events from the queue.Tivoli Workload Scheduler for z/OS produces tracking, submission, and synch events. Tracking events are preceded by the letter A (for JES2 complexes) or by the letter B (for JES3 complexes).

Note: Tivoli Workload Scheduler for z/OS does not track jobs with TYPRUN=SCAN.

The z/OS tracking events are:

1	Reader event
2	Start event
3S	Step-end events
3J	Job or started-task end event
3P	Job or started-task termination event
4	Print-end event
5	Purge event.

The submit events are:

IJ0	Submit synch event
IJ1	Submit job JCL
IJ2	Submit started-task JCL
IJ3	Submit standalone cleanup job
IWTO	Submit WTO message
IREL	Submit a release command

User-created events (created with the EQQUSIN*x* subroutines or the corresponding TSO commands) are sorted between jobs in event-creation order.

Events started by distributed agents are 0-type event records.

Most events processed by the queue handler subroutine are automatically created job-tracking events. To process the event, the subroutine:

- Updates the event reader position of the event data set that this event was read from.
- Checks if the event is for an operation defined in the current plan.
  - If the event is for an operation that can be matched with an operation in the current plan, the event manager processes the event by updating the current plan.
  - If the event is not for an operation that is controlled by Tivoli Workload Scheduler for z/OS but is a reader event, a start event, or a user event with status Q, the EM checks if the event matches an event-triggered tracking (ETT) trigger, if ETT is active.
  - If the event is not for an operation that is controlled by Tivoli Workload Scheduler for z/OS and the job is held by the Tivoli Workload Scheduler for z/OS subsystem (via the HOLDJOB parameter of the EWTROPTS initialization statement), EM releases the job from hold.
- Writes a record to the current job-tracking log if the event is for a Tivoli Workload Scheduler for z/OS operation and was successfully processed.
- If the event is for a Tivoli Workload Scheduler for z/OS submit synchronization response, the operations for the corresponding workstation are analyzed; if the submit requests are lost, the operations are flagged as requiring the submit requests to be resent, and the workstation analyzer is posted.
- If the event is for a Tivoli Workload Scheduler for z/OS job but cannot be processed immediately, the EM updates the job-tracking log and places the event on a suspend chain. Suspended events are examined by the EM every time the subtask gets control. If the event cannot be successfully matched against an operation in the current plan within 5 minutes, the event is discarded. If the event can be matched but is received out of sequence and is still out of sequence after 5 minutes, the event is processed and the operation might be set to ended in error status.

Subtask initialization consists of building the EM environment and opening all logical VSAM files that are required.

Subtask mainline logic consists of providing service for the following events:
- Timer expiration. Retry processing of suspended events.
- Subtask stop request.
- Add an event record to the event manager queue.

Subtask termination consists of closing all open files.

Event reader (ER)

The event reader (ER) subtask provides support for reading event records from an event data set. If new data arrives in this data set, it is read and passed to an in-storage queue (the event manager queue) via the data router subtask.

Subtask initialization consists of building control blocks for the ER subtask, reading and processing the parameter library member, and opening the input event data set.

Subtask mainline logic provides service for the following events:
- Timer expiration. Process the event data set.
- Subtask stop request. Return to caller.
- Reading suspend request. Start/stop reading from the event data set.

Up to 16 ER subtasks can be started for each Tivoli Workload Scheduler for z/OS subsystem.

Event writer (EW)

The event writer (EW) subtask provides support for writing event records to an event data set. An event record is built from job-tracking-exit records that are transmitted to the EW subtask via an in-storage queue, the WTRQ queue.

The EW subtask also regularly checks the contents of a submit/release data set. If new data arrives in this data set, the data is read and passed via the data router subtask to the submit subtask for processing. Data in the submit/release data set is either a release-job record or one or more submit-job records.

The EW subtask passes job and started-task termination (3P) events to the job completion checker (JC) subtask if the JC is active. The EW can also pass events to the data router queue (if the EWSEQNO parameter of the EWTROPTS initialization statement is used), and the EW subtask can act as an event reader.

Subtask initialization consists of building control blocks for the EW subtask, reading and processing the parameter library member, and opening the event data sets.

Subtask mainline logic provides service for the following events:
- Timer expiration. Process the submit/release data set.
- Subtask stop request. Return to caller.
- Process all events on the EW queue.

Subtask termination closes all open files and returns to the caller.

Event creation and processing

Figure 9 on page 64 shows the activities that can cause events to be created and how the events are processed by Tivoli Workload Scheduler for z/OS. The arrows show the flow of events among programs, central storage, and DASD storage. The flow of events is described with reference to the numbering on the diagram.

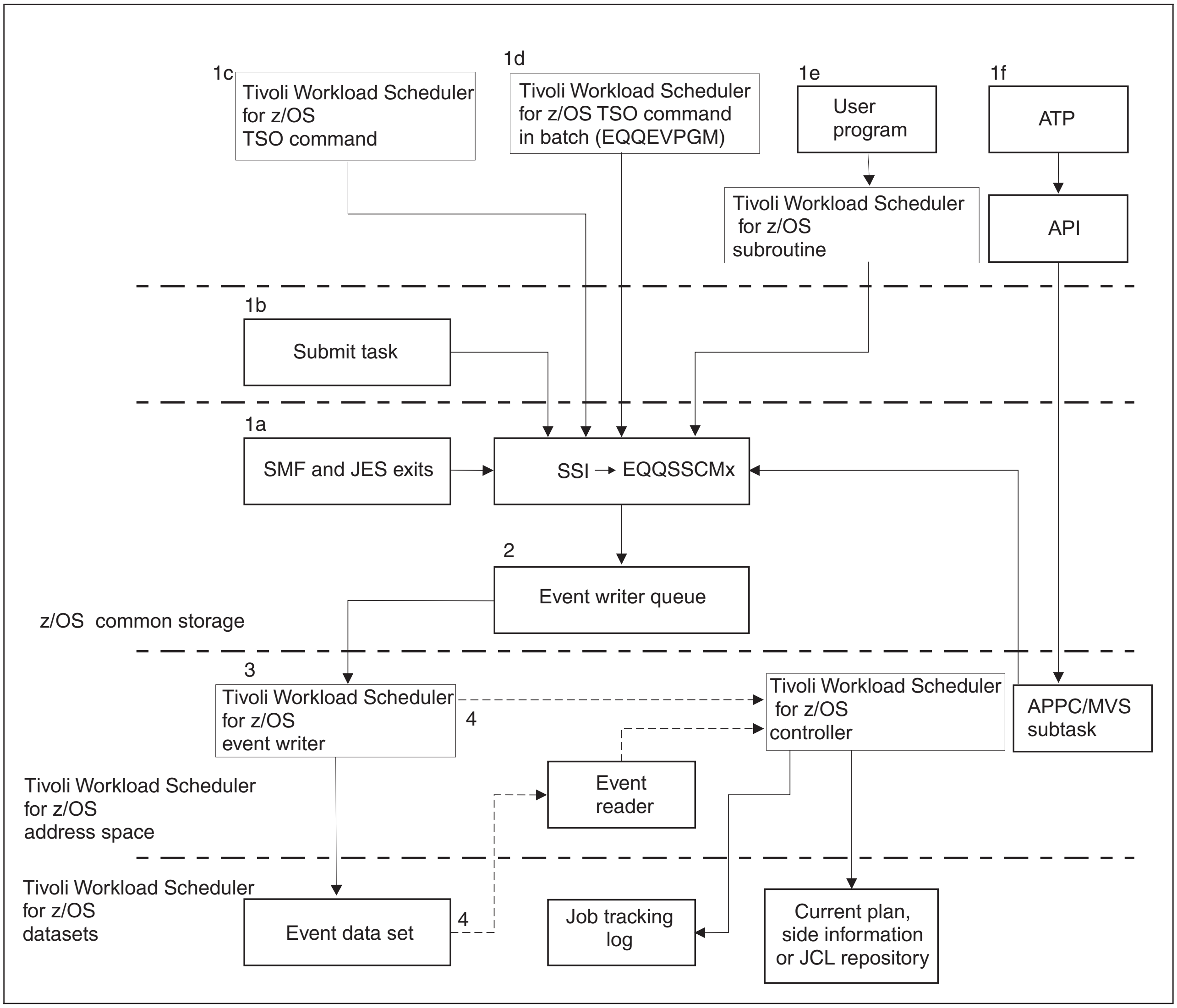

1c
Tivoli Workload Scheduler for z/OS TSO command
1d
Tivoli Workload Scheduler for z/OS TSO command in batch (EQQEVPGM)
1e
User program
1f
ATP
Tivoli Workload Scheduler for z/OS subroutine
API
1b
Submit task
1a
SMF and JES exits
SSI → EQQSSCMx
2
Event writer queue
z/OS common storage
3
Tivoli Workload Scheduler for z/OS event writer
4
Tivoli Workload Scheduler for z/OS controller
APPC/MVS subtask
Tivoli Workload Scheduler for z/OS address space
Event reader
Tivoli Workload Scheduler for z/OS datasets
Event data set
4
Job tracking log
Current plan, side information or JCL repository

Figure 9. Tivoli Workload Scheduler for z/OS event creation and processing

1. Event information is reported in one of these ways:
   a. z/OS calls the SMF and JES exits at certain stages in the life of a job. For example, the job initiation exit, IEFUJI, is called whenever a job starts. Tivoli Workload Scheduler for z/OS code in the exits collects relevant information about the event and passes it to the event-creation module, EQQSSCMJ, via the z/OS subsystem interface. Relevant information for a job that has started would include the name and number of the job, its starting date and time, and, if catalog management is active, data set information.
   b. All Tivoli Workload Scheduler for z/OS address spaces start a submit task. It initiates work on the system that the controller or tracker is started on and that represents the destination defined in the workstation description. When the submit task starts work, it uses EQQSSCMJ to create initialization events, depending on the work to be started. An IJ1 event is created for batch jobs, IJ2 for started tasks, IWTO for write-to-operator (WTO) operations, and IREL for release commands. Submit-checkpointing events (IJ0) are created for all work that Tivoli Workload Scheduler for z/OS submits, except operations that are routed to a user-defined destination ID.
   c. You provide information about the event as parameters to the BACKUP, OPINFO, OPSTAT, SRSTAT, or WSSTAT command, which is issued from the

TSO environment. The parameters are checked and then passed to the event-generation module, EQQSSCMJ, via the z/OS subsystem interface.

   d. The BACKUP, OPINFO, OPSTAT, SRSTAT, or WSSTAT command is run from a batch job, using the EQQEVPGM event-generating batch program. Parameters, which are input to EQQEVPGM through the SYSIN JCL statement, are checked and then passed to EQQSSCMJ.
   e. A user program provides information about the event in a parameter list and passes it to the EQQUSIN, EQQUSINB, EQQUSINS, EQQUSINO, EQQUSINW, or EQQUSINT subroutine. The subroutine checks the parameters and passes them to the event-generation module, EQQSSCMJ, via the z/OS subsystem interface.
   f. An application transaction program (ATP) passes a CREATE request to Tivoli Workload Scheduler for z/OS in an APP buffer through the application programming interface (API). The APPC subtask validates the buffer and then internally invokes the EQQUSIN subroutine.
2. The event-generation module, EQQSSCMJ, uses the information to build an event record and places the record in the event writer queue in ECSA.

   Note: Except for requests submitted through the API, the processing in these first two steps can take place as soon as the z/OS subsystem interface is started at IPL time. Tivoli Workload Scheduler for z/OS itself need not be active. If the product is not active (in particular, if the event writer subtask is not active), event records remain in the event writer queue until the event writer starts and processes them.

   A request submitted through the API must be passed to an active Tivoli Workload Scheduler for z/OS address space where the APPC subtask is started. If an event writer is not started in the same address space, the event must be broadcast.

   Event records are generated for *all* z/OS jobs and started tasks, even though they might not be relevant to a particular Tivoli Workload Scheduler for z/OS address space. It is not possible for the programs creating the event records to determine if a particular job is relevant to a particular Tivoli Workload Scheduler for z/OS address space. The event creation programs reside in z/OS common storage and do not belong to, or have access to, the data or resources of any Tivoli Workload Scheduler for z/OS address space that might be running on the same system or some other system.
3. The Tivoli Workload Scheduler for z/OS event-writer subtask of the tracker reads event records from the event writer queue and writes them to an event data set.
4. Events are transmitted to the controller by an event reader function. This is performed either by an event reader function of the event writer, or a separate event reader task. An event writer can use an XCF, NCF, or TCP/IP connection to transmit events to the controller. Where a separate event reader is used, the event reader can be active at the controller, or at a tracker that is connected to the controller via XCF, NCF, or TCP/IP. The event manager subtask that is started at the controller processes the events, and the relevant action is then taken by Tivoli Workload Scheduler for z/OS. If the event writer is active but no connection exists to the controller, or if the event reader is not active, events simply stay in the event data set until the required function is available.

Events are never lost, providing that the following two conditions are satisfied:

- The event writer queue in ECSA is large enough to hold all the event records that might be created while the event writer is not active.
- The event data set is large enough to hold all the event records that might be created while a connection to the controller is lost, or an event reader is not active.

Note: When Tivoli Workload Scheduler for z/OS is started and the BUILDSSX keyword of OPCOPTS has the value REBUILD, the event-writer queue from the old SSX (subsystem communication-vector-table (CVT) extension) is not referenced in the new SSX. See *Customization and Tuning* for more information about BUILDSSX.

External router (EX)

The external router (EX) subtask receives submit requests from the data router (DX) subtask when an operation is ready to be started at a computer automatic workstation that specifies a user-defined destination ID. The external router calls the operation-initiation exit, EQQUX009, to handle communications with the user-defined destination.

The EX subtask regularly checks for new requests on the EXA queue. When a new request arrives the EX subtask builds a parameter list and calls EQQUX009. If EQQUX009 abends or is not loaded, the EX subtask creates a workstation offline event for all user-defined destinations. In this case, the status of the operation that corresponds to the request is set according to the SUBFAILACTION keyword of the JTOPTS initialization statement. Subsequent submit request on the EXA queue are discarded and the corresponding operation status is set according to the WSOFFLINE keyword of JTOPTS.

The EXA queue allows for 32000 elements. If the queue becomes full the EX subtask issues a workstation offline event for all user-defined destinations. EQQUX009 must return control to the EX subtask before any subsequent queue elements are processed by the EX subtask. If the return code from EQQUX009 is not 0, the EX subtask sets operation status or generates workstation offline events as specified in the description of the operation-initiation exit in *Customization and Tuning*.

Subtask mainline logic provides service for the following events:
- Subtask stop request. Return to caller.
- Process all requests on the EXA queue and call EQQUX009.
- Generate workstation offline events in response to return code received from EQQUX009.

General service (GS)

The general service (GS) subtask services a queue of requests from the dialogs, batch command interface tool (BCIT), batch loader (BL), control language (OCL), Dynamic Workload Console, and program interface (PIF) to the Tivoli Workload Scheduler for z/OS subsystem. The first request in the queue is selected and passed to a general-service executor. This process continues while there are requests in the queue. The GS subtask then waits for more work to arrive or for a stop-processing-ECB to be posted.

Input to the GS subtask is:

- A queue element consisting of an identifier of the caller, its ASCB address, and its TCB address (the field SSCID of the subsystem-caller control block). In addition to this ID, the queue element also contains the time stamp of the request.
- The subsystem-caller control block (SSC), which holds a description of the request in the SSCREQ field.

General-service executor tasks

The general service (GS) executors process the requests that are on the GS queue. The GS task can attach up to five GS executor tasks to prevent service requests from being queued.

HTTP subtasks

The HTTP tracker connection subtasks (HTC and HTS) connect Tivoli Workload Scheduler for z/OS Agents, dynamic domain managers, and remote engines to the controller through an HTTP or HTTPS connection.

The HTC and HTS subtasks run in the controller address space.

HTTP client task (HTC)

The HTC subtask removes the outbound requests from the HTTQ queue, which contains requests queued by the data router, received mainly from the workstation analyzer or the general service subtasks. In particular, the HTTQ queue contains the following request types:

- Job submission, from the workstation analyzer.
- Job log retrieval, from the general service.
- Job cancel, from the general service.
- Bind requests, from the workstation analyzer.

HTTP server task (HTS)

The HTS subtask listens for inbound requests, using the port number specified in the HTTPOPTS statement.

Some of these requests, for example, the notification of operation status changes, are converted into events and are sent to the event manager through the data router.

Job completion checker (JCC)

The job completion checker (JCC) subtask provides support for job-specific and general checking of SYSOUT data sets for jobs entering the JES output queues.

The JCC subtask is “triggered” by job and started-subtask termination (3P) events. A 3P event is generated when SYSOUT is spooled. The 3P event then informs the JCC subtask that SYSOUT data sets are available for the current job in the output classes defined for JCC. The JCC subtask then starts to scan all JOBLOG and other SYSOUT data set records and tries to match these records against the message tables defined to the JCC. There are messages tables:

- General, for all jobs
- Specific, for the job name.

You can create these messages tables by using the EQQJCCT macro. The general message table is mandatory. The JCC subtask searches for job-specific table entries first. If it does not find any, it then searches the entries in the general table.

If a match occurs, an error code can be assigned to the operation (CA=error and EID=0) if you have coded the EQQJCCT macro accurately. This means that the 3P event has been amended with the error code set by the macro. Errors like S222 abends, and JCL errors are reported before 3P events are reported. By specifying the CA and EID, you will override anything reported by earlier events or the current 3P event. Note that a match could occur, but it does not have to be reported as an error to Tivoli Workload Scheduler for z/OS.

If a match does not occur, everything reported by earlier events or the current event (which is 3P) is passed to Tivoli Workload Scheduler for z/OS unaltered. The status for the operation is then updated accordingly.

Subtask initialization consists of:
- Building control blocks for the JCC subtask
- Reading and processing the parameter library member
- Opening the JCC-message-table data set
- Retrieving the general (default) message table member.

Subtask mainline logic provides service for the following events:
- Timer expiration. Retry processing for all delayed jobs.
- Subtask stop request. Return to caller.
- Process all events (3P) on the JCC queue.

Job-tracking-log archiver (JL)

This subtask, which is part of the controller, asynchronously copies the contents of the inactive job-tracking data set to the JT archive data set. The NM subtask attaches and provides work for the job-tracking-log archiver subtask (for a description of the NM, see "Normal mode manager (NM)" on page 70).

The JT archive data set is defined by DD name EQQJTARC and is used as a repository for JT log records associated with each new current plan. The job-tracking-log archiver (JL) subtask owns the JT archive data set, and will OPEN and CLOSE this data set as required.

To enable the JT log data sets to be reusable, the records must be copied to the JT archive data set. The copy routine is called once for each complete, non-archived JT log. The copy is initiated by a CP backup and is driven by the normal-mode-manager (NM) subtask.

The archiving process

When the CP backup is complete, the NM subtask calculates the next JT log sequence number, closes the current JT log, and switches to the next JT log. The NM subtask then posts the JL subtask, which will take one of the following actions, depending upon the ECB posted:
- Work ECB is posted. The subtask reopens the old JT log data set, copies (appends) all non-archived JT logs to the JT archive data set, and posts the NM subtask when the copy is complete. The JT log is normally copied to the archive data set before it is needed again as the current JT log data set. The last record appended is the CP backup record.
- Stop ECB is posted. The subtask frees all resources and terminates.

The JT archive data set is copied to the tracklog (EQQTROUT) data set by the daily plan batch programs. The NM subtask empties the JT archive data set when a daily plan batch job has created a new current plan, and NCP takeover is completed successfully.

Input/output errors

When a write error is found on the current JT log, the log that is in use is switched, and the NM subtask is posted to make a backup. The JL subtask is then posted to copy the JT log that had the output error. If a new output error occurs before the JL subtask has finished copying the first JT log, it is possible that there are no empty JT logs available. In this situation, the NM subtask is posted to stop. To reduce the risk of this occurring, you should always use a minimum of three JT log data sets.

If the dual-logging function is active, Tivoli Workload Scheduler for z/OS duplicates the JT records in the corresponding dual JT log. Dual logs are switched at the same time and in the same sequence as the JT logs. The number of dual JT data sets is determined by the number of normal JT data sets.

JT log sequencing requires that the log that is to be made current has previously been archived. If not, the NM subtask terminates. (Repeated write errors could cause this situation to occur if the JL subtask cannot copy the JT logs as fast as the normal-mode-manager subtask can switch JT logs.)

The subtask terminates if open errors are found on the JT archive data set. Also, the JL subtask terminates if it is unable to copy the JT log to the JT log archive data set because, for example, of an I/O error on the archive data set. If the JT log or archive data set cannot be opened, the subtask terminates. If the subtask terminates abnormally, the JT log data set is not made available for reuse.

Tivoli Workload Scheduler for z/OS automatically recovers from read errors on a JT log during restart. If the error occurs on the first record of the log, the Normal Mode Manager task regards the JT log as empty. A read error on a record other than the first is treated as an end-of-file on the file.

Network communication function (CB)

The network communication function (CB) subtask supports the transmission of data between the controlling system and controlled systems. This data can be events, JCL, commands, or WTO messages. This subtask provides services for:
- Session establishment
- Session termination
- Subsystem stop
- Queued event-record handling
- SEND processing
- RECEIVE processing.

CB subtask initialization consists of:
- Building the internal control blocks (NAB and SCBs)
- Building the VTAM control blocks (ACB, NIB, RPLs, and EXLST)
- Processing initialization parameters
- Building ECB and associated events lists
- Opening the ACB
- Issuing the SETLOGON macro to allow logons.

VTAM exit modules include:
- The LOGON exit
- The VTAM SCIP exit
- Completion exits
- The VTAM LOSTERM exit
- The VTAM TPEND exit

- The VTAM NSEXIT exit.

CB subtask termination consists of closing the ACB.

Normal mode manager (NM)

The normal mode manager (NM) subtask manages these data sets:
- Current plan and its backups (CP)
- Current plan extension (CX)
- Long-term plan (LTP)
- JCL repository (JS)
- Checkpoint
- Job-tracking log
- Side information (SI)
- Extended data and its backups (XD)

The NM subtask keeps track of which data sets are now *current*. It also controls the CP turnover process.

The NM subtask manages IBM Tivoli Workload Scheduler for z/OS startup processing by using the job-tracking-log data set to update the current plan if IBM Tivoli Workload Scheduler for z/OS has to be restarted from a checkpoint (that is, a data set copy) of the current plan. The Tivoli Workload Scheduler for z/OS checkpoint data set contains status information that is used to reposition Tivoli Workload Scheduler for z/OS sequential data sets when the product is restarted (for example, by the hot standby function) after ending abnormally.

NM performs regular backup and reorganization of the primary JS data set to the alternate JS data set. The frequency of this procedure is determined by the maximum size of the JS data set that you have specified with the MAXJSFILE keyword of the JTOPTS statement. See *Customization and Tuning* for more information about MAXJSFILE. You can schedule or request backups of the JS data set using the BACKUP command. Refer to *Managing the Workload* for details.

Subtask initialization consists of:
- Initializing the checkpoint data set.
- Building control blocks for all job-tracking functions.
- Opening the CP, CX, LTP, XD, job-tracking-log, and side information data sets.
- If required, updating the current plan with the events since the last CP checkpoint. The NM subtask tests the checkpoint data set by reading the first record in the data set. If this record is not valid, the checkpoint data set will be formatted.

Subtask mainline logic consists of providing service for the following events:
- Timer expiration. If required, start the CP turnover process.
- Subtask stop request.
- Post the NM work ECB.

Subtask termination consists of closing all open files.

Current plan turnover

Refer to the current plan reference information in *Managing the Workload*.

RODM (RM)

You can use the Resource Object Data Manager to track the status of real resources used by Tivoli Workload Scheduler for z/OS operations. RODM is a data cache that contains information about real resources at your installation. Products such as AOC report actual resource status to RODM; RODM reflects the status by updating values of fields in classes or objects that represent the real resources. Subsystems on the same z/OS image as RODM can subscribe to RODM fields. When RODM updates a field, all subscribers to the field are notified.

Tivoli Workload Scheduler for z/OS support for RODM lets you subscribe to RODM fields for fields in special resources. When RODM notifies a change, Tivoli Workload Scheduler for z/OS updates resource fields that have a subscription to RODM. You can subscribe to RODM for these fields:

AVAILABLE
The *Available* field in the resource. This value overrides the default and interval values.

QUANTITY
The *Quantity* field in the resource. This value overrides the default and interval values.

DEVIATION
The *Deviation* field. You use this field to make a temporary adjustment to quantity. Tivoli Workload Scheduler for z/OS adds quantity and deviation together to decide the amount that operations can allocate. For example, if quantity is 10 and deviation is -3, operations can allocate up to 7 of the resource.

You specify these keywords to invoke monitoring through RODM:

RODMTASK
Is specified on the OPCOPTS statement for the controller and for each tracker that communicates with a RODM subsystem.

RODMPARM
Is specified on the OPCOPTS statement for the controller and identifies the member of the parameter library that contains RODMOPTS statements.

RODMOPTS
Is specified for a controller and contains destination and subscription information.

A RODMOPTS statement is required for each field in every resource that you want to monitor. Each statement is used to subscribe to a field in a RODM class or RODM object for a field in a special resource. The RODM field value is used to set the value of the resource field.

RODMOPTS statements are read when the controller is started. When a tracker that communicates with RODM is started, it requests parameters from the controller. The controller sends subscription information to the tracker, which then subscribes to RODM. An event is created when RODM returns a value, which is used to update the special resource field in the current plan. Tivoli Workload Scheduler for z/OS does not schedule operations that use a special resource until RODM has returned the current field value and Tivoli Workload Scheduler for z/OS has updated the resource.

To use RODM monitoring you must ensure that:

- A tracker is started on the same z/OS image as the RODM subsystem that requests are sent to, and RODMTASK(YES) is specified for both the tracker and the controller.
- An event writer is started in the Tivoli Workload Scheduler for z/OS address space that communicates with RODM. This address space creates resource events (type S) from RODM notifications, which Tivoli Workload Scheduler for z/OS uses to update the current plan.
- The controller is connected to the tracker through XCF, NCF, TCP/IP, or a submit/release data set.
- Each address space has a unique RACF user ID if more than one Tivoli Workload Scheduler for z/OS address space communicates with a RODM subsystem, such as when you start production and test systems that subscribe to the same RODM subsystem.

Tivoli Workload Scheduler for z/OS does not load or maintain data models in the RODM cache, or require a specific data model. You need not write programs or methods to use RODM through Tivoli Workload Scheduler for z/OS, or define specific objects or fields in RODM. Tivoli Workload Scheduler for z/OS does not update RODM-defined data.

RODM fields have several subfields. The RODM field that Tivoli Workload Scheduler for z/OS subscribes to must have a *notify* subfield. Through a subscription to this subfield, RODM notifies the product of changes to the *value* subfield. Tivoli Workload Scheduler for z/OS uses changes to the value subfield to monitor special resources. But only these data types are valid for Tivoli Workload Scheduler for z/OS RODM support:

Table 4. Valid RODM data types for value subfields

Abstract data type	**Data type ID**
CharVar (Char)	4
Integer (Bin 31)	10
Smallint (Bin 15)	21

Tivoli Workload Scheduler for z/OS maintains a RODM status for all special resources in the current plan. You can check the current status in the SPECIAL RESOURCE MONITOR dialog. Each special resource has one of these values:

N Not monitored. The special resource is not monitored through RODM.

I Inactive. Monitoring is not currently active. Tivoli Workload Scheduler for z/OS sets this status for all subscriptions to a RODM subsystem that the controller cannot communicate with. This can occur when communication is lost with RODM or with the tracker. The controller sets the value of each monitored field according to the RODMLOST keyword of RODMOPTS.

P Pending. Tivoli Workload Scheduler for z/OS has sent a subscription request to RODM, but RODM has not returned a value.

A Active. Tivoli Workload Scheduler for z/OS has received a value from RODM and the special resource field has been updated.

RODM support is implemented using the RODM USER API. Tivoli Workload Scheduler for z/OS loads RODM interface module EKGUAPI, which must be accessible via LINKLST or STEPLIB.

The RODM subtask is attached by the subsystem task. The subtask creates a new queue (RODQ) and a table with 1 entry for each active subscription. The table also contains 1 system entry for each connected RODM subsystem. Each subscription entry contains data received from RODM for the subscription and parameter list areas, including the ECB, used for all asynchronous requests.

RODM and Tivoli Workload Scheduler for z/OS interface in this way:

- The queue server contains a *work-to-do* loop, which waits on an ECB list. Initially, this ECB list contains a stop ECB and a queue ECB. The RODM subtask connects and disconnects RODM.
- Inbound subscription messages are placed on the RODM queue (RODQ) by the Tivoli Workload Scheduler for z/OS subsystem task. When a message is detected on the queue, a new subscription entry is added to the table. For the first subscription request to a RODM subsystem, a system entry is also created in the table and a system ECB and a RODM-stop ECB are added to the ECB list.
- A query is issued to the RODM subsystem. The current value is translated and then passed to the writer queue (WTRQ).
- A subscribe is performed and the program goes into WAIT on the ECB list.
- When the subscribe ECB is posted by RODM, the RODM subtask retrieves information from the subscription queue and builds a resource event, which it puts on the WTRQ. All other entries in the table are checked for posted ECBs and the program then returns to a WAIT state.
- The event manager analyzes the incoming resource event and updates the special resource field.

This processing continues until the subtask is stopped or the RODM subsystem ends. If RODM ends, entries in the table are reset. The entries are reprocessed when RODM returns. If the RODM subtask ends, requests outstanding are processed when the subtask is restarted. If Tivoli Workload Scheduler for z/OS is stopped, all subscriptions are deleted.

Submit (SU)

The submit (SU) subtask initiates these functions: job submit, job release, started-subtask initiation, and WLM job promotion. When the submit subtask receives a request, for example, to submit a job, it checkpoints the submit sequence-number, loads the corresponding JCL in a storage buffer, and submits this to the internal reader. If the submit is successful, a submit event (type IJ1, IJ2, IJ3, IWTO, IREL, or IWLM) is built, and a subsystem request is issued to broadcast the event-creation routine. If there is no more work for the submit subtask to process, a WAIT is entered.

The submittor task is part of the tracker, but job submission is initiated by the workstation analyzer (WA) task of the controller. Checkpointing of submission requests is the means by which the two tasks cooperate to ensure that all requests for submission are honored.

When the connection between the controller and a tracker is initially established, and subsequently each time the connection is reestablished after some disruption, the controller requests synchronization. A synch request is sent for each workstation defined with a destination of the particular tracker to synchronize submit requests, as some requests might have been lost because of a bad connection or tracker outage. The tracker responds to the request by issuing a synchronization response event, which is written to the event data and communicated to the controller in the same way as other job-tracking events. The event is prefixed IJ0 (see Chapter 5, “Data areas,” on page 83 for control block mapping) and contains checkpoint information that describes the number of submit events processed by the tracker. When the controller reads the synchronization response event, a check is made to determine the number of submit requests that should have been processed by the tracker. If there is a discrepancy, the controller resends the submit request.

Operations that have been rerouted to an alternate workstation will not be resubmitted.

If the tracker, during normal processing, receives a submit request out of order, the request is discarded, and an out-of-sequence synchronization event is created. Upon receipt of this event, the controller immediately suspends job submission for new operations and resends all requests for the reportedly lost job submission request and all following requests. The lost submit-event record is also prefixed IJ0.

Releasing a job on a JES2 system

On a JES2 system, the Tivoli Workload Scheduler for z/OS release routine does the following:

- If the job is a local job, sends a status request to JES2 to verify that the job is known to, and held by, JES2
- If the job is a local job, builds a release command in the following format:
  ```
  $AJnnnnn                    (nnnnn is job number)
  ```
- If the job is a remote job, builds a release command in the following format:
  ```
  $G A node,'jobname',Jnnnnn,O=orgnode
  ```
- Sends the release command to JES2 using SVC 34.

Releasing a job on a JES3 system

On a JES3 system, the Tivoli Workload Scheduler for z/OS release routine does the following:

- If the job is a local job, sends a status request to JES3 to verify that the job is known to, and held by, JES3
- If the job is a local job, builds a release command in the following format:
  ```
  *F  J=nnnn,R                (nnnn is job number)
  ```
- If the job is a remote job, builds a release command in the following format:
  ```
  *T  node  *F  J=rnnn,R      (rnnn is remote job number)
  ```
- Sends the release command to global JES3, using IATSSCM1.

On JES3 Version 2 Release 2 (or later) systems, the command is sent to JES3 using SVC 34.

Note that the remote release command assumes that the remote system is a JES3 system.

TCP/IP tracker connection (IP)

The TCP/IP tracker connection (IP) subtask connects standard z/OS tracker to the z/OS controller through the Transmission Control Protocol/Internet Protocol (TCP/IP). This task enables IPV6 and IPV4 standards, and SSL authentication and encryption.

The IP task has a queue in the controller and tracker named TIPQ, where the Data Router (DX) task tails the events that will be managed by the task.

The IP task runs on the controller and tracker.

IP task on controller side

On the controller, the IP task opens a main socket where it receives the connection requests from the trackers. The IP task is always waiting for an event; iteration is done if the task receives an event.

The main events that the IP task can receive are:

- There is work on the TIPQ.
- There is a connection request from a tracker (connect() from tracker).
- There are bytes to be read on a socket (write() from tracker).
- Bytes are to be written on a socket (write() to tracker).
- A started task stop request must be processed.
- The loop timeout has expired.
- Task abnormal end.

When the IP task receives an event, it starts iterating and checks:

1. Whether the loop timeout has expired or an abnormal end occurred.
2. The reading sockets to verify if:
   - There is a connection request from a tracker on the task main socket.

     The task checks that the IP address or host name and the port number match with the IP addresses, host names, and port numbers specified in the initialization parameters. If any value is not valid or already in use, the connection requested is not established.
   - There are bytes to be received from trackers

     If there are, they are received by the controller.
3. The task queue. If there is work on the TIPQ, the IP task stores the entries in another internal queue of the task for later processing.
4. The write sockets to verify if there is a connection() or write() request pending for the remote tracker that must be processed. The request result is checked.
5. Whether there are entries in the internal queue. If there are, the IP task processes the DQE and sends the data to the remote trackers.

IP task on tracker side

The IP task on the tracker decides to connect to the controller. The controller is always waiting for the connection request passively. The tracker must always be connected to the controller. If the connection goes down, the tracker tries to establish it again until it succeeds or the tracker is stopped; the error message handling at tracker side is the following:

- The error messages display is enabled at initialization time.
- The error messages display is disabled after a TCP/IP connection problem (EQQIP15E or EQQIP17I are issued in the message log), so that the following connection retries will not issue error messages.
- The error messages display is re-enabled when a connection is successful; an informational message informing about the restored connection is issued (EQQIP12I).

The IP task is always waiting for an event; iteration is done if the task receives an event.

The main events that the IP task can receive are:

- There is work on the TIPQ.
- There are bytes to be read on a socket (write() from controller).
- Bytes are to be written on a socket (write() to controller).
- A started task stop request must be processed.
- The loop timeout has expired.
- Task abnormal end.

When the IP task receives an event, it starts iterating and checks:

1. Whether the loop timeout has expired or an abnormal end occurred.
2. The reading sockets. If there are bytes to be received from controller, the IP task receives the bytes.

   If some buffer received is completed, the received information is checked: if this is a DQE from the controller, the task enqueues the information on the data router queue (RTRQ). If the connection to the controller is not established, the tracker tries to connect even if the loop timeout has expired.
3. The task queue. If there is work on the TIPQ, the IP task stores the entries in another internal queue of the task for later processing.
4. The write sockets to verify if there is a connection() or write() request pending for the controller that must be processed. The request result is checked.
5. Whether there are entries in the internal queue. If there are, the IP task processes the DQE and sends the data to the remote trackers.

Workstation analyzer (WA)

The workstation analyzer (WA) subtask analyzes operations (jobs, started tasks, and WTO messages) that are ready to start at:

- Automatically reporting computer workstations
- Nonreporting workstations
- General workstations that have the WTO option specified.

For each active workstation that is defined in the CP, the WA sends a synchronization request, to the related submit task. The submit checkpoint mechanism is described in detail on page “Submit (SU)” on page 73. The workstation analyzer identifies the most urgent operation to be submitted from a list of ready operations. When an operation is selected, the JCL for the operation must be retrieved. The WA searches for the JCL in the following order:

- The JCL repository
- A data set indicated by the EQQUX002 exit, if the exit is present
- The Tivoli Workload Scheduler for z/OS job library (EQQJBLIB).

Variable substitution is performed for the JCL if required, and the job input is imaged to the JCL repository. The EQQUX001 is called if present. A submit request, which includes the JCL and specified destination, is queued to the data router (DX) task. The workstation analyzer updates the current plan and creates a job-tracking record TRL25 to record the submit request. The operation status is changed to SU to indicate that submission is in progress.

Variable substitution is performed for the System Automation command text if required, and the result is also saved as CPLREC33. The EQQUXSAZ is called, if present. A submit request, which includes the System Automation command and specified destination, is queued to the data router (DX) task. The workstation analyzer updates the current plan and creates a tracking record to track the submit request. The operation status is changed to SU, to indicate that submission is in progress.

The WA subtask also processes any workstation operation that has a deadline WTO message specified. Operations with a deadline WTO message that have reached their latest start times are always selected first. This process continues while there are operations that can be started. The workstation analyzer subtask then waits for more work to arrive or for the next scheduled event to occur. The WA resends submit requests for operations that are flagged as having lost their submit requests.

If you specify the LATEOPER or DURATION keywords on the ALERT initialization statement, the WA is responsible for issuing the relevant alert when the condition is detected.

If you specify the WLM initialization statement, the WA regularly checks if an operation has reached particular time limits that activate the policy. These limits are specified either by default, in the WLM initialization statement and valid for all critical operations or by you, in the Modify Current Plan or the Application Description environment and valid only for the particular operation. In these instances, the WA sends a request to the related submit task for WLM promotion.

Note: On fault-tolerant workstations, the WA subtask manages only non-centralized scripts that have some dependencies on special resources or centralized scripts. Non-centralized scripts without any dependencies on special resources are managed by the batchman process of the fault-tolerant workstations on which the job is scheduled.

Note: If the Workstation Analyzer subtask has no operations to schedule, it is woken up every two minutes so that message EQQE038I can be issued with two minutes of delay.

When an operation is eligible to be started

An operation can be started when:

- The workstation is active (that is, the controller can communicate with it), is open, and has an available parallel server (or the WA has been requested not to use parallel servers in scheduling decisions). For jobs on a fault-tolerant workstation, the workstation must be linked active, CPULIMIT must be nonzero, and the jobs running on the workstation must be fewer than the CPULIMIT set.
- The status of the operation is either ready or arrived, and the manual hold (MH) CP function has not been specified.
- The manual release (MR) CP function has been specified, following a manual hold (MH) request for the operation.
- The duration of the operation is such that the operation can be processed during the time that its associated workstation remains open. This is dependent on what has been specified in the shutdown policy defined in the SHUTDOWN parameter of the JTOPTS initialization statement. See *Customization and Tuning*.
- Either the operation does not use any special resources, or it uses special resources that are available and that can satisfy the allocation criteria.

  The operation specifies a quantity and type for each allocation. Quantity is a number from 1 to 999999, or blank. Blank means the current maximum quantity of the special resource. The type is either shared (other operations can share the allocation concurrently) or exclusive (the operation must be the only user of the allocated quantity).

If an operation needs a special resource that is not defined in the current plan, Tivoli Workload Scheduler for z/OS uses the DYNAMICADD keyword of the RESOPTS statement to determine if it creates the required special resource dynamically.

- Job submission has not been prevented for this operation.
- The execute (EX) CP function has been requested for this operation.
- There have been no errors during a previous attempt to start this operation.
- This is either a normal (non-time-dependent) operation, or it is a time job that can be started now.

  A time job is an operation on a computer workstation that should start at a particular time, as opposed to the normal Tivoli Workload Scheduler for z/OS situation where an operation is started when all its predecessors are complete.

  A time job can have the cancel-if-late attribute. If so, it might not be started after its start time has passed, depending on the SUPPRESSPOLICY parameter specified for the Tivoli Workload Scheduler for z/OS system in the JTOPTS initialization statement. Refer to *Customization and Tuning*.
- Either the operation does not use any workstation resources, or it does not use more workstation resources than are available.

  A workstation resource is related to a specific workstation. For example, you can specify that one workstation has 2 optical readers and 10 tape units. Operations that need tapes or optical readers should then be defined so that Tivoli Workload Scheduler for z/OS knows how many are needed to start the operation. If this is done, operations do not be start (by the WA) on this workstation unless there are enough tapes and optical readers available.

Operation priority

If there is more than one operation that can be started, the WA scheduler selects an operation by investigating these characteristics in the stated order:

1. Priority 9.
2. Earlier latest-start time.

   The latest-start time is calculated internally by Tivoli Workload Scheduler for z/OS, and is the latest time (based on the operation duration) that the operation must start if it is to be completed before its deadline. If no deadline is specified for the operation,Tivoli Workload Scheduler for z/OS uses the occurrence deadline and subtracts the duration for each successor operation on the critical path.
3. Priority 8–1.
4. Shortest estimated duration.
5. First-in first-out basis.

Retrieving JCL

The WA retrieves JCL for the job that is ready to be started and sends it to a submit subtask via the data router (DX) subtask. The WA extracts the JCL for the job from:

- The JS data set, if the JCL to be used is saved there.
- The data set indicated by the EQQUX002 job-library-read exit, if the exit is requested by the EXITS initialization statement.
- The Tivoli Workload Scheduler for z/OS job library (EQQJBLIB).
- The EQQUX001 job-submit exit, if requested.
- A user-defined JCL imbed exit, or EQQJBLIB, if the FETCH directive is included in the `//*%OPC` JCL statement.

Operations handled by the external router (EX) subtask are not required to have JCL fetched by the controller, instead the JCL may be fetched by the operation-initiation exit, EQQUX009, or located at the receiving user-defined destination. If JCL can be located by the controller it will be processed and passed to the EX task, and subsequently to EQQUX009, as normal.

The values for any JCL variables that are present in the JCL are retrieved from JCL variable tables or from a user-defined JCL-variable-substitution exit. When the job is found, the JCL is retrieved and, if necessary, a new JS record is built.

When a JCL directive statement is actioned, it is changed by Tivoli Workload Scheduler for z/OS to a JCL comment statement. JCL directive statements are defined in the job input as z/OS JCL statements. The statement is changed when the operation is submitted to a valid comment statement for the target operating system. The image of the job input in the JCL repository is not altered.

Subtask initialization consists of building the WA environment and opening all required files.

Subtask mainline logic consists of providing service for the following events:
- Timer expiration. Trying to find an operation to be started.
- Post the work ECB. Trying to find an operation to be started.
- Subtask stop request.

Subtask termination consists of closing all open files.

Data store network common function (FN)

This is the task that handles the SNA communication between the controller and Data Store (the XCF Connection is handled by the FL task). Its internal logic is similar to the NCF task.

Data store TCP/IP communication task (ID)

This is the task that handles the TCP/IP communication between the controller and Data Store. Its internal logic is similar to the IP task.

Fetch joblog (FL)

This task is in charge of the Joblog Retrieval and Restart and Clean up functions. It communicates with the Data Store address space to get joblog (Joblog Retrieval) and structured joblog data information (Restart and Clean up).

It is also in charge of starting the local data store used for storing the oper_info structures.

The task is queue-driven. The queues are:

ARCQ
SSX queue used for controller/FL communication.

ANCQ
Internal queue used for data store/FL communication.

Controller requests to FL are sent via DQE of type "ARC":

Type=LOG
Get z/OS joblog

Type=OPI
Ask Oper_info structure

Type=SDEL
Start deletion of old entries in local data store

Pre-submitter task (PSU)

This task is in charge of the JCL tailoring needed to implement the Restart and Cleanup function. It adds the EQQCLEAN pre-step to the JCL in order to:

1. Execute a step restart.
2. Execute a clean up.

PSU interacts with the following other tasks:

- GS
- WSA
- FL
- SU

The task is queue-driven. Its SSX queue is the PSUQ.

Requests to PSU are sent via DQE of type "PSU":

Type=APL
Apply logic: tailor JCL to do clean up and/or step restart

Type=CLN
Stand-alone clean up: create a new job to execute clean up

Tivoli Workload Scheduler for z/OS data store subtasks

Writer

This triggers the reading process for the SYSOUT data, and stores it in the database. More than one Writer task can be active at the same time, to allow parallel operations of SYSOUT data analysis.

Reader

This looks for SYSOUT data in the database and transmits it to the requesting Controller. Every time the Controller asks the Communication subtask for a SYSOUT, The Communication subtask activates the reader. If the Reader does not find the SYSOUT in the Database, then the JES Queue reader is activated.

JesQueue

This triggers the reading process of the job IDs found in the spool classes reserved for the Data Store, and then stores the same IDs in a queue, so allowing the synchronization of the different requests for the active Writer subtasks.

Communication

This performs the following main functions:

- Links the Tivoli Workload Scheduler for z/OS controller by SNA or XCF connection type, depending on the startup parameters
- Receives job SYSOUT requests from the Tivoli Workload Scheduler for z/OS dialog and forwards them to the Data Store reader component

- Sends the data requested by the Controller.

Command

This is the interface between the operator and the Data Store. The Command subtask manages all the external Data Store commands.

Cleanup

This component manages the deletion of SYSOUTs from the database, according to the selection criteria specified in the parameters member whose name corresponds to the CLNPARM parameter in the DSTOPTS initialization statements.

User exit and task relationships

Table 5 and Table 6 show the subsystems in which each user exit of Tivoli Workload Scheduler for z/OS is run. Table 5 lists the user exits and assigns them a key heading that you can match in Table 6 to find the corresponding subsystem and task.

Table 5. User exits in Tivoli Workload Scheduler for z/OS

Exit Name	Exit Type	Heading in Table 6
EQQUX000	Start/Stop	0
EQQUX001	Job-Submit	1
EQQUX002	Job-Library-Read	2
EQQUX003	Application-Description-Feedback	3
EQQUX004	Event-Filtering	4
EQQUX005	SYSOUT Archiving	5
EQQUX006	Incident-Record-Create	6
EQQUX007	Operation-Status-Change	7
EQQUX009	Operation-Initiation	9
EQQUX011	Job-Tracking Log Write	11
EQQUX013	Job-Tailoring Prevention	1
EQQUXPIF	Application-Description-Validation	AV
EQQUXSAZ	System Automation command	SAZ
User defined	JCL-Imbed (FETCH Directive)	JCL
User defined	JCL-Variable-Substitution	JVAR
User defined	Automatic-Job-Recovery	AR
User defined	Daily-Planning-Report	DP

Table 6 shows whether each exit is executed in the Controller, the Tracker, or in a batch program. Tracker exits are run also in the Controller if the Controller has an Event Writer task; that is, if EWTRTASK(YES) is specified in OPCOPTS.

Table 6. User exit and task relationships

Subsystem	Task	0	1	2	3	4	5	6	7	9	11	JCL	JVAR	AR	DP	AV	SAZ
All	System Task Dispatcher	X															
Controller	Automatic Recovery				S				S		S			S			

Table 6. User exit and task relationships (continued)

Subsystem	Task	0	1	2	3	4	5	6	7	9	11	JCL	JVAR	AR	DP	AV	SAZ
Controller	Event Manager				S				S		S						
Controller	External Router									X							
Controller	General Service			S	S				S		S	S	S				
Controller	Normal Mode Manager				S				S		S						
Controller	Workstation Analyzer		S	S	S				S		S	S	S				S
Trackers/ Controller	Event Writer					X											
Trackers/ Controller	Job Completion Checker						X	X									
Batch	Daily Planning														X		
PIF	AD Validation															X	

Notes:

1. The *S* mark means that the execution in the part of the task where the exit is run is subject to serialization. The *X* mark implies that there is no serialization.
2. When a task operates on a resource, it holds it exclusively. Normally the resource is the current plan. While the task is updating the current plan, no other task can reference the current plan at the same time.
3. An exit should avoid processing, such as I/O operations, that may result in system waits and must not perform requests to the Tivoli Workload Scheduler for z/OS program interface.
4. Termination of the Job Completion Checker exits (EQQUX005 and EQQUX006) precludes the ending of the connected job.

Chapter 5. Data areas

This chapter provides graphic representation of many data areas used by Tivoli Workload Scheduler for z/OS. This publication attempts to provide the information that is most often needed, not all data areas are documented here. In general, any of the following specifications were considered to qualify a data area for inclusion in this publication:

- The data area is a physical record mapping
- The data area is used extensively in debugging

Data areas appear alphabetically, by name of the mapping macro. The data areas in this publication are not intended to be used as a customer user interface. The areas are product sensitive and can be changed at any time during the current release without documentation updates to this publication.

The data area map

The data area is described field by field. These field descriptions are taken directly from the system code.

For each field in the data area, the table provides the following information:

Offsets
The address of the field, shown in both decimal and hexadecimal (hexadecimal address in parentheses), relative to the beginning of the data area.

Type The kind of program data defined for this field, such as CHARACTER, SIGNED, UNSIGNED.

Len Size of the field in bytes (decimal).

Name The name of the field, bit, or mask.

Bit or mask names are preceded by a description of the bit position and values, as follows:

```
1...  ....  Refers to bit 0.
....  ..11  Refers to bits 6 and 7.
...1  ....  Refers to bit 3.
11..  1111  Refers to bits 0, 1, 4, 5, 6, and 7.
```

Description
A description of the purpose or meaning of the field, bit, or mask.

The cross-reference table

For each data area with more than 40 fields, a cross-reference table shows the following:

- *Hex Offset*: The hexadecimal offset of the field into the data area (for bits, the hexadecimal offset of the field containing the bit).
- *Hex Value*: Hex values are shown only for bits. The Hex value shown implies the position of the bit in the field containing the bit. For an example illustrating how to use the Hex Value, see the bit DOACMD_FAIL in the DOA data area.

In the DOA data area, the cross-reference table for the DOACMD_FAIL bit looks like this:

Table 7. cross-reference table for DOACMD_FAIL

Name	Hex Offset	Hex Value	Level
DOACMD_FAIL	8F	80	3

In the mapping of the DOA, the DOACMD_FAIL bit appears like this:

Table 8. Mapping of the DOA

143	(8F)	BITSTRING	1	DOAFLAG3	FLAG BYTE
		1...		DOACMD_FAIL	OP COMMAND FAIL FLAG

Hex Offset 8F is the offset of field DOAFLAG3, a 1-byte field. The first bit in this field is named DOACMD_FAIL. Ignoring the other bits in the field DOAFLAG3, if the DOACMD_FAIL bit is on, the value of field DOAFLAG3 would be 1000 0000, which is equivalent to hex 80. This value (hex 80) is shown in the Hex Value column of the cross-reference table. When a bit is on, the condition described in the Description field of the mapping is true.

Date formats used in Tivoli Workload Scheduler for z/OS

Different components of Tivoli Workload Scheduler for z/OS use different formats to represent dates. Table 9 describes how these formats are used.

Table 9. Date formats used in Tivoli Workload Scheduler for z/OS

Tivoli Workload Scheduler for z/OS component	Date format	Length	Date type
Controller	YYMMDD	6	Internal date
	0nYYDDDF	4	Time stamp
Trackers	YYMMDD	6	Real date
	0nYYDDDF	4	Time stamp
Description databases	YYMMDD	6	Internal date
Batch loader and SYSIN for batch programs	YYMMDD	6	Real date
Dialogs and reports	Defined by the user, for example YY/MM/DD	8	Real date
Exits	YYMMDD	6	Real date
	0nYYDDDF	4	Time stamp
Programming interfaces	YYMMDD	6	Defined by the user

Note:

1. Real date means that 1 January 1998 is represented as 980101.
2. For the internal date, Tivoli Workload Scheduler for z/OS uses 1972 as the base year for the century window. This means that the year 1972 is represented as 00 and 2071 is represented as 99.
3. The z/OS time stamp uses 00YYDDDF for dates in the 20th century, and 01YYDDDF for dates in the 21st century.

4. The date format of the programming interfaces depends on the base year of the PIF century window defined by the PIFCWB keyword of the INTFOPTS statement, or the CWBASE keyword of the INIT statement. 00 implies real dates. 72 corresponds to theTivoli Workload Scheduler for z/OS internal date format.

ADR - application description

```
Name : DCLADR

Function:
This segment declares an application or job description in the application description
database (EQQADDS)

 The AD record is structured as follows:
     1 adr
       2 common data
       2 run cycles ( )
         3 number of positive offsets ( )
         3 number of negative offsets ( )
         3 rule ( )
       2 operations ( )
         3 internal predecessors ( )
         3 external predecessors ( )
         3 special resources ( )
         3 extended name ( )
         3 system automation info ( )
         3 condition information( )
         3 condition dependency( )
    The following structures and pointers are defined:
       adr based(adrptr)
       adrrun based(runptr)
       adrop based(opptr)
       adropint internal predecessors section
       adropext based(extptr) external pred. section
       adropsrs resource section
       adrop_extinfo section
       adrop_autoinfo section
       adrop_cond_info
       adrop_cond_simple

    Offsets to first run cycle and first operation are saved in the fixed part of the record.  Run
    cycles and operations are chained together by offsets from beginning of each segment.
```

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	adr	

Fixed part of common data

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	208	adrcommon	
0	(0)	CHARACTER	2	*	used by IO routines
2	(2)	CHARACTER	23	adrkey	key of the record
2	(2)	CHARACTER	16	adrid	application id
18	(12)	CHARACTER	1	adrstat	status and type adgc

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
19	(13)	CHARACTER	6	adrto	valid to
25	(19)	CHARACTER	4	adreye	eye catcher
29	(1D)	UNSIGNED	1	adrvers	version number
30	(1E)	CHARACTER	6	adrfrom	valid from
36	(24)	CHARACTER	24	adrdescr	descriptive text
60	(3C)	CHARACTER	8	adrgroup	authority group name
68	(44)	CHARACTER	16	adrowner	owner id
84	(54)	CHARACTER	24	adrodesc	owner description
108	(6C)	CHARACTER	1	*	spare
109	(6D)	CHARACTER	6	adrldate	last update date
115	(73)	CHARACTER	4	adrltime	last update time
119	(77)	CHARACTER	8	adrluser	lastupdate user-id
127	(7F)	CHARACTER	1	adrprior	priority
128	(80)	CHARACTER	3	adrdsm	Deadline smoothing factor
131	(83)	CHARACTER	3	adrdlim	Deadline feedback limit
134	(86)	SIGNED	2	adrop	total number of operations
136	(88)	SIGNED	4	adrrun	total number of runcycles

offsets and flags

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
140	(8C)	SIGNED	4	adrfrun	offset to first runcycle
144	(90)	SIGNED	4	adrfop	offset to first op
148	(94)	CHARACTER	1	adrflags	ad flags
		1...		adrerd	deadline cannot be met
		.1..		adrjobd	1 application is job description compliant
		..1.		adrmon	1 any mon. job in appl
		...1		adrcaldef	work bit, off in DB
		 1111		*	free
149	(95)	CHARACTER	16	adrcal	calendar identity
165	(A5)	CHARACTER	16	adrgroupid	group application adga
181	(B5)	CHARACTER	19	*	free
200	(C8)	CHARACTER	8	adrluts	last update timestamp
208	(D0)	CHARACTER	*	adrvarsect	variable length section

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	160	adrrun	

run cycle definition

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	SIGNED	4	adrrnrun	offset to next runcycle rel to beginning of adr
4	(4)	SIGNED	2	adrr	run cycle number
6	(6)	CHARACTER	8	adrrper	period name, rule name
14	(E)	SIGNED	2	adrrnpos	number of pos offsets
16	(10)	SIGNED	2	adrrnneg	number of neg offsets
18	(12)	UNSIGNED	1	adrrule	free day rule

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
19	(13)	CHARACTER	6	adrrfrom	run cycle valid from
25	(19)	CHARACTER	6	adrrout	run cycle not valid from
31	(1F)	CHARACTER	50	adrrdesc	run cycle description
81	(51)	CHARACTER	1	adrrtype	run cycle type see rc type constants
82	(52)	CHARACTER	4	adrrarri	input arrival time
86	(56)	CHARACTER	6	adrrdead	deadline
86	(56)	CHARACTER	2	adrrday	deadline day rel. to start
88	(58)	CHARACTER	4	adrrtime	deadline time
92	(5C)	CHARACTER	16	adrrjvt	jcl var table name
108	(6C)	SIGNED	2	adrrrdlen	rule definition length
110	(6E)	CHARACTER	4	adrrevyrr	EVERY repeat range
114	(72)	CHARACTER	4	adrrevyet	EVERY end time
118	(76)	CHARACTER	2	*	free
120	(78)	SIGNED	4	adrrshift	SHIFT
124	(7C)	CHARACTER	1	adrrshtype	W D work day day
125	(7D)	CHARACTER	3	*	free
128	(80)	CHARACTER	8	adrruleper (4)	rule periods
160	(A0)	CHARACTER		adrrunend	end of run cycle def

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	adrrdaypos	positive run days
0	(0)	CHARACTER	3	adrrdayp (*)	run day offsets

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	adrrdayneg	negative run days
0	(0)	CHARACTER	3	adrrdayn (*)	run day offsets

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	adrrrdefseg	rule definition segmen follows the variable number of positive and negative offsets
0	(0)	CHARACTER	*	adrrrdef	rule definition value

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	128	adrop	

operation definition

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	SIGNED	4	adronop	offset to next op rel to beginning of adr
4	(4)	CHARACTER	6	adropid	operation id
4	(4)	CHARACTER	4	adrwsid	workstation
8	(8)	SIGNED	2	adropno	operation number
10	(A)	CHARACTER	24	adropdes	operation description
34	(22)	SIGNED	4	adropdur	dura, 100th of sec
38	(26)	CHARACTER	3	adropsm	smoothing factor

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
41	(29)	CHARACTER	3	adroplim	limit for feedback
44	(2C)	CHARACTER	1	adropaec	auto error completion YN
45	(2D)	CHARACTER	8	adropjn	job name
53	(35)	CHARACTER	1	adropjcl	job class
54	(36)	SIGNED	2	adrophrc	highest rc not in error
56	(38)	CHARACTER	1	adroppcl	print class
57	(39)	CHARACTER	8	adropfor	form number
65	(41)	CHARACTER	6	adropst	earliest start
65	(41)	CHARACTER	2	adropstd	relative day
67	(43)	CHARACTER	4	adropstt	start time
71	(47)	CHARACTER	6	adropded	deadline
71	(47)	CHARACTER	2	adropdd	relative day
73	(49)	CHARACTER	4	adropdt	start time
77	(4D)	CHARACTER	1	adropflg	operation flags
		1...		adropajr	automatic cpu release
		.1..		adropwk1	not used
		..1.		adropcan	suppress if late & timejob
		...1		adroptim	submit job on time
		.... 1...		adropwk2	not used
		.... .1..		adropsub	automatic submit
		.... ..1.		adrhrcs	highest rc (adrophrc) set
		1		adropwto	deadline wto
78	(4E)	SIGNED	2	adropr1	ws resource amount requird
80	(50)	BITSTRING	1	adropflg2	Adrop flags
		1...		adropcondrjob	Cond Recovery job
		.111 1111		*	free
81	(51)	UNSIGNED	1	adroptemp	temp for ws info
82	(52)	SIGNED	2	adropr2	ws resource amount requird
84	(54)	CHARACTER	2	*	free
86	(56)	SIGNED	2	adropps	no. of servers used
88	(58)	SIGNED	2	adropin	no. of internal pred.
90	(5A)	SIGNED	2	adropex	no. of external pred.
92	(5C)	SIGNED	2	adropsr	no. of special resources
94	(5E)	CHARACTER	1	adrop_rerut	op rerouteable
95	(5F)	CHARACTER	1	adrop_resta	op restartable
96	(60)	CHARACTER	1	adropcm	catalog management, cata A = Automatic, D90Ccata I = Immediate, D90Ccata M = Manual, D90Ccata N = None cata
97	(61)	UNSIGNED	1	adrtrtemp	temp for retrieval d19a D23A-start
98	(62)	CHARACTER	2	adropwlm	WLM fields
98	(62)	CHARACTER	1	adropjobcrt	Critical Job
99	(63)	CHARACTER	1	adropjobpol	Late Jobe Policy D23A---end
100	(64)	CHARACTER	1	adrflag2	D90A
		1...		adrop_UserSYS	1 = Need User Sysout D90A
		.1..		adrop_ExpJCL	1 = Need Expanded JCLD90A
		..1.		ADROPMON	1 monitor the job
		...1		adropscript	1 central script used
		 1111		*	
101	(65)	UNSIGNED	1	adropExtInf	Extended JobName area 0 = not allocated 1 = allocated
102	(66)	CHARACTER	1	adrop_UseExtInf	Extended JobName info N = not used Y = used

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
103	(67)	CHARACTER	1	adrop_UseSEinf	Scheduling Env. Name CCUA N = not used CCUA Y = used CCUA
104	(68)	UNSIGNED	1	adropAutInf	Automation Info areaCHWA 0 = not allocated CHWA 1 = allocated CHWA
105	(69)	CHARACTER	1	adrop_UseAutInf	Sys Automation info CHWA N = not used CHWA Y = used CHWA
106	(6A)	CHARACTER	8	adropWLMClass	WLM Service class
114	(72)	SIGNED	2	adropUF	Number of User Fields
116	(74)	SIGNED	2	adropCC	Number of Cond
118	(76)	SIGNED	2	adropCS	Number of Simple Cond
120	(78)	UNSIGNED	1	adrop_Reng	remote engine area 0 = not allocated 1 = allocated
121	(79)	CHARACTER	1	*	free (total)
122	(7A)	SIGNED	2	adropexIVL	Nbr of NORMAL ext d IVL
124	(7C)	SIGNED	2	adropCI	Total Cond IVL
126	(7E)	CHARACTER	2	*	free
128	(80)	CHARACTER		adropFixEnd	end adrop

operation dependencies

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	16	adropint (*)	internal dependencies.
0	(0)	CHARACTER	6	adroppre	predecessor id
0	(0)	CHARACTER	4	adrip_wsid	work station id
4	(4)	SIGNED	2	adrip_opno	operation number
6	(6)	CHARACTER	4	adroptrt	transport time
6	(6)	CHARACTER	2	adrophh	hours
8	(8)	CHARACTER	2	adropmm	minutes
10	(A)	CHARACTER	6	*	free

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	84	adropext (*)	external dependencies.
0	(0)	CHARACTER	22	adropepr	predecessor id
0	(0)	CHARACTER	16	adropead	external predecessor ad id
16	(10)	CHARACTER	6	adropeop	predecessor id
16	(10)	CHARACTER	4	adrep_wsid	work station id
20	(14)	SIGNED	2	adrep_opno	operation number
22	(16)	CHARACTER	4	adropetr	transport time
26	(1A)	CHARACTER	50	adropede	description
76	(4C)	CHARACTER	1	adropltp	ltp print option
77	(4D)	CHARACTER	1	adropeflg	flags
		1...		adropewrk	work
		.111 1111		*	
78	(4E)	CHARACTER	1	adrXCsel	- C (closest prec) - S (same day) - R (relative ivl) - A (absolute ivl)
79	(4F)	CHARACTER	1	adrXMandatory	- P (plan) - C (control) - N (no)
80	(50)	CHARACTER	4	*	
84	(54)	CHARACTER		adropextEND	

ADROPEXTIVL Operation external dependencies IVL
the number of them is get from adropexIVL ADROP

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	40	adropextIVL (*)	

pred key

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	22	adropextIVL_key	pred id
0	(0)	CHARACTER	16	adropextIVL_Adid	- ext pred ADID
16	(10)	CHARACTER	6	adropextIVL_op	- predecessor id
16	(10)	CHARACTER	4	adropextIVL_wsid	WSID
20	(14)	SIGNED	2	adropextIVL_opno	OP NUMBER

IVL info

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
22	(16)	CHARACTER	1	adropextIVL_typ	A=abs R=rel
23	(17)	CHARACTER	3	*	
26	(1A)	CHARACTER	7	adropextIVLFr	
26	(1A)	CHARACTER	1	adropextIVLfr_when	BA
27	(1B)	CHARACTER	1	adropextIVLfr_day	0 to 7 days
28	(1C)	CHARACTER	3	adropextIVLfr_HHH	000 to 167
28	(1C)	CHARACTER	1	*	
29	(1D)	CHARACTER	2	adropextIVLfr_HH	00 to 23
31	(1F)	CHARACTER	2	adropextIVLfr_MM	00 to 59
33	(21)	CHARACTER	7	adropextIVLTo	
33	(21)	CHARACTER	1	adropextIVLto_when	BA
34	(22)	CHARACTER	1	adropextIVLto_day	0 to 7 days
35	(23)	CHARACTER	3	adropextIVLto_HHH	000 to 167
35	(23)	CHARACTER	1	*	
36	(24)	CHARACTER	2	adropextIVLto_HH	00 to 23
38	(26)	CHARACTER	2	adropextIVLto_MM	00 to 59

ADROPSRS resources for this operation
the number of ADROPSRS is get from adropsr ADROP

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	64	adropsrs (*)	special resources.

resources for this operation

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	44	adropsrn	name resource
44	(2C)	SIGNED	4	adropsra	numeric amount
48	(30)	CHARACTER	1	adropsrt	s = shared, x = exclusive

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
49	(31)	CHARACTER	1	adropsrk	Keep on error flag Y , N , blank
50	(32)	CHARACTER	1	adropsrAVAco	Set SR availaibility to Y , N , R or blank
51	(33)	CHARACTER	13	*	free

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	200	adrop_ExtInfo	

2A Extended Information for this operation $CCUC
22A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	54	adropEXTname	Extended Job Name
54	(36)	CHARACTER	16	adropSEname	Scheduling Env. Name CCUA
70	(46)	CHARACTER	130	*	free 22ACCUC CHWA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	336	adrop_AutoInfo	Automation Info CHWA CHWA CHWA

$CH
WA Automation info for this operation $CHWA
$CHWA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	256	adropCommText	SA op command texCHWA
0	(0)	CHARACTER	64	adropCommText1	op com line 1 CHWA
64	(40)	CHARACTER	64	adropCommText2	op com line 2 CHWA
128	(80)	CHARACTER	64	adropCommText3	op com line 3 CHWA
192	(C0)	CHARACTER	63	adropCommText4	op com line 4 CHWA
255	(FF)	CHARACTER	1	adropFiller	Reserved CHWA
256	(100)	CHARACTER	8	adropAutoOper	SA automated operCHWA
264	(108)	CHARACTER	8	adropSecElem	SA security elem CHWA
272	(110)	CHARACTER	64	adropCompInfo	SA completion infCHWA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	32	adrop_COND_INFO (*)	
0	(0)	SIGNED	2	adropCO_ID	
2	(2)	SIGNED	2	adropCO_SimpNo	
4	(4)	SIGNED	2	adropCO_Type	
6	(6)	SIGNED	2	adropCO_Count	
8	(8)	CHARACTER	24	adropCO_Descr	

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	56	adrop_COND_SIMPLE (*)	
0	(0)	SIGNED	2	adropSC_ID	
2	(2)	CHARACTER	22	adropSC_Predid	
2	(2)	CHARACTER	16	adropSC_preAD	
18	(12)	CHARACTER	6	adropSC_preOP	
18	(12)	CHARACTER	4	adropSC_prewsid	
22	(16)	SIGNED	2	adropSC_preopno	
24	(18)	CHARACTER	2	adropSC_PreTyp	RC or ST
26	(1A)	CHARACTER	2	adropSC_PreLog	GE,GT,EQ,LT,LE,NE,RG
28	(1C)	CHARACTER	4	adropSC_ValRC	
32	(20)	CHARACTER	4	adropSC_ValRC2	
36	(24)	CHARACTER	1	adropSC_ValST	
37	(25)	CHARACTER	1	adropSC_DepTyp	I=INT E=EXT
38	(26)	CHARACTER	1	adropSC_Csel	CSRA mat ctiteria not used
39	(27)	BITSTRING	1	adropSC_Flags	
		1...		adropSC_wrk	
		.1..		adropSC_manP	mandatory P
		..1.		adropSC_manC	mandatory C
		...1 1111		*	
40	(28)	CHARACTER	8	adropSC_Step	STEPNAME
48	(30)	CHARACTER	8	adropSC_PStep	PROCSTEP

ADROPCONDCEXTIVL

Matching criteria info for external cond pred
NOTE we can have only one ADROPEXTIVL
for each ext cond dep but the total number of
these intervals is get fropm adropCOExtIVL

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	36	adrop_Cond_CextIVL (*)	

pred key

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	CHARACTER	16	adropCextIVL_Adid	ADID
16	(10)	SIGNED	2	adropCextIVL_Opnum	Opnum
18	(12)	SIGNED	2	adropCextIVL_CondId	CondId

IVL info

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
20	(14)	CHARACTER	1	adropCextIVL_typ	A=abs R=rel
21	(15)	CHARACTER	1	*	free
22	(16)	CHARACTER	7	adropCextIVLFr	
22	(16)	CHARACTER	1	adropCextIVLfr_when	BA
23	(17)	CHARACTER	1	adropCextIVLfr_day	0 to 7 days
24	(18)	CHARACTER	3	adropCextIVLfr_HHH	000 to 167

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
24	(18)	CHARACTER	1	*	
25	(19)	CHARACTER	2	adropCextIVLfr_HH	00 to 23
27	(1B)	CHARACTER	2	adropCextIVLfr_MM	00 to 59
29	(1D)	CHARACTER	7	adropCextIVLTo	
29	(1D)	CHARACTER	1	adropCextIVLto_when	BA
30	(1E)	CHARACTER	1	adropCextIVLto_day	0 to 7 days
31	(1F)	CHARACTER	3	adropCextIVLto_HHH	000 to 167
31	(1F)	CHARACTER	1	*	
32	(20)	CHARACTER	2	adropCextIVLto_HH	00 to 23
34	(22)	CHARACTER	2	adropCextIVLto_MM	00 to 59

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	74	adrop_UserFields (*)	
0	(0)	CHARACTER	16	adropUF_Name	Field name
16	(10)	CHARACTER	54	adropUF_Value	Field value
70	(46)	CHARACTER	4	*	not used

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	200	adrop_reng_info	
0	(0)	CHARACTER	16	adropRE_jsname	ad id
16	(10)	CHARACTER	1	adropRE_shadow	Y for shadow jobs
17	(11)	CHARACTER	3	adropRE_opno	operation number TWSz
20	(14)	CHARACTER	40	adropRE_jobname	job name TWSd
60	(3C)	CHARACTER	16	adropRE_jsws	jobstream ws TWSd
76	(4C)	CHARACTER	1	adropRE_comp	complete on bind err
77	(4D)	CHARACTER	1	adropRE_isMand	PCN mandatory
78	(4E)	CHARACTER	1	adropRE_cSel	CSRA mat criteria
79	(4F)	CHARACTER	16	adropRE_IVL	
79	(4F)	CHARACTER	1	*	free
80	(50)	CHARACTER	1	*	free
81	(51)	CHARACTER	7	adropRE_IVLFr	
81	(51)	CHARACTER	1	adropRE_IVLfr_when	BA
82	(52)	CHARACTER	1	adropRE_IVLfr_day	0 to 7 days
83	(53)	CHARACTER	3	adropRE_IVLfr_HHH	000 to 167
83	(53)	CHARACTER	1	*	
84	(54)	CHARACTER	2	adropRE_IVLfr_HH	00 to 23
86	(56)	CHARACTER	2	adropRE_IVLfr_MM	00 to 59
88	(58)	CHARACTER	7	adropRE_IVLTo	
88	(58)	CHARACTER	1	adropRE_IVLto_when	BA
89	(59)	CHARACTER	1	adropRE_IVLto_day	0 to 7 days
90	(5A)	CHARACTER	3	adropRE_IVLto_HHH	000 to 167
90	(5A)	CHARACTER	1	*	
91	(5B)	CHARACTER	2	adropRE_IVLto_HH	00 to 23
93	(5D)	CHARACTER	2	adropRE_IVLto_MM	00 to 59
95	(5F)	CHARACTER	105	*	free

Cross reference

Name	Hex Offset	Hex Value	Level
adr	0		1
adrcal	95		3
adrcaldef	94	10	4
adrcommon	0		2
adrdescr	24		3
adrdlim	83		3
adrdsm	80		3
adrep_opno	14		4
adrep_wsid	10		4
adrerd	94	80	4
adreye	19		3
adrflags	94		3
adrflag2	64		2
adrfop	90		3
adrfrom	1E		3
adrfrun	8C		3
adrgroup	3C		3
adrgroupid	A5		3
adrhrcs	4D	02	3
adrid	2		4
adrip_opno	4		3
adrip_wsid	0		3
adrjobd	94	40	4
adrkey	2		3
adrldate	6D		3
adrltime	73		3
adrluser	77		3
adrluts	C8		3
adrmon	94	20	4
adrodesc	54		3
adronop	0		2
adrop	0		1
adrop_AutoInfo	0		1
adrop_Cond_CextIVL	0		1
adrop_COND_INFO	0		1
adrop_COND_SIMPLE	0		1
adrop_ExpJCL	64	40	3
adrop_ExtInfo	0		1
adrop_Reng	78		2

Name	Hex Offset	Hex Value	Level
adrop_reng_info	0		1
adrop_rerut	5E		2
adrop_resta	5F		2
adrop_UseAutInf	69		2
adrop_UseExtInf	66		2
adrop_UserFields	0		1
adrop_UserSYS	64	80	3
adrop_UseSEinf	67		2
adrop	86		3
adropCC	74		2
adropCI	7C		2
adropCS	76		2
adropex	5A		2
adropexIVL	7A		2
adropin	58		2
adropps	56		2
adropr1	4E		2
adropr2	52		2
adropsr	5C		2
adropUF	72		2
adropaec	2C		2
adropajr	4D	80	3
adropAutInf	68		2
adropAutoOper	100		2
adropcan	4D	20	3
adropCextIVL_Adid	0		2
adropCextIVL_CondId	12		2
adropCextIVL_Opnum	10		2
adropCextIVL_typ	14		2
adropCextIVLFr	16		2
adropCextIVLfr_day	17		3
adropCextIVLfr_HH	19		4
adropCextIVLfr_HHH	18		3
adropCextIVLfr_MM	1B		3
adropCextIVLfr_when	16		3
adropCextIVLTo	1D		2
adropCextIVLto_day	1E		3
adropCextIVLto_HH	20		4
adropCextIVLto_HHH	1F		3
adropCextIVLto_MM	22		3
adropCextIVLto_when	1D		3

Name	Hex Offset	Hex Value	Level
adropcm	60		2
adropCO_Count	6		2
adropCO_Descr	8		2
adropCO_ID	0		2
adropCO_SimpNo	2		2
adropCO_Type	4		2
adropCommText	0		2
adropCommText1	0		3
adropCommText2	40		3
adropCommText3	80		3
adropCommText4	C0		3
adropCompInfo	110		2
adropcondrjob	50	80	3
adropdd	47		3
adropded	47		2
adropdes	A		2
adropdt	49		3
adropdur	22		2
adropead	0		3
adropede	1A		2
adropeflg	4D		2
adropeop	10		3
adropepr	0		2
adropetr	16		2
adropewrk	4D	80	3
adropext	0		1
adropextEND	54		2
adropExtInf	65		2
adropextIVL	0		1
adropextIVL_Adid	0		3
adropextIVL_key	0		2
adropextIVL_op	10		3
adropextIVL_opno	14		4
adropextIVL_typ	16		2
adropextIVL_wsid	10		4
adropextIVLFr	1A		2
adropextIVLfr_day	1B		3
adropextIVLfr_HH	1D		4
adropextIVLfr_HHH	1C		3
adropextIVLfr_MM	1F		3
adropextIVLfr_when	1A		3

Name	Hex Offset	Hex Value	Level
adropextIVLTo	21		2
adropextIVLto_day	22		3
adropextIVLto_HH	24		4
adropextIVLto_HHH	23		3
adropextIVLto_MM	26		3
adropextIVLto_when	21		3
adropEXTname	0		2
adropFiller	FF		3
adropFixEnd	80		2
adropflg	4D		2
adropflg2	50		2
adropfor	39		2
adrophh	6		3
adrophrc	36		2
adropid	4		2
adropint	0		1
adropjcl	35		2
adropjn	2D		2
adropjobcrt	62		3
adropjobpol	63		3
adroplim	29		2
adropltp	4C		2
adropmm	8		3
ADROPMON	64	20	3
adropno	8		3
adroppcl	38		2
adroppre	0		2
adropRE_comp	4C		2
adropRE_cSel	4E		2
adropRE_isMand	4D		2
adropRE_IVL	4F		2
adropRE_IVLFr	51		3
adropRE_IVLfr_day	52		4
adropRE_IVLfr_HH	54		5
adropRE_IVLfr_HHH	53		4
adropRE_IVLfr_MM	56		4
adropRE_IVLfr_when	51		4
adropRE_IVLTo	58		3
adropRE_IVLto_day	59		4
adropRE_IVLto_HH	5B		5
adropRE_IVLto_HHH	5A		4

Name	Hex Offset	Hex Value	Level
adropRE_IVLto_MM	5D		4
adropRE_IVLto_when	58		4
adropRE_jobname	14		2
adropRE_jsname	0		2
adropRE_jsws	3C		2
adropRE_opno	11		2
adropRE_shadow	10		2
adropSC_Csel	26		2
adropSC_DepTyp	25		2
adropSC_Flags	27		2
adropSC_ID	0		2
adropSC_manC	27	20	3
adropSC_manP	27	40	3
adropSC_preAD	2		3
adropSC_Predid	2		2
adropSC_PreLog	1A		2
adropSC_preOP	12		3
adropSC_preopno	16		4
adropSC_PreTyp	18		2
adropSC_prewsid	12		4
adropSC_PStep	30		2
adropSC_Step	28		2
adropSC_ValRC	1C		2
adropSC_ValRC2	20		2
adropSC_ValST	24		2
adropSC_wrk	27	80	3
adropscript	64	10	3
adropSecElem	108		2
adropSEname	36		2
adropsm	26		2
adropsra	2C		2
adropsrAVAco	32		2
adropsrk	31		2
adropsrn	0		2
adropsrs	0		1
adropsrt	30		2
adropst	41		2
adropstd	41		3
adropstt	43		3
adropsub	4D	04	3
adroptemp	51		2

Name	Hex Offset	Hex Value	Level
adroptim	4D	10	3
adroptrt	6		2
adropUF_Name	0		2
adropUF_Value	10		2
adropwk1	4D	40	3
adropwk2	4D	08	3
adropwlm	62		2
adropWLMClass	6A		2
adropwto	4D	01	3
adrowner	44		3
adrprior	7F		3
adrr	4		2
adrrarri	52		2
adrrday	56		3
adrrdayn	0		2
adrrdayneg	0		1
adrrdayp	0		2
adrrdaypos	0		1
adrrdead	56		2
adrrdesc	1F		2
adrrevyet	72		2
adrrevyrr	6E		2
adrrfrom	13		2
adrrjvt	5C		2
adrrnneg	10		2
adrrnpos	E		2
adrrnrun	0		2
adrrout	19		2
adrrper	6		2
adrrrdef	0		2
adrrrdefseg	0		1
adrrrdlen	6C		2
adrrshift	78		2
adrrshtype	7C		2
adrrtime	58		3
adrrtype	51		2
adrrule	12		2
adrruleper	80		2
adrrun	0		1
adrrun	88		3
adrrunend	A0		2

Name	Hex Offset	Hex Value	Level
adrstat	12		4
adrto	13		4
adrtrtemp	61		2
adrvarsect	D0		2
adrvers	1D		3
adrwsid	4		3
adrXCsel	4E		2
adrXMandatory	4F		2

AWSN - Actual workstation resource report record in DP

Name : DCLAWSN

Function:
Describes layout of parallel servers, R1 and R2 indicating the actual workstation load measured for these resources. There is one record for each workstation and 'day' in the plan. A 'day' starts|ends on the time given in PLANSTART. For each 15 minutes an amount is recorded of PS/R1/R2 resources required (according to the operation descriptions) multiplied by 15 planned server minutes for all operations with start-end contained in the interval (if 1/3 of the interval covered - 1/3*amount). An example: an operation running between 14.30 and 14.48 using 1 server, 50 R1, and 2 R2 will record 15 server minutes, 15*50 R1 units and 15*2 R2 units in the 14.30-14.45 slot and will record 3 server minutes, 3*50 R1 units and 3*2 R2 units in the 14.45-15.00 slot. The records have the standard recsize and keysize of DPIN records.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	263	AWSAREA	PARALLEL SERV|R1|R2 REC
0	(0)	CHARACTER	114	AWSKEY	KEY
0	(0)	SIGNED	2	AWSTYPE	ALWAYS TYPE 7
2	(2)	CHARACTER	34	*	
36	(24)	CHARACTER	4	AWSWSID	WORKSTATION ID
40	(28)	SIGNED	2	AWSSTYPE	SUBTYPE SERVER MINUTES= 1 R1 = 2 R2 = 3
42	(2A)	CHARACTER	6	AWSDATE	REPORT 'DAY' (YYMMDD)
48	(30)	CHARACTER	2	AWSSTRNG	R1|R2 STRING IN REPORTS BLANK FOR PARALLEL SERV
50	(32)	CHARACTER	64	*	TO ADD UP THE KEY
114	(72)	CHARACTER	149	AWSDATA	DATA PART OF AWS RECORD
114	(72)	SIGNED	2	AWSFACT	FACTOR TO MULTIPLY AMOUNT
116	(74)	UNSIGNED	1	AWSAMT (96)	OF RESOURCES
212	(D4)	CHARACTER	51	*	TO ADD UP TO STD RECSIZE

Cross reference

Name	Hex Offset	Hex Value	Level
AWSAMT	74		3
AWSAREA	0		1

Name	Hex Offset	Hex Value	Level
AWSDATA	72		2
AWSDATE	2A		3
AWSFACT	72		3
AWSKEY	0		2
AWSSTRNG	30		3
AWSSTYPE	28		3
AWSTYPE	0		3
AWSWSID	24		3

CAL - Calendar definition record

Name : DCLCAL

Function:
This segment declares a calendar definition, physically located in the workstation description database (EQQWSDS)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	96	CALENDAR	CALENDAR HEADER
0	(0)	CHARACTER	2	CALTYPE	RECORD TYPE = '06'
2	(2)	CHARACTER	8	CALKEY	KEY OF CALENDAR
2	(2)	SIGNED	2	CALNUM	RECORD NUMBER
4	(4)	CHARACTER	4	CALEYE	EYE CATCHER (ALWAYS='CAL')
8	(8)	SIGNED	2	*	RESERVED
10	(A)	CHARACTER	16	CALIDENT	IDENTITY OF THE CALENDAR
26	(1A)	CHARACTER	30	CALDESC	DESCRIPTION OF CALENDAR
56	(38)	CHARACTER	8	CALUPDU	TSO USER IDENTITY OF LAST UPDATING USER
64	(40)	CHARACTER	6	CALUPDD	DATE (YYMMDD FORMAT) AND
70	(46)	CHARACTER	4	CALUPDT	TIME (HHMM FORMAT) WHEN CALENDAR WAS LAST UPDATED
74	(4A)	SIGNED	2	CAL#SPEC	NUMBER OF SPECIFIC DATES AND WEEK DAYS
76	(4C)	CHARACTER	4	CALSHIFT	SHIFT TIME
80	(50)	UNSIGNED	1	CALVER	RECORD LAYOUT VERSION
81	(51)	CHARACTER	7	*	FREE
88	(58)	CHARACTER	8	CALLUTS	LAST UPDATE TIMESTAMP
96	(60)	CHARACTER		CALVARS	VARIABLE SECTION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
96	(60)	STRUCTURE	*	CALDATES	EACH CALENDAR WEEK DAY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
96	(60)	CHARACTER	52	CALDAT (*)	OR SPECIFIC DATE HOLDS THE FOLLOWING FIELDS
96	(60)	CHARACTER	6	CALDATE	SPECIFIC DATE YYMMDD OR
96	(60)	CHARACTER	1	CALWEEKDAY	WEEK DAY 1=MONDAY 7=SUNDAY
97	(61)	CHARACTER	1	CALWEEKDAYM	BLANK IF WEEKDAY
102	(66)	CHARACTER	30	CALCOMM	COMMENT FOR THE WEEK DAY OR SPECIFIC DATE
132	(84)	CHARACTER	1	CALSTAT	WEEK DAY OR SPECIFIC DATE STATUS (W=WORK, F=FREE)
133	(85)	CHARACTER	15	*	SPARE

Cross reference

Name	Hex Offset	Hex Value	Level
CAL#SPEC	4A		2
CALCOMM	66		3
CALDAT	60		2
CALDATE	60		3
CALDATES	60		1
CALDESC	1A		2
CALENDAR	0		1
CALEYE	4		3
CALIDENT	A		2
CALKEY	2		2
calluts	58		2
CALNUM	2		3
CALSHIFT	4C		2
CALSTAT	84		3
CALTYPE	0		2
CALUPDD	40		2
CALUPDT	46		2
CALUPDU	38		2
CALVARS	60		2
CALVER	50		2
CALWEEKDAY	60		4
CALWEEKDAYM	61		4

CAPP - Completed/deleted application report record in DP

Name : DCLCAPP

Function:
Describes the layout of CAPP records representing report data on occurrences which have been completed/deleted in current plan. Occurrences that belong to one or more 24-hour periods that are automatically reported are sorted before occurrences which belong to the current 24-hours period. The records have the standard recsize and keysize of DPIN records.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	264	CAPAREA	CAP RECORD
0	(0)	CHARACTER	115	CAPKEY	KEY
0	(0)	SIGNED	2	CAPTYPE	TYPE IS ALWAYS 5
2	(2)	SIGNED	2	CAPSTYPE	SUBTYPE 1 = OLD 24-HOURS SUBTYPE 2 = CURRENT 24-H.
4	(4)	CHARACTER	34	*	ADJUST ADID
38	(26)	CHARACTER	26	CAPOCC	OCCURRENCE IDENTIFICATION
38	(26)	CHARACTER	16	CAPADID	APPLICATION IDENTIFICATION
54	(36)	CHARACTER	10	CAPOPIA	PLANNED OCC INPUT ARRIVAL
54	(36)	CHARACTER	6	CAPOPIAD	DAY (YYMMDD)
54	(36)	CHARACTER	4	CAPOPIAY	(YYMM)
58	(3A)	CHARACTER	2	CAPOPIAE	(DD)
60	(3C)	CHARACTER	4	CAPOPIAT	TIME (HHMM)
60	(3C)	CHARACTER	2	CAPOPIAH	TIME (HH)
62	(3E)	CHARACTER	2	CAPOPIAM	TIME (MM)
64	(40)	CHARACTER	10	CAPORC	REAL OCCURRENCE COMPLETION (BLANK FOR DELETED)
64	(40)	CHARACTER	6	CAPORCDT	DAY (YYMMDD) |BLANK
64	(40)	CHARACTER	4	CAPORCY	(YYMM)
68	(44)	CHARACTER	2	CAPORCD	(DD)
70	(46)	CHARACTER	4	CAPORCT	TIME (HHMM) |BLANK
70	(46)	CHARACTER	2	CAPORCH	TIME (HH)
72	(48)	CHARACTER	2	CAPORCM	TIME (MM)
74	(4A)	CHARACTER	7	CAPWSID	WSID OF OPERATION
74	(4A)	CHARACTER	3	CAPWSIDN	OPERATION NUMBER
77	(4D)	CHARACTER	4	CAPWSIDS	WORKSTATION NAME
81	(51)	CHARACTER	16	CAPOWID	OWNER OF OPERATION
97	(61)	CHARACTER	8	CAPJOBN	JOB NAME FOR OPERATION
105	(69)	CHARACTER	1	CAPADDED	HOW ADDED TO PLAN
106	(6A)	CHARACTER	9	*	TO FILL UP THE KEY
115	(73)	CHARACTER	149	CAPDATA	DATA PART OF CAP RECORD
115	(73)	CHARACTER	24	CAPOTXT	APPLICATION TEXT
139	(8B)	CHARACTER	1	CAPOPRI	PRIORITY
140	(8C)	CHARACTER	10	CAPOAIA	ACTUAL OCC INPUT ARRIVAL (BLANK FOR OCC DELETED BEFORE START)
140	(8C)	CHARACTER	6	CAPOAIAD	DAY (YYMMDD) |BLANK
140	(8C)	CHARACTER	4	CAPOAIAY	(YYMM)
144	(90)	CHARACTER	2	CAPOAIAE	(DD)
146	(92)	CHARACTER	4	CAPOAIAT	TIME (HHMM) |BLANK
146	(92)	CHARACTER	2	CAPOAIAH	TIME (HH)
148	(94)	CHARACTER	2	CAPOAIAM	TIME (MM)
150	(96)	CHARACTER	4	CAPAIDL	OCCURENCE INPUT DELAY (9959 FOR ALL ABOVE 100 H)
150	(96)	CHARACTER	2	CAPAIDLH	TIME (HH) |BLANK
152	(98)	CHARACTER	2	CAPAIDLM	(MM)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
154	(9A)	CHARACTER	4	CAPAODL	OCCURENCE OUTPUT DELAY (9959 FOR ALL ABOVE 100 H)
154	(9A)	CHARACTER	2	CAPAODLH	TIME (HH) |BLANK
156	(9C)	CHARACTER	2	CAPAODLM	(MM)
158	(9E)	CHARACTER	10	CAPOPC	PLANNED OCC. COMPLETION
158	(9E)	CHARACTER	6	CAPOPCDT	DAY (YYMMDD)
158	(9E)	CHARACTER	4	CAPOPCY	(YYMM)
162	(A2)	CHARACTER	2	CAPOPCD	(DD)
164	(A4)	CHARACTER	4	CAPOPCT	TIME (HHMM)
164	(A4)	CHARACTER	2	CAPOPCH	TIME (HH)
166	(A6)	CHARACTER	2	CAPOPCM	TIME (MM)
168	(A8)	CHARACTER	4	CAPOERR	OCCURENCE ERROR CODE
172	(AC)	CHARACTER	24	CAPOPTXT	OPERATION TEXT
196	(C4)	CHARACTER	10	CAPPPIA	SPEC OPER INPUT ARRIVAL
196	(C4)	CHARACTER	6	CAPPPIAD	DAY (YYMMDD)
196	(C4)	CHARACTER	4	CAPPPIAY	(YYMM)
200	(C8)	CHARACTER	2	CAPPPIAE	(DD)
202	(CA)	CHARACTER	4	CAPPPIAT	TIME (HHMM)
202	(CA)	CHARACTER	2	CAPPPIAH	TIME (HH)
204	(CC)	CHARACTER	2	CAPPPIAM	TIME (MM)
206	(CE)	CHARACTER	10	CAPPAIA	ACTUAL OPER INPUT ARRIVAL (BLANK FOR DELETED OPER BEFORE STARTED)
206	(CE)	CHARACTER	6	CAPPAIAD	DAY (YYMMDD) |BLANK
206	(CE)	CHARACTER	4	CAPPAIAY	(YYMM)
210	(D2)	CHARACTER	2	CAPPAIAE	(DD)
212	(D4)	CHARACTER	4	CAPPAIAT	TIME (HHMM) |BLANK
212	(D4)	CHARACTER	2	CAPPAIAH	TIME (HH)
214	(D6)	CHARACTER	2	CAPPAIAM	TIME (MM)
216	(D8)	CHARACTER	4	CAPOIDL	OPERATION INPUT DELAY (9959 FOR ALL ABOVE 100 H)
216	(D8)	CHARACTER	2	CAPOIDLH	TIME (HH) |BLANK
218	(DA)	CHARACTER	2	CAPOIDLM	(MM)
220	(DC)	CHARACTER	10	CAPPPC	SPECIFIC OPER COMPLETION
220	(DC)	CHARACTER	6	CAPPPCDT	DAY (YYMMDD)
220	(DC)	CHARACTER	4	CAPPPCY	(YYMM)
224	(E0)	CHARACTER	2	CAPPPCD	(DD)
226	(E2)	CHARACTER	4	CAPPPCT	TIME (HHMM)
226	(E2)	CHARACTER	2	CAPPPCH	TIME (HH)
228	(E4)	CHARACTER	2	CAPPPCM	TIME (MM)
230	(E6)	CHARACTER	10	CAPPRC	REAL OPERATION COMPLETION (BLANK FOR DELETED OPER)
230	(E6)	CHARACTER	6	CAPPRCT	DAY (YYMMDD) |BLANK
230	(E6)	CHARACTER	4	CAPPRCY	(YYMM)
234	(EA)	CHARACTER	2	CAPPRCD	(DD)
236	(EC)	CHARACTER	4	CAPPACT	TIME (HHMM) |BLANK
236	(EC)	CHARACTER	2	CAPPRCH	TIME (HH)
238	(EE)	CHARACTER	2	CAPPRCM	TIME (MM)
240	(F0)	CHARACTER	4	CAPOODL	OPERATION OUTPUT DELAY (9959 FOR ALL ABOVE 100 H)
240	(F0)	CHARACTER	2	CAPOODLH	TIME (HH) |BLANK
242	(F2)	CHARACTER	2	CAPOODLM	(MM)
244	(F4)	CHARACTER	4	CAPOPERR	OPERATION ERROR CODE
248	(F8)	CHARACTER	16	CAPOPUDATA	OPERATION USERDATA

Cross reference

Name	Hex Offset	Hex Value	Level
CAPADDED	69		3
CAPADID	26		4
CAPAIDL	96		3
CAPAIDLH	96		4
CAPAIDLM	98		4
CAPAODL	9A		3
CAPAODLH	9A		4
CAPAODLM	9C		4
CAPAREA	0		1
CAPDATA	73		2
CAPJOBN	61		3
CAPKEY	0		2
CAPOAIA	8C		3
CAPOAIAD	8C		4
CAPOAIAE	90		5
CAPOAIAH	92		5
CAPOAIAM	94		5
CAPOAIAT	92		4
CAPOAIAY	8C		5
CAPOCC	26		3
CAPOERR	A8		3
CAPOIDL	D8		3
CAPOIDLH	D8		4
CAPOIDLM	DA		4
CAPOODL	F0		3
CAPOODLH	F0		4
CAPOODLM	F2		4
CAPOPC	9E		3
CAPOPCD	A2		5
CAPOPCDT	9E		4
CAPOPCH	A4		5
CAPOPCM	A6		5
CAPOPCT	A4		4
CAPOPCY	9E		5
CAPOPERR	F4		3
CAPOPIA	36		4
CAPOPIAD	36		5
CAPOPIAE	3A		6

Name	Hex Offset	Hex Value	Level
CAPOPIAH	3C		6
CAPOPIAM	3E		6
CAPOPIAT	3C		5
CAPOPIAY	36		6
CAPOPRI	8B		3
CAPOPTXT	AC		3
CAPOPUDATA	F8		3
CAPORC	40		3
CAPORCD	44		5
CAPORCDT	40		4
CAPORCH	46		5
CAPORCM	48		5
CAPORCT	46		4
CAPORCY	40		5
CAPOTXT	73		3
CAPOWID	51		3
CAPPACT	EC		4
CAPPAIA	CE		3
CAPPAIAD	CE		4
CAPPAIAE	D2		5
CAPPAIAH	D4		5
CAPPAIAM	D6		5
CAPPAIAT	D4		4
CAPPAIAY	CE		5
CAPPPC	DC		3
CAPPPCD	E0		5
CAPPPCDT	DC		4
CAPPPCH	E2		5
CAPPPCM	E4		5
CAPPPCT	E2		4
CAPPPCY	DC		5
CAPPPIA	C4		3
CAPPPIAD	C4		4
CAPPPIAE	C8		5
CAPPPIAH	CA		5
CAPPPIAM	CC		5
CAPPPIAT	CA		4
CAPPPIAY	C4		5
CAPPRC	E6		3
CAPPRCD	EA		5
CAPPRCH	EC		5

Name	Hex Offset	Hex Value	Level
CAPPRCM	EE		5
CAPPRCT	E6		4
CAPPRCY	E6		5
CAPSTYPE	2		3
CAPTYPE	0		3
CAPWSID	4A		3
CAPWSIDN	4A		4
CAPWSIDS	4D		4

CASE - Auto recovery case code list

Name : DCLCASE

Function:
A case code is set up to represent one or more error codes or return codes. A function may use this by allowing a case code to be given instead of one or more error or return codes, the case code will then represent all the codes in its list. The list is not recursive, a code in the list is not regarded as another case code. The lists are placed adjacent in storage, the start address of list i+1 is the end address of list i.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CASELIST	CASE CODE LIST
0	(0)	CHARACTER	4	CASECODE	CODE REPRESENTING THIS LIST
4	(4)	ADDRESS	4	CASENEXT	ADDRESS TO NEXT LIST
8	(8)	CHARACTER	4	CASEMEMB (*)	MEMBERS, THE CODES COLLECTIVELY

CPLREC01 - Current plan header record

Name : DCLCPR01

Function:
This segment declares the current plan header record, physically located in EQQCPxDS and EQQNCPDS.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	190	CPLREC01	CURRENT PLAN HDR RECORD
0	(0)	CHARACTER	12	CPLKEY01	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 01
2	(2)	CHARACTER	10	CPLKF01	SHOULD CONTAIN BLANKS
12	(C)	CHARACTER	178	CPLBDY01	HDR RECORD BODY
12	(C)	CHARACTER	4	CPLEYE01	EYECATCHER
16	(10)	UNSIGNED	1	CPLVERS01	VERSION NUMBER
17	(11)	CHARACTER	1	*	FREE
18	(12)	CHARACTER	10	CPLCR	CURRENT PLAN CREATION
18	(12)	CHARACTER	6	CPLCRDAT	DATE
24	(18)	CHARACTER	4	CPLCRTIM	TIME

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
28	(1C)	CHARACTER	10	CPLEND	LAST PLAN NEXT PERIOD END
28	(1C)	CHARACTER	6	CPLENDAT	DATE
34	(22)	CHARACTER	4	CPLENTIM	TIME
38	(26)	CHARACTER	10	CPLBU	LAST BACKUP
38	(26)	CHARACTER	6	CPLBUDAT	DATE
44	(2C)	CHARACTER	4	CPLBUTIM	TIME
48	(30)	CHARACTER	10	CPLLP	TAIL END
48	(30)	CHARACTER	6	CPLLPDAT	DATE
54	(36)	CHARACTER	4	CPLLPTIM	TIME
58	(3A)	CHARACTER	10	CPLREPS	REPORT PERIOD START
58	(3A)	CHARACTER	6	CPLREPSD	DATE
64	(40)	CHARACTER	4	CPLREPST	TIME
68	(44)	SIGNED	4	CPLREPL	NUMBER OF REPLANS SINCE LAST PLAN NEXT
72	(48)	SIGNED	4	CPLOPS	TOTAL NUMBER OF OPERATIONS RECORDS
76	(4C)	SIGNED	4	CPLMAXOP	THE TOTAL NUMBER OF POSSIBLE CONCURRENT OPS
80	(50)	SIGNED	4	CPLMAXOC	CURRENT MAX OCCURRENCE RECORD NUMBER
84	(54)	SIGNED	4	CPLMAXNW	CURRENT MAX NETWORK NUMBER
88	(58)	SIGNED	4	CPLMAX06	CURRENT MAX TYPE 06 RECORD
92	(5C)	SIGNED	4	CPLMAX07	CURRENT MAX TYPE 07 RECORD
96	(60)	SIGNED	4	CPLMIN08	CURRENT MIN TYPE 08 RECORD
100	(64)	CHARACTER	32	cpl01Counts	Main counters CUTA
100	(64)	SIGNED	4	cpl01C_PIFadd	&period..num PIF occ ADDs CUTA
104	(68)	SIGNED	4	cpl01C_MCPadd	&period..num MCP occ ADDs CUTA
108	(6C)	SIGNED	4	cpl01C_ETTadd	&period..num ETT occ ADDs CUTA
112	(70)	SIGNED	4	cpl01C_ARCadd	&period..num AR occ ADDs CUTA
116	(74)	SIGNED	4	cpl01C_copOP	&period..num copied opers
120	(78)	SIGNED	4	cpl01C_copOCC	&period..num copied OCCs
124	(7C)	SIGNED	4	cpl01C_critjobs	nbr of critical jobs CRFA
128	(80)	SIGNED	4	cpl01C_critpred	total num of crit preds CRFC
132	(84)	CHARACTER	8	cpl01token_full	operation full token
132	(84)	CHARACTER	4	cpl01token_pre	prefix
136	(88)	UNSIGNED	4	cpl01token	opctoken
140	(8C)	CHARACTER	4	cpl01bugmtd	last backup date, gmt
144	(90)	UNSIGNED	4	cpl01bugmtt	last backup time, gmt
148	(94)	CHARACTER	10	cpllevel	level of current plan ds
148	(94)	CHARACTER	2	cpllev_ver	verion of cp ds
150	(96)	CHARACTER	8	cpllev_fmid	fmid of cp ds
158	(9E)	CHARACTER	1	cplDeadl	I = ignore deadline DCRC
159	(9F)	CHARACTER	1	cplTDepChk	Y = new calc of PST DCRA
160	(A0)	SIGNED	4	cplmaxxnw	current max x net nbr
164	(A4)	CHARACTER	1	cplfnshift	SHIFT FNONC used (YN)
165	(A5)	CHARACTER	25	*	x3a,L9C,LVC,X0C,41C NETWORK NUMBER

Cross reference

Name	Hex Offset	Hex Value	Level
CPLOPS	48		3

Name	Hex Offset	Hex Value	Level
CPLREPL	44		3
CPLBDY01	C		2
CPLBU	26		3
CPLBUDAT	26		4
CPLBUTIM	2C		4
CPLCR	12		3
CPLCRDAT	12		4
CPLCRTIM	18		4
cplDeadl	9E		3
CPLEND	1C		3
CPLENDAT	1C		4
CPLENTIM	22		4
CPLEYE01	C		3
cplfnshift	A4		3
CPLKEY01	0		2
CPLKF01	2		3
cpllev_fmid	96		4
cpllev_ver	94		4
cpllevel	94		3
CPLLP	30		3
CPLLPDAT	30		4
CPLLPTIM	36		4
CPLMAXNW	54		3
CPLMAXOC	50		3
CPLMAXOP	4C		3
cplmaxxnw	A0		3
CPLMAX06	58		3
CPLMAX07	5C		3
CPLMIN08	60		3
CPLREC01	0		1
CPLREPS	3A		3
CPLREPSD	3A		4
CPLREPST	40		4
cplTDepChk	9F		3
CPLVERS01	10		3
cpl01bugmtd	8C		3
cpl01bugmtt	90		3
cpl01C_ARCadd	70		4
cpl01C_copOCC	78		4
cpl01C_copOP	74		4
cpl01C_critjobs	7C		4

Name	Hex Offset	Hex Value	Level
cpl01C_critpred	80		4
cpl01C_ETTadd	6C		4
cpl01C_MCPadd	68		4
cpl01C_PIFadd	64		4
cpl01Counts	64		3
cpl01token	88		4
cpl01token_full	84		3
cpl01token_pre	84		4

CPLREC02 - Current plan workstation record

Name : DCLCPR02

Function:
This segment declares a current plan workstation record, physically located in EQQCPxDS
and EQQNCPDS.
The CPLREC02 record is structured as follows:
 1 CPLREC02
  2 common data
  2 open time interval ( )
  2 access method ( )

Common data segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC02	CURRENT PLAN WS RECORD
0	(0)	CHARACTER	236	CPLCOM02	
0	(0)	CHARACTER	19	CPLKEY02	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 02
2	(2)	CHARACTER	4	CPLWSN02	WORKSTATION NAME
6	(6)	CHARACTER	13	CPLKF02	SHOULD CONTAIN BLANKS
19	(13)	CHARACTER	193	CPLBDY02	WS RECORD BODY
19	(13)	CHARACTER	4	CPLEYE02	EYECATCHER
23	(17)	UNSIGNED	1	CPLVERS02	VERSION NUMBER
24	(18)	CHARACTER	32	CPLDESCR02	WS DESCRIPTION
56	(38)	CHARACTER	12	CPLSUMC	SUM OF COMPLETED OPS
56	(38)	SIGNED	4	CPLSUMC#	NUMBER
60	(3C)	SIGNED	4	CPLSUMCE	ESTIMATED DURATION
64	(40)	SIGNED	4	CPLSUMCR	REAL DURATION
68	(44)	CHARACTER	12	CPLSUMI	SUM OF INTERRUPTED OPS
68	(44)	SIGNED	4	CPLSUMI#	NUMBER
72	(48)	SIGNED	4	CPLSUMIE	ESTIMATED DURATION
76	(4C)	SIGNED	4	CPLSUMIR	REAL DURATION
80	(50)	CHARACTER	8	CPLSUMS	SUM OF STARTED OPS
80	(50)	SIGNED	4	CPLSUMS#	NUMBER
84	(54)	SIGNED	4	CPLSUMSE	ESTIMATED DURATION
88	(58)	CHARACTER	8	CPLSUMR	SUM OF 'READY' OPS (R,A,*)
88	(58)	SIGNED	4	CPLSUMR#	NUMBER
92	(5C)	SIGNED	4	CPLSUMRE	ESTIMATED DURATION
96	(60)	CHARACTER	8	CPLSUMW	SUM OF WAITING OPS

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
96	(60)	SIGNED	4	CPLSUMW#	NUMBER
100	(64)	SIGNED	4	CPLSUMWE	ESTIMATED DURATION
104	(68)	SIGNED	4	CPLSUMA	SUM OF ARRIVING OPS
108	(6C)	SIGNED	4	CPLSUMNRR	SUM OF '*' READY
112	(70)	SIGNED	4	CPLSUMU	SUM OF UNDECIDED
116	(74)	SIGNED	4	CPLSUME	SUM OF ERROR OPERS
120	(78)	SIGNED	4	CPLSUML	SUM OF LATE OPERS
124	(7C)	CHARACTER	8	CPLCREVT	LAST SYNCHRONIZATION EVENT CREATION TIME
124	(7C)	SIGNED	4	CPLDATE	DATE FORMAT (00YYDDDF)
128	(80)	SIGNED	4	CPLTIME	TIME FORMAT (SECS*100)
132	(84)	SIGNED	4	*(3)	RESERVED FOR DURATION
144	(90)	CHARACTER	1	CPLWSTYP	1=GENERAL 2=COMPUTER 3=PRINT
145	(91)	CHARACTER	1	CPLWSREP	1=AUTO 2=MANUAL 3=MANUAL & COMPLETION ONLY 4=NONREP
146	(92)	CHARACTER	2	CPLWSR1N	RESOURCE NAME
148	(94)	SIGNED	2	CPLWSR1IU#	NUMBER IN USE (NOTE, PS'S ..IN USE = NO. STARTED OP'S
150	(96)	CHARACTER	2	CPLWSR1FLG	RESOURCE INDICATORS
		1...		CPLWSR1C	RESOURCE USED AT CONTROL
		.1..		*	FREE
		.1..		CPLWSWAIT	WAIT WORKSTATION
		..1.		CPLWSVIRT	VIRTUAL WORKSTATION
		...1		CPLWSVIRT	VIRTUAL WORKSTATION
		.... 1...		CPLWSZCAGENT	Z/OS CENTRIC DISTRIBUTED WORKSTATION
		.... .1..		CPLWSZNOINFO	Z-CENTRIC NO INFO AVAILABLE
		.... ..1.		CPLWSDYN	DYNAMIC WORKSTATION
150	(96)	BITSTRING	1	*	FREE
152	(98)	CHARACTER	2	CPLWSR2N	RESOURCE NAME
154	(9A)	UNSIGNED	2	CPLWSR2IU#	NUMBER IN USE
156	(9C)	CHARACTER	2	CPLWSR2FLG	RESOURCE INDICATORS
		1...		CPLWSR2C	RESOURCE USED AT CONTROL
		.1..		CPLWSSTAT	WSSTAT GENERATED STATUS
		..1.		CPLUX009	UX009 GENERATED STATUS
156	(9C)	BITSTRING	1	*	RESERVED
158	(9E)	CHARACTER	1	CPLRLTYP	READY LIST TYPE
159	(9F)	CHARACTER	1	CPLFLG02	FLAG BITS
		1...		CPLPREP	JOB SETUP ABILITY
		.1..		CPLIVLNU	IVL NOT USED AT ALL
		..1.		CPLIVLNP	NO PARALLEL SERVERS
		...1		CPLSTC	STARTED TASK SUPPORT
		.... 1...		CPLWTO	WTO DEADLINE SUPPORT
		.... .1..		CPLOFF_PEND	WS IS PENDING OFFLINE
		.... ..1.		CPLRTR_PEND	T EVENT PENDING
		1		CPLAWS_VARY	VARIED AWS SET
160	(A0)	CHARACTER	8	CPLPREEV	PREVIOUS EVENT
160	(A0)	CHARACTER	4	CPLPREDT	DATE IN TIME MACRO FORM
164	(A4)	CHARACTER	4	CPLPRETM	TIME IN TIME MACRO FORM
168	(A8)	CHARACTER	8	CPLSUDS	DDNAME OF SUBM./REL DS

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
176	(B0)	SIGNED	2	CPLIVL#I	NUMBER OF OPEN INTERVALS
178	(B2)	SIGNED	2	CPLMAX15	MAX NO OF EVENTS IN 15 MIN.
180	(B4)	CHARACTER	1	CPLWSSTA	USED BY DIALOG
181	(B5)	CHARACTER	1	CPLCWS_STAT	WORKSTATION STATUS
182	(B6)	SIGNED	2	CPLTRT02	TRANSPORT TIME DEFAULT
184	(B8)	CHARACTER	8	CPLCOFF02	OFFLINE TIME STAMP
184	(B8)	CHARACTER	4	CPLCOFF_DATE	OFFLINE DATE
188	(BC)	CHARACTER	4	CPLCOFF_TIME	OFFLINE TIME
192	(C0)	CHARACTER	4	CPLCAWS02	CURRENT ALTERNATE WS
196	(C4)	UNSIGNED	2	CPLSSEQ#02	CURRENT SUBMIT SEQUENCE
198	(C6)	UNSIGNED	1	CPLRSEQ#02	CURRENT REQUEST SEQUENC
199	(C7)	UNSIGNED	1	CPLSUMCES	DURATION FIELDS ABOVE
200	(C8)	UNSIGNED	1	CPLSUMCRS	HOLDS NUMBER OF MINUTES
201	(C9)	UNSIGNED	1	CPLSUMIES	OF ESTIMATED OR REAL
202	(CA)	UNSIGNED	1	CPLSUMIRS	DURATION FOR ALL OPERATIONS
203	(CB)	UNSIGNED	1	CPLSUMSES	ON THIS WORKSTATION PER
204	(CC)	UNSIGNED	1	CPLSUMRES	STATUS (C,I,S,R AND W)
205	(CD)	UNSIGNED	1	CPLSUMWES	HERE ARE THE SECONDS
206	(CE)	CHARACTER	1	CPLWSTWS	FTA WORKSTATION
207	(CF)	CHARACTER	1	CPLLNK02	FTA WS LINKED STATUS
208	(D0)	CHARACTER	1	CPLFLGTWS	FTA SPECIFIC FLAGS
				CPLLNKVA	LINKED STATUS MANUALLY VARIED
				CPLSTAVA	STATUS MANUALLY VARIED
				CPLLNKDW	LINKED DOWN TO CPU
				CPLWRTST	WRITER STATUS FOR CPU
				CPLCMDLNK	COMMAND LINK FOR FTW: ON=LINK OFF=UNLINK
				CPLCMDSTA	COMMAND STATUS FOR FTW: ON=START OFF=STOP
				CPLFULLNK	FTA WS FULLY LINKED STATUS: ON=FULLY LINKED OFF=NOT FULLY LINKED
209	(D1)	CHARACTER	1	CPLAUTO	SYSTEM AUTOMATION Y|N
210	(D2)	SIGNED	2	CPLACCM#	0='NO ACC MET DATA' 1='ACC MET DATA PRESENT
212	(D4)	SIGNED	2	CPLLIMIT	CPU LIMIT VALUE
214	(D6)	CHARACTER	22	CPLFULLYACT	Y/N
215	(D7)	CHARACTER	1	CPLRENGT	REMOTE ENGINE TYPE
216	(D8)	SIGNED	4	CPLVMAX15	MAXNO EVENT IN 15 MIN
220	(DC)	SIGNED	4	CPLSUMD	SUM OF DELETED OPERATIONS
224	(E0)	SIGNED	4	CPLSUMH	SUM OF MANUALLY HELD OPERATIONS
228	(E4)	SIGNED	2	CPLEOPT#	0='NO E2E OPTIONS DATA' 1='E2E OPTIONS PRESENT
230	(E6)	CHARACTER	6	*	FREE
236	(EC)	CHARACTER	*	CPLWSVAR	VARIABLE LENGTH SECTION

Open time interval segment

The value of CPLIVL#I in the common data segment describes the number of open time interval segments that follow the common data.

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
236	(EC)	STRUCTURE	48	CPLIVLOI (*)	
236	(EC)	CHARACTER	10	CPLIVLFR	INTERVAL START
236	(EC)	CHARACTER	6	CPLIVLFD	DATE YYMMDD
242	(F2)	CHARACTER	4	CPLIVLFT	TIME HHMM
246	(F6)	CHARACTER	10	CPLIVLTO	INTERVAL END
246	(F6)	CHARACTER	6	CPLIVLTD	DATE YYMMDD
252	FC)	CHARACTER	4	CPLIVLTT	TIME HHMM
256	(100)	UNSIGNED	2	CPLIVL#PS	MAX PARALLEL SERVERS
258	(102)	UNSIGNED	2	CPLIVL#DPPS	SAME, SET AT DAILY PLANNING
260	(104)	BITSTRING	1	CPLIVLF2	FLAGS
		1...		CPLIVLMC	IVL SET BY MCP
		.1..		CPLIVLDP	IVL SET BY DP (WSD)
		..11 1111		*	FREE
261	(105)	CHARACTER	1	*	FREE
262	(106)	UNSIGNED	2	CPLIVL#R1	CURRENT RESOURCE CAPACITY
264	(108)	UNSIGNED	2	CPLIVL#DPR1	CAPACITY SET AT DAILY PLAN
266	(10A)	UNSIGNED	2	CPLIVL#R2	CURRENT RESOURCE CAPACITY
268	(10C)	UNSIGNED	2	CPLIVL#DPR2	CAPACITY SET AT DAILY PLAN
270	(10E)	CHARACTER	4	CPLIVLAWS	ALTERNATE WS NAME
274	(112)	CHARACTER	4	CPLIVLDPAWS	ALT WS BY DP
278	(116)	CHARACTER	6	*	FREE

Access method segment

You can have the access method segment only if CPLACCM# is greater than 0 in the common data segment.

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	72	CPLACC (*)	
0	(0)	CHARACTER	12	CPLACCMN	ACCESS METHOD NAME
12	(C)	CHARACTER	52	CPLADR	SAP HOST ADDRESS
12	(C)	CHARACTER	26	CPLADR1	SAP ADDR 1ST PART
38	(26)	CHARACTER	26	CPLADR2	SAP ADDR 2ND PART
64	(40)	SIGNED	4	CPLPORT	SAP PORT NUMBER
68	(44)	CHARACTER	4	*	FREE

E2E Options Data

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	175	CPLEOPT (*)	
0	(0)	CHARACTER	47	CPLEOJU	DEFAULT JOB USER
47	(2F)	CHARACTER	1	CPLEOJP	DEFAULT JOB PASSWORD
48	(30)	CHARACTER	40	CPLEOJT	DEFAULT JOB PASSWORD
88	(58)	CHARACTER	1	CPLEOBR	IS A BROKER (Y/N)
89	(59)	CHARACTER	40	CPLEOPL	BROKER POOL
129	(81)	CHARACTER	40	CPLEODP	BROKER DYNAMIC POOL

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
169	(A9)	CHARACTER	7	*	FREE

Cross reference

Name	Hex Offset	Hex Value	Level
CPLACC	0		1
CPLACCM#	D2		4
CPLACCMN	0		2
CPLADR	C		2
CPLADR1	C		3
CPLADR2	26		3
CPLAUTO	D1		4
CPLAWS_VARY	9F	01	5
CPLBDY02	13		3
CPLCAWS02	C0		4
CPLCMDLNK	D0	08	5
CPLCMDSTA	D0	04	5
CPLCOFF_DATE	B8		5
CPLCOFF_TIME	BC		5
CPLCOFF02	B8		4
CPLCOM02	0		2
CPLCREVT	7C		4
CPLCWS_STAT	B5		4
CPLDATE	7C		5
CPLDESCR02	18		4
CPLEOBR	58		2
CPLEODP	81		2
CPLEOJP	2F		2
CPLEOJT	30		2
CPLEOJU	0		2
CPLEOPL	59		2
CPLEOPT	0		1
CPLEOPT#	E4		4
CPLEYE02	13		4
CPLFLGTWS	D0		4
CPLFLG02	9F		4
CPLFULLNK	D0	02	5
CPLFULLYACT	D6		4
CPLIVL#DPPS	102		2
CPLIVL#DPR1	108		2

Name	Hex Offset	Hex Value	Level
CPLIVL#DPR2	10C		2
CPLIVL#I	B0		4
CPLIVL#PS	100		2
CPLIVL#R1	106		2
CPLIVL#R2	10A		2
CPLIVLAWS	10E		2
CPLIVLDP	104	40	3
CPLIVLDPAWS	112		2
CPLIVLFD	EC		3
CPLIVLFR	EC		2
CPLIVLFT	F2		3
CPLIVLF2	104		2
CPLIVLMC	104	80	3
CPLIVLNP	9F	20	5
CPLIVLNU	9F	40	5
CPLIVLOI	EC		1
CPLIVLTD	F6		3
CPLIVLTO	F6		2
CPLIVLTT	FC		3
CPLKEY02	0		3
CPLKF02	6		4
CPLLIMIT	D4		4
CPLLNKDW	DO	20	5
CPLLNKVA	D0	80	5
CPLLNK02	CF		4
CPLMAX15	B2		4
CPLOFF_PEND	9F	04	5
CPLPORT	40		2
CPLPREDT	A0		5
CPLPREEV	A0		4
CPLPREP	9F	80	5
CPLPRETM	A4		5
CPLREC02	0		1
CPLRENGT	D7		4
CPLRLTYP	9E		4
CPLRSEQ#02	C6		4
CPLRTR_PEND	9F	02	5
CPLSSEQ#02	C4		4
CPLSTAVA	D0	40	5
CPLSTC	9F	10	5
CPLSUDS	A8		4

Name	Hex Offset	Hex Value	Level
CPLSUMA	68		4
CPLSUMC	38		4
CPLSUMC#	38		5
CPLSUMCE	3C		5
CPLSUMCES	C7		4
CPLSUMCR	40		5
CPLSUMCRS	C8		4
CPLSUMD	DC		4
CPLSUME	74		4
CPLSUMH	EO		4
CPLSUMI	44		4
CPLSUMI#	44		5
CPLSUMIE	48		5
CPLSUMIES	C9		4
CPLSUMIR	4C		5
CPLSUMIRS	CA		4
CPLSUML	78		4
CPLSUMNRR	6C		4
CPLSUMR	58		4
CPLSUMR#	58		5
CPLSUMRE	5C		5
CPLSUMRES	CC		4
CPLSUMS	50		4
CPLSUMS#	50		5
CPLSUMSE	54		5
CPLSUMSES	CB		4
CPLSUMU	70		4
CPLSUMW	60		4
CPLSUMW#	60		5
CPLSUMWE	64		5
CPLSUMWES	CD		4
CPLSUMX	84		4
CPLTIME	80		5
CPLTRT02	B6		4
CPLUX009	9C	20	5
CPLVERS02	17		4
CPLVMAX15	D8		4
CPLWRTST	D0	10	5
CPLWSDYN	96	02	5
CPLWSN02	2		4
CPLWSREP	91		4

Name	Hex Offset	Hex Value	Level
CPLWSR1C	96	80	5
CPLWSR1FLG	96		4
CPLWSR1IU#	94		4
CPLWSR1N	92		4
CPLWSR2C	9C	80	5
CPLWSR2FLG	9C		4
CPLWSR2IU#	9A		4
CPLWSR2N	98		4
CPLWSSTA	B4		4
CPLWSSTAT	9C	40	5
CPLWSTWS	CE		4
CPLWSTYP	90		4
CPLWSVAR	EC		2
CPLWSVIRT	96	10	5
CPLWSWAIT	96	20	5
CPLWSZCAGENT	96	08	5
CPLWTO	9F	08	5

CPLREC04 - Current plan job name table record

Name : DCLCPR04

Function:
This segment declares a jobname table record, physically located in EQQCPxDS and EQQNCPDS.
The CPLREC04 record is structured as follows:

```
    1 CPLREC04
       2 common data
       2 jnt entries ( )
       2 step entries ( )
```

Common data segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC04	JOBNAME TABLE RECORD
0	(0)	CHARACTER	19	CPLKEY04	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 04
2	(2)	CHARACTER	8	CPLJOB04	JOBNAME
10	(A)	CHARACTER	5	CPLLUT04	LATEST OUT OF CPU OP
10	(A)	UNSIGNED	1	CPLLUY04	YEAR
11	(B)	UNSIGNED	1	CPLLUM04	MONTH
12	(C)	UNSIGNED	1	CPLLUD04	DAY
13	(D)	SIGNED	2	CPLLUI04	TIME IN MIN. (HH*100+MM)
15	(F)	CHARACTER	4	CPLCID04	CPU OPERATION IDENTIFICAT-N
15	(F)	UNSIGNED	3	CPLCOC04	OCCURRENCE NUMBER
18	(12)	UNSIGNED	1	CPLCOP04	OPERATION NUMBER
19	(13)	CHARACTER	77	CPLBDY04	BODY OF JOBNAME TABLE
19	(13)	CHARACTER	4	CPLEYE04	EYECATCHER

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
23	(17)	UNSIGNED	1	CPLVERS04	VERSION NUMBER
24	(18)	SIGNED	2	CPLASID04	ASID
26	(1A)	SIGNED	2	CPLNET04	NETWORK ID
28	(1C)	CHARACTER	8	CPLJES04	JES2 OR JES3 JOB ID
36	(24)	SIGNED	4	CPLRDD04	SMF READER DATE
40	(28)	SIGNED	4	CPLRDT04	SMF READER TIME
44	(2C)	SIGNED	4	CPL#RC04	NUMBER OF JNT ENTRIES
48	(30)	CHARACTER	1	CPLJST04	JOB STATUS H=HELD, Q=ONQUE
49	(31)	CHARACTER	1	CPLSYS04	SYSTEM A=JES2, B=JES3
50	(32)	CHARACTER	1	CPLFLG04	STATUS FLAGS
		1...		CPLAJR04	ON = RELEASE JOB IF HELD
		.1..		*	DO NOT USE
		..1.		CPLERR04	ON = ERROR MESSAGE ISSUED
		...1		CPLTJT04	ON=CPU OP IS TIME JOB
		.... 1...		CPLAEC04	ON=AUTOMATIC ERROR COMPL.
		.... .1..		CPL3SJ04	ON=JOB HAS ENDED
		.... ..1.		CPL2EV04	ON=JOB HAS STARTED
		1		CPLSOO04	SUBMITTED OUTSIDE THE SCHEDULER
51	(33)	CHARACTER	1	CPLFLG042	MORE STATUS FLAGS
		1...		CPLAJS04	ON = AUTOMATIC SUBMIT OFF= DO NOT SUBMIT
		.1..		CPLOTO04	ON = OVERRIDE TRACK OPTION IN EV.MATCHING
		..1.		CPLRST04	JOB HAS BEEN RESTARTED
		...1		CPLODL04	CORRESP OPER DELETED
		.... 1...		CPLSPO04	SPIN OFF DS SENT ON NJE
		.... .1..		CPLPUR04	B5 EVENT SKIPPED
		.... ..1.		CPLNJ#04	NO JOBNUMBER FROM EXI
		1		CPLASET04	ERRRES IN ACTION
52	(34)	BITSTRING	1	CPLFLG043	FLAGS
		1...		CPLWRER04	ON=CPU OP REROUTED
		.1..		CPLRERUT04	ON=CPU OP REROUTEABLE
		..1.		CPLDIRER04	ON=INST PARM REREOTE
		...1		CPLRESTA04	ON=CPU OP RESTARTABLE
		.... 1...		CPLDIRES04	ON=INST PARM RESTART
		.... .1..		CPLRDR04	SAME JOBNAME/RDR TIME
		11		CPLETT04	ADDED VIA ETT
				*	RESERVED
53	(35)	CHARACTER	1	*	RESERVED
54	(36)	SIGNED	2	CPLGMT04	GMT OFFSET FOR RDR EVT
56	(38)	SIGNED	4	CPLORG04	NJE ORIGIN JOB NUMBER
60	(3C)	CHARACTER	8	CPLSUB04	SUBMIT DATE AND TIME
60	(3C)	CHARACTER	4	CPLSUD04	DATE (FORMAT 00YYDDDF)
64	(40)	CHARACTER	4	CPLSUT04	TIME (FORMAT 100*SECS)
68	(44)	CHARACTER	4	CPLERC04	ER CODE FROM 3SJ EVENT
72	(48)	SIGNED	4	CPL#SC04	NO. STEP COMP. ENTRIES
76	(4C)	SIGNED	4	CPLABN04	INDEX OF ABENDING STEP
80	(50)	BITSTRING	1	CPLARB04	
		1...		CPLARF04	AUTO RECOVER, 1ST TIME
		.1..		CPLBSN04	IMPROPER STEP NAMING
		..1.		CPLMAS04	FOR TERMINATING STEP
		...1		CPLMSS04	ALL FOR A JS REC. STEP
		.... 1...		CPLMIS04	AT LEAST ONE

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
		.... .1..		CPLSND04	STEP END ENTRY AVAILABLE
		.... ..1.		CPLEOSYS04	ERROR SET BY WS FAIL
		1		CPLEOFFL04	ERROR SET BY WS OFFL
81	(51)	CHARACTER	8	CPLNOD04	CURRENT NJE NODE NAME
89	(59)	CHARACTER	4	CPLAWS04	REROUTED ALT WS NAME
93	(5D)	CHARACTER	1	*	STEP EVENTS
		1...		CPLSNZ04	STEP EVENTS NZERO FLAG
		.1..		CPLSTALL04	STEPEVENTS(ALL)
		..1.		CPLWLMAR	ALREADY HPWLM CLASS RQ
		...1 1111		*	FREE
94	(5E)	SIGNED	2	CPLARSTE04	AR RESTART STEP NUMBER
96	(60)	CHARACTER	*	CPLVAR04	VARIABLE PART OF RECORD

JNT entries segment

The value of CPL#RC04 in the common data segment describes the number of JNT entries segments.

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
96	(60)	STRUCTURE	*	CPLENS04	JNT ENTRIES
96	(60)	CHARACTER	20	CPLENT04 (*)	JNT ENTRY
96	(60)	CHARACTER	4	CPLWSN04	WORKSTATION NAME
100	(64)	CHARACTER	4	CPLOPX04	INDEX TO OPER RECORD
100	(64)	UNSIGNED	3	CPLOCC04	OCC NUMBER
103	(67)	UNSIGNED	1	CPLOP04	OPERATION NUMBER
104	(68)	CHARACTER	1	CPLCLS04	JOB OR SYSOUTCLASS
105	(69)	CHARACTER	8	CPLFRM04	FORMNUMBER (PRINT WS)
105	(69)	CHARACTER	8	CPLNJE04	ORIGIN NJE NODE (CPU WS)
113	(71)	CHARACTER	1	CPLOST04	OPERATION STATUS
114	(72)	CHARACTER	2	*	FREE

Step entries segment

The value of CPL#SC04 in the common data segment describes the number of step entries segments.

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	*	CPLSTP04	
0	(0)	CHARACTER	20	CPLSMF04 (*)	
0	(0)	CHARACTER	8	CPLSCP04	NAME OF STEP CALLING PROC.
		1...		*	RESERVED
		.1..		CPLSUR04	POSITIVE IDENTIFICATION
8	(8)	CHARACTER	8	CPLPSN04	PROCEDURE STEP NAME

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
16	(10)	CHARACTER	4	CPLSCC04	STEP COMPLETION CODE FROM SMF KEPT HERE UNTIL JOB ENDS. IT HAS THE FOLLOWING FORMAT: 'SXXX' (SYSTEM ABEND) 'UXXX' (USER ABEND) 'FLSH' (STEP FLUSHED) 'XXXX' (RETURN CODE)

Cross reference

Name	Hex Offset	Hex Value	Level
CPL#RC04	2C		3
CPL#SC04	48		3
CPLABN04	4C		3
CPLAEC04	32	08	4
CPLAJR04	32	80	4
CPLAJS04	33	80	4
CPLARB04	50		3
CPLARF04	50	80	4
CPLARSTE04	5E		3
CPLASET04	33	01	4
CPLASID04	18		3
CPLAWS04	59		3
CPLBDY04	13		2
CPLBSN04	50	40	4
CPLCID04	F		3
CPLCLS04	68		3
CPLCOC04	F		4
CPLCOP04	12		4
CPLDIRER04	34	20	4
CPLDIRES04	34	08	4
CPLENS04	60		1
CPLENT04	60		2
CPLEOFFL04	50	01	4
CPLEOSYS04	50	02	4
CPLERC04	44		3
CPLERR04	32	20	4
CPLETT04	34	02	4

Name	Hex Offset	Hex Value	Level
CPLEYE04	13		3
CPLFLG04	32		3
CPLFLG042	33		3
CPLFLG043	34		3
CPLFRM04	69		3
CPLGMT04	36		3
CPLJES04	1C		3
CPLJOB04	2		3
CPLJST04	30		3
CPLKEY04	0		2
CPLLUD04	C		4
CPLLUI04	D		4
CPLLUM04	B		4
CPLLUT04	A		3
CPLLUY04	A		4
CPLMAS04	50	20	4
CPLMIS04	50	08	4
CPLMSS04	50	10	4
CPLNET04	1A		3
CPLNJ#04	33	02	4
CPLNJE04	69		4
CPLNOD04	51		3
CPLOCC04	64		4
CPLODL04	33	10	4
CPLOPX04	64		3
CPLOP04	67		4
CPLORG04	38		3
CPLOST04	71		3
CPLOTO04	33	40	4
CPLPSN04	8		3
CPLPUR04	33	04	4
CPLRDD04	24		3
CPLRDR04	34	04	4
CPLRDT04	28		3
CPLREC04	0		1
CPLRERUT04	34	40	4
CPLRESTA04	34	10	4
CPLRST04	33	20	4
CPLSCC04	10		3
CPLSCP04	0		3
CPLSMF04	0		2

Name	Hex Offset	Hex Value	Level
CPLSND04	50	04	4
CPLSNZ04	5D	80	4
CPLSOO04	32	01	4
CPLSPO04	33	08	4
CPLSTALL04	5D	40	4
CPLSTP04	0		1
CPLSUB04	3C		3
CPLSUD04	3C		4
CPLSUR04	0	40	4
CPLSUT04	40		4
CPLSYS04	31		3
CPLTJT04	32	10	4
CPLVAR04	60		2
CPLVERS04	17		3
CPLWLMAR	5D	20	4
CPLWRER04	34	80	4
CPLWSN04	60		3
CPL2EV04	32	02	4
CPL3SJ04	32	04	4

CPLREC05 - Current plan application index record

Name : DCLCPR05

Function:
This segment declares an application index record, physically located in EQQCPxDS and EQQNCPDS.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC05	AD ID INDEX RECORD
0	(0)	CHARACTER	19	CPLKEY05	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 05
2	(2)	CHARACTER	16	CPLAID05	APPLICATION
18	(12)	UNSIGNED	1	CPLSEQ05	CP05 SEQUENCE NUMBER 0 FOR SPECIAL CP05 REC
19	(13)	CHARACTER	*	CPLBDY05	BODY OF AD ID INDEX RECORD
19	(13)	CHARACTER	4	CPLEYE05	EYECATCHER
23	(17)	UNSIGNED	1	CPLVERS05	VERSION NUMBER
24	(18)	SIGNED	2	CPLOC05	NUMBER OF OCCURRENCES FOR INDEX RECORD (CPLSEQ05X=0) CONTAINS THE NUMBER OF CP05 RECORDS IN THE CHAIN
26	(1A)	CHARACTER	6	*	FREE
32	(20)	CHARACTER	*	CPLVAR05	VARIABLE PART OF RECORD

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	*	CPLOCS05	OCCURRENCES DATA
0	(0)	CHARACTER	32	CPLOCC05 (*)	OCCURRENCE DATA
0	(0)	CHARACTER	10	CPLIA05	INPUT ARRIVAL
0	(0)	CHARACTER	6	CPLIAD05	DATE
6	(6)	CHARACTER	4	CPLIAT05	TIME
10	(A)	CHARACTER	10	CPLADL05	ACTUAL COMPLETION DELETION
10	(A)	CHARACTER	6	CPLDLD05	DATE BLANK
16	(10)	CHARACTER	4	CPLDLT05	TIME 0000
20	(14)	UNSIGNED	3	CPLOCX05	OCCURRENCE NUMBER
23	(17)	SIGNED	2	CPLNET05	NETWORK ID
25	(19)	CHARACTER	1	CPLSTA05	OCCURRENCE STATUS U = UNDECIDED W = WAITING S = STARTED E = ENDED IN ERROR C = COMPLETED D = DELETED P = PENDING PREDECESSOR M = Mandatory pending
26	(1A)	1...		CPLCLT05	1 MEANS ALREADY MARKED..COMPLETE ON LTP
		.1..		CPLMCP05	1 MEANS ADDED BY MCP
		..1.		CPLIAM05	1 MEANS IA MODIFIED BY MCP
		...1		CPLCFR05	1 MEANS COPIED FOR REPORT
		.... 1...		CPLMPEND05	1 MEANS ORIG MPEND
		.... .111		*	FREE
27	(1B)	CHARACTER	1	CPLOCA05	D MEANS ADDED BY MCP- DIALOG E MEANS ADDED BY ETT P MEANS ADDED BY PIF A MEANS ADDED BY AR BLANK MEANS DP
28	(1C)	CHARACTER	1	CPLOCDB2	Y = ADDED TO DB2
29	(1D)	CHARACTER	1	CPLCP16AD	Y = ADDED TO CP16 D90A
30	(1E)	CHARACTER	2	*	FREE D90C

ENTRIES OF INDEX RECORD CPLSEQ050

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	*	CPLIDX05	INDEX RECORD ENTRY
0	(0)	CHARACTER	32	CPLOCCX05 (*)	SINGLE CP05 RECORDS IX
0	(0)	CHARACTER	10	CPLFIA05	FIRST INPUT ARRIVAL
0	(0)	CHARACTER	6	CPLFIAD05	DATE
6	(6)	CHARACTER	4	CPLFIAT05	TIME
10	(A)	CHARACTER	10	CPLLIA05	LAST INPUT ARRIVAL
10	(A)	CHARACTER	6	CPLLIAD05	DATE
16	(10)	CHARACTER	4	CPLLIAT05	TIME
20	(14)	SIGNED	2	CPLOCN05	OCCURRENCE NUMBER
22	(16)	CHARACTER	10	*	RESERVED

Cross reference

Name	Hex Offset	Hex Value	Level
cplbnd	1AC		4
cplcdeop	1A0		4
CPLCond_F	19C		4

Name	Hex Offset	Hex Value	Level
CPLCond_T	19A		4
CPLCond_U	19E		4
CPLCondTot	198		4
CPLCPROP	194		4
CPLCSUOP	196		4
CPLDEOP	E8		4
CPLPCOP	EA		4
CPLPROP	E6		4
CPLPSOP	D2		4
CPLPXOP	1A2		4
CPLR1OP	D4		4
CPLR2OP	D6		4
CPLSROP	EC		4
CPLSUOP	E4		4
CPLAADOP	AA		5
CPLAAOP	AA		4
CPLAATOP	B0		5
CPLADIOP	1A		4
CPLADUOP	CC		4
CPLAECOP	DD		4
CPLAEDOP	BE		5
CPLAEOP	BE		4
CPLAETOP	C4		5
CPLAJR	EE	02	5
cplAlcTime	38		2
CPLALTWS	FE		4
CPLASDOP	A0		5
CPLASIDJOB	128		4
CPLASOP	A0		4
CPLASSOP	EE	20	5
CPLASTOP	A6		5
CPLASUOP	EE	04	5
cplAutInf	15F		4
CPLBDYOP	C		3
CPLBNDDEL	8	80	3
CPLBNDFLAG	8		2
CPLBNDOP	0		1
CPLBND99K	0		2
cplclncm	65	02	5
cplclnDia	14A		4
cplclnRes	149		4

Name	Hex Offset	Hex Value	Level
cplClnStat	19		4
cplClnTyp	12A		4
CPLCOM3P	0		2
CPLCondId	1A8		4
cplcondrjob	16E	02	5
CPLCPCOOP	0	40	3
CPLCPMatc	1		2
CPLCPNDOP	0	80	3
CPLCPOCOP	4		3
CPLCPOP	4		2
CPLCPOP_cid	2		2
CPLCPOPOP	7		3
CPLCPREOP	0		1
CPLCPSWOP	0		2
cplCP15req	65	20	5
cplCritJobPred	16E	10	5
cplCritPath	178		4
cplCritPred	0	10	3
cplCrtPthX	180		4
CPLCSOCOP	4		3
CPLCSOP	4		2
CPLCSOP_cid	2		2
CPLCSOPOP	7		3
CPLCSSWOP	0		2
CPLCSTOP	D8		4
CPLCSUCOP	0		1
CPLCSUFLAG	1		2
CPLCSUSTEP	1	80	3
CPLDEFLT	65	80	5
CPLDEPTOP	F8		4
CPLDIRER	F9	10	5
CPLDIRES	F9	40	5
cplDOAtime	154		5
cpldprem	16E	08	5
cpldrecov	1AE	40	5
CPLDWTO	F9	04	5
CPLDWTOP	F9	01	5
CPLDWTOS	F9	02	5
cpldyn	1AE	80	5
cplEarE	17C		4
CPLEDUOP	C8		4

Name	Hex Offset	Hex Value	Level
cplEQQCLJBN	154		4
cplEQQCLjid	14B		4
cplerfail	153	08	5
CPLERKOP	E0		4
CPLEROOP	E2		4
CPLERROP	D9		4
cplEXdest	120		4
cplexec	65	04	5
cplexit13rc	1AE	08	5
cplExpJCL	114	01	5
cplexpjclUSED	153	40	5
cplExtInf	15D		4
CPLEXTOP	EE	40	5
CPLEYE3P	C		4
cplFakeSub	16F	02	5
cplflag05	114		4
cplflag06	153		4
cplflag07	15C		4
cplflag08	16E		4
cplflag09	16F		4
cplflag10	1AE		4
CPLFLGOP	EE		4
CPLFLGO2	EF		4
CPLFLGO3	F9		4
CPLFLG05	65		4
CPLFRMOP	66		4
cplFromPLC	15C	10	5
cplFTSAnop	16E	20	5
cplftwRC	160		4
CPLGROUP3P	11		4
CPLHRCOP	FA		4
CPLHRUOP	EF	04	5
CPLIADOP	2A		5
CPLIAOP	2A		4
CPLIATOP	30		5
CPLIDOP	5E		4
cplimmerr	114	04	5
CPLISDOP	B4		5
cplisdopf2	B4		7
cplisdopl4	B6		6
CPLISOP	B4		4

Name	Hex Offset	Hex Value	Level
CPLISTOP	BA		5
CPLJBNOP	4E		4
CPLJCLOP	64		4
cplJESdest	184		4
CPLJIDOP	56		4
CPLJIDOPN	5A		5
CPLJIDOPP	56		5
cplJLInfo	114	40	5
cplJLOdest	13C		4
cpljobcrt	116		5
cpljobpol	117		5
cplJobTblIdx	190		4
cplJRstat	115		4
CPLKEYOP	0		3
CPLKF03P	6		4
CPLKYP3P	0		4
CPLLATEP	EE	01	5
CPLLODOP	96		5
CPLLONGP	EF	01	5
CPLLOOP	96		4
CPLLOTOP	9C		5
cplluserop	134		4
CPLMCPUP	F0		4
cplmhld	65	10	5
cplmonp	153	80	5
cplMoreRun	114	80	5
cplmpendp	1AE	01	5
CPLMPREDopno	102		5
cplMvDeadl	1AE	10	5
CPLMXLVL	102		4
CPLNDXOP	2		4
CPLNETOP	34		4
cplnoerr	16F	08	5
cplNoerrJT	1AE	04	5
cplNoerrSkip	1AE	02	5
cplNoEval	16E	01	5
cplnop	65	08	5
CPLNUMOP	62		5
CPLOC03P	2		5
CPLODDOP	8C		5
CPLODOP	8C		4

Name	Hex Offset	Hex Value	Level
CPLODTOP	92		5
CPLOIDOP	82		5
CPLOIOP	82		4
CPLOITOP	88		5
cplOpEx03p	148		4
cplOpInfo	114	20	5
CPLOPIRET	D0		4
CPLOP03P	5		5
CPLOrigRC	1A4		4
CPLPCOOP	0	80	3
CPLPEDOP	78		5
CPLPENDP	EF	02	5
CPLPEOP	78		4
CPLPETOP	7E		5
CPLPIVLleft	8		3
CPLPIVLright	A		3
CPLPMandPend	0	08	3
CPLPMatc	1		2
CPLPMatcIVL	8		2
CPLPNDOP	0	20	3
CPLPNROP	0	40	3
CPLPOCOP	4		3
CPLPOP	4		2
CPLPOPOP	7		3
CPLPREOP	0		1
CPLPRIOP	DE		4
CPLPRPOP	EF	20	5
CPLPSDOP	6E		5
CPLPSOP	6E		4
CPLPSTOP	74		5
cplPSUdone	114	10	5
cplPSUtoken	144		4
CPLPSUWAIT	65	40	5
CPLPSWOP	0		2
cplPthOc	180		5
cplPthOp	183		5
CPLPTTOP	2		2
cplRecEnd	15C	40	5
cplrecjob	153	02	5
cplRecovOn	16F	10	5
cplrecprmp	153	04	5

Name	Hex Offset	Hex Value	Level
cplrecrun	153	01	5
cplRec24Is	15C	80	5
CPLREC3P	0		1
cplrejst	1AB		4
CPLRELOP	EF	10	5
cplRENG	0		1
cplRENG_comp	4B	80	3
cplRENG_flags	4B		2
cplRENG_IA	4C		2
cplRENG_IAD	4C		3
cplRENG_IAT	52		3
cplRENG_IATH	52		4
cplRENG_IATM	54		4
cplRENG_jobname	10		2
cplRENG_jsname	0		2
cplRENG_jsws	38		2
cplRENG_opno	48		2
cplRENG_zjob	10		3
cplRENG_zws	38		3
cplRengInf	1AA		4
CPLREROP	EE	80	5
CPLRERUT	F9	20	5
cplrescnew	2D	04	3
CPLRESTA	F9	80	5
cplRUSER	164		4
cplscript	15C	20	5
CPLSCROP	0	80	3
cplsending	153	20	5
cplshadow	16F	01	5
cplshoseq	1AE	20	5
CPLSOCOP	2		3
CPLSOP	2		2
CPLSOPOP	5		3
CPLSPDL	EE	08	5
CPLSPIA	EE	10	5
cplsrAVAco	40		2
CPLSRAVOP	2D	80	3
cplsrerr	2E		2
cplsridx	34		2
CPLSRIUSOP	2D	40	3
CPLSRIUXOP	2D	20	3

Name	Hex Offset	Hex Value	Level
cplsrkept	2D	10	3
cplsrkeptx	2D	08	3
CPLSRNOP	0		2
cplsrnum	30		2
CPLSRSOP	0		1
CPLSRTOP	2C		2
cplsrwrs	2F		2
CPLSSEQ	65	01	5
CPLSSEQOP	B4		6
CPLSSWOP	0		2
cplstch	12C		4
cplstchd	12C		5
cplstcht	130		5
cplstepdep	16F	80	5
cplStepSucc	16F	40	5
cplsuber	153	10	5
CPLSUBOP	EF	80	5
cplsubtoken	14B		5
CPLSUCOP	0		1
cplsyssfail	114	02	5
cplTaskType	1AF		4
CPLTJTOP	EF	40	5
CPLTMPOP	FC		4
cpltoken	11C		5
cpltoken_full	118		4
cpltoken_pre	118		5
cplTWSjbnm	15E		4
cpltwsjob	12B		4
CPLTXTOP	36		4
cplTZOFF	18E		4
cplTZONE	16E	04	5
CPLudata	104		4
cplUnexpRC	16F	04	5
cplURGmsg	15C	01	5
cplUserSYS	114	08	5
CPLVAR03	1B4		2
cplVdest	186		4
CPLVERS3P	10		4
cplWaitSE	15C	04	5
cplWaitWQA	15C	02	5
cplWASUJ	15C	08	5

Name	Hex Offset	Hex Value	Level
cplwlm	116		4
cplWLMclass	170		4
cplWLMmsg	16E	40	5
cplWLMprom	16E	80	5
CPLWRER	F9	08	5
CPLWSOP	5E		5
CPLXSTOP	DF		4
cplzcentric	16F	20	5
CPL2EVOP	EF	08	5
cpl3pExtInfo	0		1
cpl3pEXTname	0		2
cpl3pSEname	36		2
cpl3pSPLEX	16C		4

CPLREC06 - Current plan remaining slack record

Name : DCLCPR06

Function:
This segment declares the remaining slack record, physically located in EQQCPxDS and EQQNCPDS. The CPLREC06 record is structured as follows:

```
1 CPLREC06
  2 common data
  2 occurrence entries ( )
```

Common data segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	26	CPLREC06	REMAINING SLACK CHAIN REC
0	(0)	CHARACTER	19	CPLKEY06	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 06
2	(2)	SIGNED	2	CPLSEQ06	REM SLACK CHAIN REC NUMBER
4	(4)	CHARACTER	15	CPLKF06	FILLER, CONTAINS BLANKS
19	(13)	CHARACTER	7	CPLBDY06	BODY OF REM. SLACK RECORD
19	(13)	CHARACTER	4	CPLEYE06	EYECATCHER
23	(17)	UNSIGNED	1	CPLVERS06	VERSION NUMBER
24	(18)	SIGNED	2	CPL#OC06	NUMBER OF ENTRIES IN REC
26	(1A)	CHARACTER		CPLVAR06	VARIABLE PART OF RECORD

Occurrence entries segment

The value of CPL#OC06 in the common data segment describes the number of occurrence entries segments.

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLOCS06	OCCURRENCE ENTRIES
0	(0)	CHARACTER	16	CPLOCC06 (*)	OCCURRENCE ENTRY
0	(0)	UNSIGNED	3	CPLOCX06	OCCURRENCE RECORD INDEX
3	(3)	SIGNED	2	CPLNET06	NETWORK ID
5	(5)	CHARACTER	1	CPLSTA06	OCCURRENCE STATUS BLANK = NONCOMPLETED C = COMPLETED D = DELETED
			3 CHAR(1), FILLER		
6	(6)	CHARACTER	10	CPLLO06	EARLIEST LATEST STA IN OCC
6	(6)	CHARACTER	6	CPLLOD06	DATE
12	(C)	CHARACTER	4	CPLLOT06	TIME

Cross reference

Name	Hex Offset	Hex Value	Level
CPL#OC06	18		3
CPLBDY06	13		2
CPLEYE06	13		3
CPLKEY06	0		2
CPLKF06	4		3
CPLLOD06	6		4
CPLLOT06	C		4
CPLLO06	6		3
CPLNET06	3		3
CPLOCC06	0		2
CPLOCS06	0		1
CPLOCX06	0		3
CPLREC06	0		1
CPLSEQ06	2		3
CPLSTA06	5		3
CPLVAR06	1A		3
CPLVERS06	17		3

CPLREC07 - Current plan undecided operation record

Name : DCLCPR07

Function:
This segment declares an undecided operation record, physically located in EQQCPxDS and EQQNCPDS.
The CPLREC07 record is structured as follows:

```
   1 CPLREC07
     2 common data
     2 occurrence entries ( )
```

Common data segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC07	UNDECIDED OCC. RECORD
0	(0)	CHARACTER	19	CPLKEY07	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 07
2	(2)	SIGNED	2	CPLSEQ07	UNDECIDED OCC: REC NUMBER
4	(4)	CHARACTER	15	CPLKF07	FILLER CONTAINS BLANKS
19	(13)	CHARACTER	*	CPLBDY07	BODY OF UNDECIDED REC
19	(13)	CHARACTER	4	CPLEYE07	EYECATCHER
23	(17)	UNSIGNED	1	CPLVERS07	VERSION NUMBER
24	(18)	SIGNED	2	CPL#OC07	NUMBER OF ENTRIES IN REC
26	(1A)	CHARACTER	*	CPLVAR07	VARIABLE PART OF RECORD

Occurrence entries segment

The value of CPL#OC07 in the common data segment describes the number of occurrence entries segments.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLOCS07	OCCURRENCE ENTRIES
0	(0)	CHARACTER	56	CPLOCC07 (*)	OCCURRENCE ENTRY
0	(0)	CHARACTER	16	CPLAID07	APPLICATION ID
16	(10)	CHARACTER	10	CPLIA07	INPUT ARRIVAL (FINAL)
16	(10)	CHARACTER	6	CPLIAD07	DATE
22	(16)	CHARACTER	4	CPLIAT07	TIME
26	(1A)	CHARACTER	24	CPLJXT07	DESCRIPTIVE TEXT
50	(32)	UNSIGNED	3	CPLOCX07	OCCURRENCE RECORD INDEX
53	(35)	SIGNED	2	CPLNET07	NETWORK ID
55	(37)	CHARACTER	1	CPLSTA07	OCCURRENCE STATUS U = UNDECIDED D = DELETED W = WAITING

Cross reference

Name	Hex Offset	Hex Value	Level
CPL#OC07	18		3
CPLAID07	0		3
CPLBDY07	13		2

Name	Hex Offset	Hex Value	Level
CPLEYE07	13		3
CPLIAD07	10		4
CPLIAT07	16		4
CPLIA07	10		3
CPLJXT07	1A		3
CPLKEY07	0		2
CPLKF07	4		3
CPLNET07	35		3
CPLOCC07	0		2
CPLOCS07	0		1
CPLOCX07	32		3
CPLREC07	0		1
CPLSEQ07	2		3
CPLSTA07	37		3
CPLVAR07	1A		3
CPLVERS07	17		3

CPLREC08 - Current plan ended-in-error record

Name : DCLCPR08

Function:
This segment declares the ended-in-error record, physically located in EQQCPxDS and EQQNCPDS. The CPLREC08 record is structured as follows:

```
1 CPLREC08
  2 common data
  2 ended-in-error entries ( )
```

Common data segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC08	ENDED IN ERROR RECORD
0	(0)	CHARACTER	19	CPLKEY08	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 08
2	(2)	SIGNED	2	CPLSEQ08	ERROR RECORD NUMBER
4	(4)	BITSTRING	2	CPLIND08	
				CPLARR08	AUTO RECOVERY REQUEST
				CPLARM08	AUTO RECOVERY REQUEST
				CPLNPP08	ON: AUTO RECOVERY NOT POSTPONED
19	(13)	CHARACTER	*	CPLBDY08	BODY OF ERROR RECORD
19	(13)	CHARACTER	4	CPLEYE08	EYECATCHER
23	(17)	UNSIGNED	1	CPLVERS08	VERSION NUMBER
24	(18)	SIGNED	2	CPL#OP08	NUMBER OF ENTRIES IN REC
26	(1A)	CHARACTER	*	CPLVAR08	VARIABLE PART OF RECORD

Ended-in-error entry segment

The value of CPL#OC08 in the common data segment describes the number of ended-in-error entries segments.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	*	CPLOPS08	INDEXES TO OD RCDS IN ERROR
0	(0)	CHARACTER	6	CPLENT08 (*)	ENDED-IN-ERROR ENTRY
0	(0)	CHARACTER	4	CPLOPX08	INDEX TO OP RECORD
0	(0)	UNSIGNED	3	CPLOCX08	OCC NUMBER
3	(3)	UNSIGNED	1	CPLOP#08	OPERATION NUMBER A ZERO ENTRY MEANS OP NOT ENDED IN ERROR ANYMORE
4	(4)	BITSTRING	2	CPLIND08	
		1...		CPLARR08	AUTO RECOVERY REQUEST
		.1..		CPLARM08	AUTO RECOVERY REQUEST
4	(4)	BITSTRING	1	*	FOR FUTURE USE

Cross reference

Name	Hex Offset	Hex Value	Level
CPL#OP08	18		3
CPLARM08	4	40	4
CPLARR08	4	80	4
CPLBDY08	13		2
CPLENT08	0		2
CPLEYE08	13		3
CPLIND08	4		3
CPLKEY08	0		2
CPLKF08	4		3
CPLNPP08	4	20	4
CPLOCX08	0		4
CPLOP#08	3		4
CPLOPS08	0		1
CPLOPX08	0		3
CPLREC08	0		1
CPLSEQ08	2		3
CPLVAR08	1A		3
CPLVERS08	17		3

CPLREC09 - Current plan rerun record

Name : DCLCPR09

Function:
This segment declares the rerun operation record, physically located in EQQCPxDS and EQQNCPDS.

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	328	CPLREC09	RERUN OPERATION RECORD
0	(0)	CHARACTER	12	CPLKEY09	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 09
2	(2)	UNSIGNED	3	CPLOCC09	OCC NUMBER
5	(5)	UNSIGNED	1	CPLOP09	OPERATION NUMBER
6	(6)	SIGNED	2	CPLRR#09	RERUN NUMBER
8	(8)	CHARACTER	4	CPLKF09	ALWAYS BLANKS
12	(C)	CHARACTER	174	CPLBDY09	CP3P BOBY

Cross reference

Name	Hex Offset	Hex Value	Level
CPLBDY09	C		2
CPLKEY09	0		2
CPLKF09	8		3
CPLOCC09	2		3
CPLOP09	5		3
CPLREC09	0		1
CPLRR#09	6		3

CPLREC10 - Current plan workstation activity record

Name : DCLCPR10

Function:
This segment declares the workstation activity record, physically located in EQQCPxDS and EQQNCPDS.

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	216	CPLREC10	WS ACTIVITY RECORD
0	(0)	CHARACTER	19	CPLKEY10	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 10
2	(2)	CHARACTER	4	CPLWSN10	WSID
6	(6)	CHARACTER	6	CPLDAY10	DATE
12	(C)	CHARACTER	7	*	SHOULD ALWAYS BE BLANKS
19	(13)	CHARACTER	197	CPLBDY10	RECORD BODY
19	(13)	CHARACTER	4	CPLEYE10	EYECATCHER
23	(17)	UNSIGNED	1	CPLVERS10	VERSION NUMBER
24	(18)	CHARACTER	2	CPL#OP10 (96)	NUMBER OF PARALLELL OPS PER 15 MINUTES

Cross reference

Name	Hex Offset	Hex Value	Level
CPL#OP10	18		3
CPLBDY10	13		2
CPLDAY10	6		3
CPLEYE10	13		3
CPLKEY10	0		2
CPLREC10	0		1
CPLVERS10	17		3
CPLWSN10	2		3

CPLREC11 - Current plan ready list record

Name : DCLCPR11

Function:
This segment declares a ready list record, physically located in EQQCPxDS and EQQNCPDS.
The CPLREC11 record is structured as follows:
1 CPLREC11
2 common data
2 ready operation entries ( )

Common data segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC11	READY LIST RECORD
0	(0)	CHARACTER	19	CPLKEY11	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 11
2	(2)	CHARACTER	2	CPLLST11	'01' FOR READY LIST '02' FOR WAITING LIST
4	(4)	CHARACTER	4	CPLWSN11	WS NAME
8	(8)	SIGNED	2	CPLSEQ11	SEQUENCE NUMBER
10	(A)	CHARACTER	9	CPLKF11	FILLER CONTAINS BLANKS
19	(13)	CHARACTER	*	CPLBDY11	READY LIST RECORD BODY
19	(13)	CHARACTER	4	CPLEYE11	EYECATCHER
23	(17)	UNSIGNED	1	CPLVERS11	VERSION NUMBER
24	(18)	SIGNED	2	CPL#DO11	NO OF ENTRIES
26	(1A)	CHARACTER	*	CPLVAR11	VARIABLE PART OF RECORD

Ready operation entries segment

The value of CPL#DO11 in the common data segment describes the number of ready operation entries segments.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLDOS11	DOA ENTRIES
0	(0)	CHARACTER	4	CPLDOA11 (*)	DOA ENTRY

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	UNSIGNED	3	CPLOCX11	OCC NUMBER
3	(3)	UNSIGNED	1	CPLOPX11	OPERATION NUMBER

Cross reference

Name	Hex Offset	Hex Value	Level
CPL#DO11	18		3
CPLBDY11	13		2
CPLDOA11	0		2
CPLDOS11	0		1
CPLEYE11	13		3
CPLKEY11	0		2
CPLKF11	A		3
CPLLST11	2		3
CPLOCX11	0		3
CPLOPX11	3		3
CPLREC11	0		1
CPLSEQ11	8		3
CPLVAR11	1A		3
CPLVERS11	17		3
CPLWSN11	4		3

CPLREC12 - Current plan potential predecessor record

Name : DCLCPR12

Function:
This segment declares a potential predecessor record, physically located in EQQCPxDS and EQQNCPDS. Each record defines application names and operation numbers for operations that could become external successors to new occurrences added to the plan by MCP functions.
The CPLREC12 record is structured as follows:

```
    1 CPLREC12
      2 common data
      2 potential predecessors ( )
```

Common data segment

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	32	CPLREC12	POTENTIAL PREDECESSOR RECORD
0	(0)	CHARACTER	19	CPLKEY12	
0	(0)	CHARACTER	2	*	RECORD TYPE IS ALWAYS 12
2	(2)	CHARACTER	16	CPLAPRE12	APPLICATION NAME
18	(12)	CHARACTER	1	*	BLANK
19	(13)	CHARACTER	13	CPLBDY12	BODY OF PREDECESSOR RECORD
19	(13)	CHARACTER	4	CPLEYE12	EYECATCHER

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
23	(17)	UNSIGNED	1	CPLVERS12	VERSION NUMBER
24	(18)	SIGNED	2	CPLNUM12	NUMBER OF SUCC ENTRIES
26	(1A)	BITSTRING	1	CPLFLAGS12	CP12 FLAGS
		1...		CPLCOND12	
		.1..		*	FREE
27	(1B)	CHARACTER	5	*	FREE
32	(20)	CHARACTER		CPLVAR12	START OF SUCCESSOR ENTRIES

Potential predecessor segment

The value of CPLNUM12 in the common data segment describes the number of potential predecessor segments.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	24	CPLENT12	POTENTIAL SUCCESSOR ENTRY
0	(0)	SIGNED	2	CPLPRE12	PREDECESSOR OPERATION NUMBER
2	(2)	SIGNED	2	CPLSUC12	SUCCESSOR OPERATION NUMBER
4	(4)	CHARACTER	16	CPLASUC12	SUCCESSOR APPLICATION NAME
20	(14)	BITSTRING	1	CPLFLG12	WORK FLAGS
		1...		CPLMAT12	MATCH FOUND IN MCP ADD
		.1..		*	FREE
21	(15)	CHARACTER	3	*	FREE

Cross reference

Name	Hex Offset	Hex Value	Level
CPLAPRE12	2		3
CPLASUC12	4		2
CPLBDY12	13		2
CPLCOND12	1A	80	4
CPLENT12	0		1
CPLEYE12	13		3
CPLFLAGS12	1A		3
CPLFLG12	14		2
CPLKEY12	0		2
CPLMAT12	14	80	3
CPLNUM12	18		3
CPLPRE12	0		2
CPLREC12	0		1
CPLSUC12	2		2
CPLVAR12	20		3
CPLVERS12	17		3

CPLREC14 - Current plan job log staging and status record

Name : DCLCPR14

Function:
This segment declares a job log staging and status record, physically located in eqqcpxds and eqqncpds.
The CPLREC14 record is structured as follows:
1 CPLREC14
2 common data
2 message line entries ( )

Common data segment

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	111	CPLREC14	SYSOUT FORM JOBLOG
0	(0)	CHARACTER	8	CPL14KEY	KEY OF RECORD
0	(0)	CHARACTER	2	CPLTYP14	RECORD TYPE (14)
2	(2)	UNSIGNED	3	CPLOCC14	OCC NUMBER IN CP
5	(5)	UNSIGNED	1	CPLOPR14	OPERATION NUMBER
6	(6)	SIGNED	2	CPLSEQ14	SEQUENCE NUMBER, 1ST = 0001
8	(8)	CHARACTER	81	CPL14BDY	
8	(8)	CHARACTER	12	*	
20	(14)	CHARACTER	4	CPL14EYE	EYE CATCHER
24	(18)	CHARACTER	4	CPL14WSN	WORK STATION NAME
28	(1C)	CHARACTER	8	CPL14JNM	JOB NAME
36	(24)	CHARACTER	8	CPL14JID	JOB ID
44	(2C)	ADDRESS	4	CPLTOTSZ	TOTAL SIZE OF RECORD , BEFORE SPLIT INTO TWO OR MORE, AND BEFORE COMPRESSION
48	(30)	SIGNED	4	CPLSIZ14	SIZE OF RECORD WHEN IN CP
48	(30)	SIGNED	4	CPL14CSZ	COMPRESSED SIZE (INCL HEADER)
52	(34)	SIGNED	4	CPLLO14	OFFSET TO LAST MESSAGE LINE
56	(38)	SIGNED	4	CPL#LINE	NUMBER OF JOBLOG RECORDS
60	(3C)	CHARACTER	16	CPL14ADID	APPLICATION ID
76	(4C)	CHARACTER	10	CPL14IA	INPUT ARRIVAL YYMMDDHHMM
76	(4C)	CHARACTER	6	CPL14IAD	IA DATE
82	(52)	CHARACTER	4	CPL14IAT	IA TIME
86	(56)	CHARACTER	1	CPL14OCLASS	ORIG MSG CLASS
87	(57)	CHARACTER	1	CPL14SYST	A=JES2, B=JES3 4=AS400 U=UNIX X=AIX O=OS/2
88	(58)	CHARACTER	1	CPL14FLAGS	FLAGS
		1...		CPL14NCMP	1= JOBLOG NOT COMPLETE
		.1..		CPL14HEAD	1= FIXED PART OF RECORD HAS BEEN PASSED TO CONTR.
		..1.		CPL14COMP	1= JOBLOG IS IN COMPRESSED FORMAT (CSRCESRV COMPR.)
		...1		CPL14NOLOG	1= NO JOBLOG FOUND
		.... 1...		CPL14R3	1= R3 OR LATER
		.... .1..		CPL14DELRQ	1= DELAYED RETRIEVAL REQ
		.... ..1.		CPL14ARC	ARCH JOB LOG RETRIEVAL
		1		CPL14NOSTR	1=DO NOT ASK JLOGSTRU 0=ASK JLOGSTRU
89	(59)	UNSIGNED	1	CPL14FSTEP	ENDED-IN-ERROR STEP NO
90	(5A)	CHARACTER	1	CPL14VERS	CB VERSION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
91	(5B)	CHARACTER	5	CPL14STATUS	STATUS OF JOBLOG
91	(5B)	CHARACTER	1	CPL14ST	ARCHIVING STATUS A = ARCHIVED IN CONTROLLER R = WAITING FOR RETRIEVAL
92	(5C)	CHARACTER	4	CPL14MVSID	SMF TRACKER MVS™ ID
96	(60)	SIGNED	4	CPL14SLO	OFFSET TO STEP LIST IN REC
100	(64)	CHARACTER	3	CPL14JR	LAST 3 NUMBERS OF JES REL
103	(67)	CHARACTER	4	CPL14JRD	JES READER DATE IN BINARY (COPY OF 3P EXRDATE)
107	(6B)	CHARACTER	4	CPL14JRT	JES READER TIME IN BINARY (COPY OF 3P EXRDATE)
111	(6F)	CHARACTER		CPLDATA14	DATA PORTION

Message line entry segment

The value of CPL#LINE in the common data segment describes the number of message line entries segments.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLSYSOUT14	
0	(0)	SIGNED	4	CPLMSG14L	LENGTH OF SYSOUT MESSAGE LINE
4	(4)	CHARACTER	2	CPLMSGFLG	FLAGS
4	(4)	CHARACTER	1	CPLLTYPE14	LINTYPE J = JOBLOG <BLANK> = USERSYSOUT
5	(5)	CHARACTER	1	*	RESERVED
6	(6)	SIGNED	2	CPLMSG14NO	OFFSET TO NEXT MESSAGE
8	(8)	CHARACTER	*	CPLMSG14	MESSAGE

Cross reference

Name	Hex Offset	Hex Value	Level
CPL#LINE	38		3
CPLDATA14	6F		2
CPLLO14	34		3
CPLLTYPE14	4		3
CPLMSGFLG	4		2
CPLMSG14	8		2
CPLMSG14L	0		2
CPLMSG14NO	6		2
CPLOCC14	2		3
CPLOPR14	5		3
CPLREC14	0		1
CPLSEQ14	6		3
CPLSIZ14	30		3

Name	Hex Offset	Hex Value	Level
CPLSYSOUT14	0		1
CPLTOTSZ	2C		3
CPLTYP14	0		3
CPL14ADID	3C		3
CPL14ARC	58	02	4
CPL14BDY	8		2
CPL14COMP	58	20	4
CPL14CSZ	30		4
CPL14DELRQ	58	04	4
CPL14EYE	14		3
CPL14FLAGS	58		3
CPL14FSTEP	59		2
CPL14HEAD	58	40	4
CPL14IA	4C		3
CPL14IAD	4C		4
CPL14IAT	52		4
CPL14JID	24		3
CPL14JNM	1C		3
CPL14JR	64		2
CPL14JRD	67		2
CPL14JRT	6B		2
CPL14KEY	0		2
CPL14MVSID	5C		3
CPL14NCMP	58	80	4
CPL14NOLOG	58	10	4
CPL14NOSTR	58	01	4
CPL14OCLASS	56		3
CPL14R3	58	08	4
CPL14SLO	60		2
CPL14ST	5B		3
CPL14STATUS	5B		2
CPL14SYST	57		3
CPL14VERS	5A		2
CPL14WSN	18		3

CPLREC15 - Stand-alone clean up record

Name : DCLCPR15

Function:
This segment declares the stand-alone clean up record.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	70	CPLREC15	STAND-ALONE CLEAN UP
0	(0)	CHARACTER	18	CPL15KEY	KEY OF RECORD
0	(0)	CHARACTER	2	CPL15TYP	RECORD TYPE (15)
2	(2)	CHARACTER	8	CPL15JNM	JOB NAME
10	(A)	CHARACTER	8	CPL15JID	JOB ID
20	(14)	CHARACTER	50	CPL15BDY	BODY OF RECORD
20	(14)	CHARACTER	6	*	RESERVED
26	(1A)	CHARACTER	4	CPL15EYE	EYE CATCHER=CP15
30	(1E)	CHARACTER	1	CPL15VER	VERSION
31	(1F)	CHARACTER	1	CPL15STAT	CPL15 STATUS
32	(20)	SIGNED	2	CPL15RETRY	ASK OPINFO RETRY COUNTER
34	(22)	CHARACTER	4	CPL15JRD	JES READER DATE IN BINARY
38	(26)	CHARACTER	4	CPL15JRT	JES READER TIME IN BINARY
42	(2A)	CHARACTER	8	CPL15TOK	OCC TOKEN
50	(32)	UNSIGNED	1	CPL15OPR	OPERATION NUMBER
51	(33)	CHARACTER	3	*	RESERVED
54	(36)	CHARACTER	8	CPL15DEST	CLEAN UP JOBLOG DESTINATION SET BY K3P
62	(3E)	CHARACTER	4	CPL15CODE	CLEAN UP JOB ERROR CODE
66	(42)	CHARACTER	24	*	FREE

Cross reference

Name	Hex Offset	Hex Value	Level
CPLREC15	0		1
CPL15BDY	14		2
CPL15CODE	3E		3
CPL15DEST	36		3
CPL15EYE	1A		3
CPL15JID	A		3
CPL15JNM	2		3
CPL15JRD	22		3
CPL15JRT	26		3
CPL15KEY	0		2
CPL15OPR	32		3
CPL15RETRY	20		3
CPL15STAT	1F		3
CPL15TOK	2A		3
CPL15TYP	0		3
CPL15VER	1E		3

CPLREC16 - Restart and clean up operinfo for deletion

Name : DCLCPR16

Function:
This segment declares the restart and clean up operation information to be deleted (created by DP batch processing).
The CPLREC16 record is structured as follows:

```
  1 CPLREC16
    2 common data
    2 occurrence entries ( )
```

Common data segment

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	*	CPLREC16	
0	(0)	CHARACTER	19	CPL16KEY	KEY OF RECORD
0	(0)	CHARACTER	2	CPL16TYP	RECORD TYPE (16)
2	(2)	SIGNED	2	CPL16SEQ	SEQUENCE NUMBER
4	(4)	CHARACTER	15	*	RESERVED
19	(13)	CHARACTER	*	CPL15BDY	BODY OF RECORD
19	(13)	CHARACTER	4	CPL16EYE	EYE CATCHER=CP16
23	(17)	CHARACTER	1	CPL16VER	VERSION
24	(18)	CHARACTER	1	*	FREE
26	(1A)	SIGNED	2	CPL16TOT	NUMBER OF ENTRIES IN THIS RECORD
28	(1C)	CHARACTER	8	CPL16TOD	CP16 ID (TOD)
36	(24)	CHARACTER	8	*	FREE
44	(2C)	CHARACTER	*	CPL16VAR	

Occurrence entry segment

The value of CPL16TOT in the common data segment describes the number of occurrence entries segments.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	*	CPL16TAB	
0	(0)	CHARACTER	8	CPL16ENT(*)	
0	(0)	CHARACTER	8	CPL16OCC	OCCURRENCE TOKEN

Cross reference

Name	Hex Offset	Hex Value	Level
CPLREC16	0		1
CPL16BDY	13		2
CPL16ENT	0		2
CPL16EYE	13		3
CPL16KEY	0		2

Name	Hex Offset	Hex Value	Level
CPL16OCC	0		3
CPL16SEQ	2		3
CPL16TAB	0		1
CPL16TOD	1C		3
CPL16TOT	1A		3
CPL16TYP	0		3
CPL16VAR	2C		3
CPL16VER	17		3

CPLREC20 - Workstation activity record for virtual workstation

Name : DCLCPR20

Function:
For each actual destination associated to a virtual workstation, defines the same kind of information as DCLCPR10.

```
Offsets       Type     Length  Name          Description
    0    (0) STRUCTURE   798  CPLREC20       WS ACTIVITY RECORD
    0    (0) CHARACTER    19  CPLKEY20
    0    (0) CHARACTER     2  CPLK20         '20'
    2    (2) CHARACTER     4  CPLWSN20       WS NAME
    6    (6) CHARACTER     1  CPLWSTYP       WS TYPE:'0' ==> NORMAL (FUTURE USE)
                                                     '1' ==> VIRTUAL
                                                     '2' ==> VIRTUAL DEST
    7    (7) CHARACTER     8  CPLDEST20      WS DESTINATION
   15    (F) CHARACTER     4  CPLDATE20      DATE
   15    (F) CHARACTER     2  CPLYY20        YEAR: 'YY'
   17   (11) UNSIGNED      2  CPLJUL20       JULIAN DAY: NNN
   19   (13) CHARACTER   779  CPLBDY20       RECORD BODY
   19   (13) CHARACTER     4  CPLEYE20       EYE CATCHER
   23   (17) UNSIGNED      1  CPLVERS20      VERSION NUMBER
   24   (18) CHARACTER     6  CPLDAY20       DATE: YYMMDD
   30   (1E) CHARACTER     8  CPL#OP20(96)   NUMBER OF PARALLEL
                                             OPERATIONS PER 15 MINUTES
```

Cross reference

```
HEX          HEX
NAME                                         OFFSET      VALUE     LEVEL
____                                         ______    ________    _____
CPL#OP20                                         1E                    3
CPLBDY20                                         13                    2
CPLDATE20                                         F                    3
CPLDAY20                                         18                    3
CPLDEST20                                         7                    3
CPLEYE20                                         13                    3
CPLJUL20                                         11                    4
CPLKEY20                                          0                    2
CPLK20                                            0                    3
CPLREC20                                          0                    1
CPLVERS20                                        17                    3
CPLWSN20                                          2                    3
CPLWSTYP                                          6                    3
CPLYY20                                           F                    4
```

CPLREC22 - Current plan virtual workstation record

Name : DCLCPR22

Function:
This segment declares a current plan workstation record, physically located in EQQCPxDS
and EQQNCPDS.
The CPLREC22 record is structured as follows:
1 CPLREC22
2 common data
2 open time interval ( )
2 access method ( )
For each actual destination associated to a virtual workstation, defines the same kind of information
as DCLCPR02.

Common data segment

Offsets		Type	Length	Name	Description
0	(0)	STRUCTURE	*	CPLREC22	CURRENT PLAN WS RECORD
0	(0)	CHARACTER	236	CPLCOM22	
0	(0)	CHARACTER	19	CPLKEY22	
0	(0)	CHARACTER	2	CPLKID22	TYPE IS ALWAYS 22
2	(2)	CHARACTER	4	CPLWSN22	WORKSTATION NAME
6	(6)	CHARACTER	8	CPLDST22	DESTINATION NAME
14	(E)	CHARACTER	5	CPLKF22	SHOULD CONTAIN BLANKS
19	(13)	CHARACTER	217	CPLBDY22	WS RECORD BODY
19	(13)	CHARACTER	4	CPLEYE22	EYECATCHER
23	(17)	UNSIGNED	1	CPLVERS22	VERSION NUMBER
24	(18)	CHARACTER	32	CPLDESCR22	WS DESCRIPTION
56	(38)	CHARACTER	12	CPLSUMC	SUM OF COMPLETED OPERATIONS
56	(38)	SIGNED	4	CPLSUMC#	NOT USED
60	(3C)	SIGNED	4	CPLSUMCE	NOT USED
64	(40)	SIGNED	4	CPLSUMCR	NOT USED
68	(44)	CHARACTER	12	CPLSUMI	SUM OF INTERRUPTED OPERATIONS
68	(44)	SIGNED	4	CPLSUMI#	NOT USED
72	(48)	SIGNED	4	CPLSUMIE	NOT USED
76	(4C)	SIGNED	4	CPLSUMIR	NOT USED
80	(50)	CHARACTER	8	CPLSUMS	SUM OF STARTED OPERATIONS
80	(50)	SIGNED	4	CPLSUMS#	NUMBER
84	(54)	SIGNED	4	CPLSUMSE	ESTIMATED DURATION
88	(58)	CHARACTER	8	CPLSUMR	SUM OF 'READY' OPERATIONS (R,A,*)
88	(58)	SIGNED	4	CPLSUMR#	NOT USED
92	(5C)	SIGNED	4	CPLSUMRE	NOT USED
96	(60)	CHARACTER	8	CPLSUMW	SUM OF WAITING OPERATIONS
96	(60)	SIGNED	4	CPLSUMW#	NOT USED
100	(64)	SIGNED	4	CPLSUMWE	NOT USED
104	(68)	SIGNED	4	CPLSUMA	NOT USED
108	(6C)	SIGNED	4	CPLSUMNRR	NOT USED
112	(70)	SIGNED	4	CPLSUMU	NOT USED
116	(74)	SIGNED	4	CPLSUME	NOT USED
120	(78)	SIGNED	4	CPLSUML	NOT USED
124	(7C)	CHARACTER	8	CPLCREVT	
124	(7C)	SIGNED	4	CPLDATE	DATE FORMAT (00YYDDDF)
128	(80)	SIGNED	4	CPLTIME	TIME FORMAT (SECS*100)
132	(84)	SIGNED	4	CPLSUMX	SUM OF DUMMY COMPL OPERATIONS
136	(88)	SIGNED	4	*(2)	
144	(90)	CHARACTER	1	CPLWSTYP	COPIED FROM OWNING CP02
145	(91)	CHARACTER	1	CPLWSREP	COPIED FROM OWNING CP02
146	(92)	CHARACTER	2	CPLWSR1N	RESOURCE NAME
148	(94)	UNSIGNED	2	CPLWSR1IU#	NUMBER IN USE (PARALLEL SERVERS IN USE = NO. STARTED OPERATIONS
150	(96)	CHARACTER	2	CPLWSR1FLG	RESOURCE INDICATORS
		1...		CPLWSR1C	RESOURCE USED AT CONTROL
		.1..		*	FREE

```
            ..1.  ....       CPLWSWAIT      COPIED FROM OWNING CP02
            ...1  ....       CPLWSVIRT      COPIED FROM OWNING CP02
            ....  1...       CPLWSZCAGENT   COPIED FROM OWNING CP02
150   (96) BITSTRING     1   *              FREE
            ....  .1..       CPLWSZNOINFO   ZCENTRIC NO INFO AVAIL
            ....  ..1.       CPLWSDYN       DYNAMIC WORKSTATION
150   (96) BITSTRING     1   *              FREE
152   (98) CHARACTER     2   CPLWSR2N       RESOURCE NAME
154   (9A) UNSIGNED      2   CPLWSR2IU#     NUMBER IN USE
156   (9C) CHARACTER     2   CPLWSR2FLG     RESOURCE INDICATORS
            1...  ....       CPLWSR2C       RESOURCE USED AT CONTROL
            .1..  ....       CPLWSSTAT      WSSTAT GENERATED STATUS
            ..1.  ....       CPLUX009       UX009 GENERATED STATUS
156   (9C) BITSTRING     1   *              RESERVED
158   (9E) CHARACTER     1   CPLRLTYP       READY LIST TYPE
159   (9F) CHARACTER     1   CPLFLG02       FLAG BITS
            1...  ....       CPLPREP        COPIED FROM OWNING CP02
            .1..  ....       CPLIVLNU       IVL NOT USED AT ALL
            ..1.  ....       CPLIVLNP       NO PARALLEL SERVERS
            ...1  ....       CPLSTC         COPIED FROM OWNING CP02
            ....  1...       CPLWTO         COPIED FROM OWNING CP02
            ....  .1..       CPLOFF_PEND    WS IS PENDING OFFLINE
            ....  ..1.       CPLRTR_PEND    T EVENT PENDING
            ....  ...1       CPLAWS_VARY    VARIED AWS SET
160   (A0) CHARACTER     8   CPLPREEV
160   (A0) CHARACTER     4   CPLPREDT       DATE IN TIME MACRO FORM
164   (A4) CHARACTER     4   CPLPRETM       TIME IN TIME MACRO FORM
168   (A8) CHARACTER     8   CPLSUDS        NOT USED
176   (B0) SIGNED        2   CPLIVL22#I     NUMBER OF OPEN INTERVLS
178   (B2) SIGNED        2   CPLMAX15       NOT USED
180   (B4) CHARACTER     1   CPLWSSTA       USED BY DIALOG
181   (B5) CHARACTER     1   CPLCWS_STAT    WORK STATION STATUS
182   (B6) SIGNED        2   CPLTRT02       NOT USED
184   (B8) CHARACTER     8   CPLCOFF02      OFFLINE TIME STAMP
184   (B8) CHARACTER     4   CPLCOFF_DATE   OFFLINE DATE
188   (BC) CHARACTER     4   CPLCOFF_TIME   OFFLINE TIME
192   (C0) CHARACTER     4   CPLCAWS02      NOT USED
196   (C4) UNSIGNED      2   CPLSSEQ#02     CURRENT SUBMIT SEQUENCE
198   (C6) UNSIGNED      1   CPLRSEQ#02     CURRENT REQUEST SEQUENC
199   (C7) UNSIGNED      1   CPLSUMCES      NOT USED
200   (C8) UNSIGNED      1   CPLSUMCRS      NOT USED
201   (C9) UNSIGNED      1   CPLSUMIES      NOT USED
202   (CA) UNSIGNED      1   CPLSUMIRS      NOT USED
203   (CB) UNSIGNED      1   CPLSUMSES
204   (CC) UNSIGNED      1   CPLSUMRES      NOT USED
205   (CD) UNSIGNED      1   CPLSUMWES      NOT USED
206   (CE) CHARACTER     1   CPLWSTWS       NOT USED
207   (CF) CHARACTER     1   CPLLNK02       NOT USED
208   (D0) CHARACTER     1   CPLFLGTWS      NOT USED
            1...  ....       CPLLNKVA       NOT USED
            .1..  ....       CPLSTAVA       NOT USED
            ..1.  ....       CPLLNKDW       NOT USED
            ...1  ....       CPLWRTST       NOT USED
            ....  1...       CPLCMDLNK      NOT USED
            ....  .1..       CPLCMDSTA      NOT USED
            ....  ..1.       CPLFULLNK      NOT USED
209   (D1) CHARACTER     1   CPLAUTO        ALWAYS N
210   (D2) SIGNED        2   CPLACCM22#     ALWAYS 0 NOT USED
212   (D4) SIGNED        2   CPLLIMIT       CPU LIMIT VALUE
214   (D6) CHARACTER     1   *              FREE
215   (D7) CHARACTER     1   CPLRENGT       REMOTE ENGINE TYPE
216   (D8) SIGNED        4   CPLVMAX15      MAC NO EVENTS IN 15 MIN
220   (DC) SIGNED        2   CPLEOPT22#     NOT USED
222   (DE) CHARACTER    14   *              FREE
236   (EC) CHARACTER     *   CPLWSVAR22     VARIABLE LENGTH SECTION
```

Open time interval segment

The value of CPLIVL22#I in the common data segment describes the number of open time interval segments that follow the common data.

```
Offsets      Type     Length  Name           Description
  236   (EC) STRUCTURE    48  CPLIVLOI22(*)
  236   (EC) CHARACTER    10  CPLIVLFR       INTERVAL START
  236   (EC) CHARACTER     6  CPLIVLFD       DATE YYMMDD
  242   (F2) CHARACTER     4  CPLIVLFT       TIME HHMM
  246   (F6) CHARACTER    10  CPLIVLTO       INTERVAL END
  246   (F6) CHARACTER     6  CPLIVLTD       DATE YYMMDD
  252   (FC) CHARACTER     4  CPLIVLTT       TIME HHMM
  256  (100) UNSIGNED      2  CPLIVL#PS      MAX PARALLEL SERVERS
  258  (102) UNSIGNED      2  CPLIVL#DPPS    SAME, SET AT DAILY PLANNING
  260  (104) BITSTRING     1  CPLIVLF2       FLAGS
             1...  ....       CPLIVLMC       IVL SET BY MCP
             .1..  ....       CPLIVLDP       IVL SET BY DP (WSD)
             ..11  1111       *              FREE
  261  (105) CHARACTER     1  *              FREE
  262  (106) UNSIGNED      2  CPLIVL#R1      CURRENT RESOURCE CAPACITY
  264  (108) UNSIGNED      2  CPLIVL#DPR1    CAPACITY SET AT DAILY PLAN
  266  (10A) UNSIGNED      2  CPLIVL#R2      CURRENT RESOURCE CAPACITY
  268  (10C) UNSIGNED      2  CPLIVL#DPR2    CAPACITY SET AT DAILY PLAN
  270  (10E) CHARACTER     4  CPLIVLAWS      NOT USED
  274  (112) CHARACTER     4  CPLIVLDPAWS    NOT USED
  278  (116) CHARACTER     6  *              FREE
```

Cross reference

```
                                             HEX         HEX
NAME                                         OFFSET      VALUE     LEVEL
____                                         ______    ________    _____
CPLACCM22#                                       D2                    4
CPLAUTO                                          D1                    4
CPLAWS_VARY                                      9F          01        5
CPLBDY22                                         13                    3
CPLCAWS02                                        C0                    4
CPLCMDLNK                                        D0          08        5
CPLCMDSTA                                        D0          04        5
CPLCOFF_DATE                                     B8                    5
CPLCOFF_TIME                                     BC                    5
CPLCOFF02                                        B8                    4
CPLCOM22                                          0                    2
CPLCREVT                                         7C                    4
CPLCWS_STAT                                      B5                    4
CPLDATE                                          7C                    5
CPLDESCR22                                       18                    4
CPLDST22                                          6                    4
CPLEOPT22#                                       DC                    4
CPLEYE22                                         13                    4
CPLFLGTWS                                        D0                    4
CPLFLG02                                         9F                    4
CPLFULLNK                                        D0          02        5
CPLIVL#DPPS                                     102                    2
CPLIVL#DPR1                                     108                    2
CPLIVL#DPR2                                     10C                    2
CPLIVL#PS                                       100                    2
CPLIVL#R1                                       106                    2
CPLIVL#R2                                       10A                    2
CPLIVLAWS                                       10E                    2
CPLIVLDP                                        104          40        3
CPLIVLDPAWS                                     112                    2
CPLIVLFD                                         EC                    3
CPLIVLFR                                         EC                    2
CPLIVLFT                                         F2                    3
```

CPLIVLF2	104		2
CPLIVLMC	104	80	3
CPLIVLNP	9F	20	5
CPLIVLNU	9F	40	5
CPLIVLOI22	EC		1
CPLIVLTD	F6		3
CPLIVLTO	F6		2
CPLIVLTT	FC		3
CPLIVL22#I	B0		4
CPLKEY22	0		3
CPLKF22	E		4
CPLKID22	0		4
CPLLIMIT	D4		4
CPLLNKDW	D0	20	5
CPLLNKVA	D0	80	5
CPLLNK02	CF		4
CPLMAX15	B2		4
CPLOFF_PEND	9F	04	5
CPLPREDT	A0		5
CPLPREEV	A0		4
CPLPREP	9F	80	5
CPLPRETM	A4		5
CPLREC22	0		1
CPLRENGT	D7		4
CPLRLTYP	9E		4
CPLRSEQ#02	C6		4
CPLRTR_PEND	9F	02	5
CPLSSEQ#02	C4		4
CPLSTAVA	D0	40	5
CPLSTC	9F	10	5
CPLSUDS	A8		4
CPLSUMA	68		4
CPLSUMC	38		4
CPLSUMC#	38		5
CPLSUMCE	3C		5
CPLSUMCES	C7		4
CPLSUMCR	40		5
CPLSUMCRS	C8		4
CPLSUME	74		4
CPLSUMI	44		4
CPLSUMI#	44		5
CPLSUMIE	48		5
CPLSUMIES	C9		4
CPLSUMIR	4C		5
CPLSUMIRS	CA		4
CPLSUML	78		4
CPLSUMNRR	6C		4
CPLSUMR	58		4
CPLSUMR#	58		5
CPLSUMRE	5C		5
CPLSUMRES	CC		4
CPLSUMS	50		4
CPLSUMS#	50		5
CPLSUMSE	54		5
CPLSUMSES	CB		4
CPLSUMU	70		4
CPLSUMW	60		4
CPLSUMW#	60		5
CPLSUMWE	64		5
CPLSUMWES	CD		4
CPLSUMX	84		4
CPLTIME	80		5
CPLTRT02	B6		4
CPLUX009	9C	20	5
CPLVERS22	17		4
CPLVMAX15	D8		4
CPLWRTST	D0	10	5

CPLWSDYN	96	02	5
CPLWSN22	2		4
CPLWSREP	91		4
CPLWSR1C	96	80	5
CPLWSR1FLG	96		4
CPLWSR1IU#	94		4
CPLWSR1N	92		4
CPLWSR2C	9C	80	5
CPLWSR2FLG	9C		4
CPLWSR2IU#	9A		4
CPLWSR2N	98		4
CPLWSSTA	B4		4
CPLWSSTAT	9C	40	5
CPLWSTWS	CE		4
CPLWSTYP	90		4
CPLWSVAR22	EC		2
CPLWSVIRT	96	10	5
CPLWSWAIT	96	20	5
CPLWSZCAGENT	96	08	5
CPLWSZNOINFO	96	04	5
CPLWTO	9F	08	5

CPLREC24 - Current plan job recovery record

Name : DCLCPR24

Function:
This segment declares a jobname table record, physically located in EQQCPxDS and EQQNCPDS.
The CPLREC24 record is structured as follows:
1 CPLREC24
2 common data
2 message line entries ( )

Common data segment

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	208	CPLREC24	JOB RECOVERY RECORD
0	(0)	CHARACTER	8	CPL24KEY	KEY OF RECORD
0	(0)	CHARACTER	2	CPLTYP24	TYPE IS ALWAYS 24
2	(2)	UNSIGNED	3	CPLOCC24	OCCURENCE NUMBER
5	(5)	UNSIGNED	1	CPLOPR24	OPERATION NUMBER
6	(6)	SIGNED	2	CPLSEQ24	SEQUENCE NUMBER, 1ST=0001 IS NOT USED
8	(8)	CHARACTER	81	CPL24BDY	BODY MUST BE 85
8	(8)	CHARACTER	4	CPL24EYE	EYE CATCHER
12	(C)	CHARACTER	16	CPL24ADID	OCCURRENCE ADID
28	(1C)	CHARACTER	10	CPL24IA	OCCURRENCE INPUT ARRIVAL YYMMDDHHMM
28	(1C)	CHARACTER	6	CPL24IAD	IA DATE
34	(22)	CHARACTER	4	CPL24IAT	IA TIME
38	(26)	CHARACTER	8	CPL24JREID	JOB ID OF RECOVERY JOB
38	(26)	CHARACTER	4	CPL24JREIDP	JOBID PREFIX
42	(2A)	SIGNED	4	CPL24JREIDN	JOB ID NUMBER USED IN DATA PROCESSING IS NOT DISPLAYED
46	(2E)	CHARACTER	4	CPL24WSN	WORKSTATION NAME
50	(32)	SIGNED	1	CPL24RJST	RECOVERY JOB STATUS
51	(33)	CHARACTER	1	CPL24FLAGS	FLAGS

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
				CPL24COMP	1= JOBLOG IS IN COMPRESSED FORMAT (CSRCESRV COMPR.)
				CPL24NOLOG	1= NO JOBLOG FOUND
				CPL24FAIL	JOB FAIL
				*	RESERVED
52	(34)	SIGNED	4	CPL24TOTSZ	TOTAL SIZE OF RECORD, BEFORE THE SPLIT INTO TWO OR MORE RECORDS AND BEFORE COMPRESSION
56	(38)	SIGNED	4	CPLSIZ24	SIZE OF RECORD WHEN IN CP
56	(38)	SIGNED	4	CPL24CSZ	COMPRESSED SIZE (INCL HEADER)
60	(3C)	CHARACTER	10	CPL24START	RECOVERY JOB START
60	(3C)	CHARACTER	6	CPL24STARTD	DATE | BLANK
66	(42)	SIGNED	4	CPL24STARTT	TIME IN SEC*100 | 0
70	(46)	CHARACTER	10	CPL24END	RECOVERY JOB END TIME
70	(46)	CHARACTER	6	CPL24ENDD	DATE | BLANK
76	(4C)	SIGNED	4	CPL24ENDT	TIME IN SEC*100 | 0 TIME IN SEC*100 | 0
80	(50)	SIGNED	4	CPL#LINE24	NUMBER OF JOBLOG RECORDS
84	(54)	SIGNED	4	CPLLO24	OFFSET TO LAST MESSAGE LINE
88	(58)	CHARACTER	1	*	RESERVED
89	(59)	CHARACTER	1	CPL24ST	STATUS OF JOBLOG ARCHIVING: ' ' =NOT REQUIRED W=WAITING FOR RETRIEVAL C=RETRIEVE COMPLETED E=RETRIEVE ENDED IN ERROR
90	(5A)	CHARACTER	1	CPL24VERS	VERSION
91	(5B)	CHARACTER	1	*	RESERVED
92	(5C)	CHARACTER	8	CPL24JID	ID OF JOB TO RECOVER
100	(64)	CHARACTER	8	CPL24RJID	RECOVERY JOB ID TO DISPLAY
108	(6C)	SIGNED	4	CPL24DUR	RECOVERY JOB DURATION (SEC*100)
112	(70)	SIGNED	4	CPL24PROMPTID	RECOVERY PROMPT ID
116	(74)	CHARACTER	64	CPL24PRTMSG	RECOVERY MESSAGE
180	(B4)	CHARACTER	1	CPL24PRTSTAT	RECOVERY PROMPT STATUS: ' ' = NO PROMPT 'A' = ASKED (NO REPLY) 'N' = REPLY WITH N 'Y' = REPLY WITH Y

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
181	(B5)	CHARACTER	1	CPL24RECTYPE	
					RECOVERY TYPE USED FOR DISPLAY FOR CODE USE EX0: S = STOP C = CONTINUE R = RERUN
182	(B6)	CHARACTER	4	CPL24ERRC	ERROR CODE
186	(34)	CHARACTER	4	CPL24MAINWS	RECOVER JOB WORKSTATION
190	(BE)	CHARACTER	2	*	RESERVED
192	(C0)	CHARACTER	4	CPL24ERRCFIX	ERROR CODE FIXED
196	(C4)	SIGNED	12	*	RESERVED
208	(D0)	CHARACTER		CPLDATA24	DATA PORTION

Messages line segment

The value of CPL#LINE24 in the common data segment describes the number of message line segments.

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	*	CPLSYSOUT24	
0	(0)		4	CPLMSG24L	LENGTH OF SYSOUT MESSAGE LINE
4	(4)	CHARACTER	2	CPLMSG24FLG	FLAGS
4	(4)	CHARACTER	1	CPLLTYPE24	
					LINE TYPE: J = JOBLOG ' ' =USERSYSOUT
5	(5)		1	*	RESERVED
6	(6)		2	CPLMSG24NO	OFFSET TO NEXT MESSAGE
8	(8)		*	CPLMSG24	MESSAGE

Cross reference

Name	Hex Offset	Hex Value	Level
CPL#LINE24	50		3
CPLDATA24	D0		2
CPLLO24	54		3
CPLLTYPE24	4		3
CPLMSG24	8		2
CPLMSG24FLG	4		2
CPLMSG24L	0		2
CPLMSG24NO	6		2
CPLOCC24	2		3
CPLOPR24	5		3
CPLREC24	0		1

Name	Hex Offset	Hex Value	Level
CPLSEQ24	6		3
CPLSIZ24	38		3
CPLSYSOUT24	0		1
CPLTYP24	0		3
CPL24ADID	C		3
CPL24BDY	8		2
CPL24COMP	33	80	4
CPL24CSZ	38		4
CPL24DUR	6C		2
CPL24END	46		3
CPL24ENDD	46		4
CPL24ENDT	4C		4
CPL24ERRC	B6		2
CPL24ERRCFIX	C0		2
CPL24EYE	8		3
CPL24FAIL		20	4
CPL24FLAGS	33		3
CPL24IA	1C		3
CPL24IAD	1C		4
CPL24IAT	22		4
CPL24JID	5C		2
CPL24JREID	26		3
CPL24JREIDN	2A		4
CPL24JREIDP	26		2
CPL24KEY	0		2
CPL24MAINWS	BA		2
CPL24NOLOG	33	40	4
CPL24PROMPTID	70		2
CPL24PRTMSG	74		2
CPL24PRTSTAT	B4		2
CPL24RECTYPE	B5		2
CPL24RJID	64		2
CPL24RJST	32		3
CPL24ST	59		2
CPL24START	3C		3
CPL24STARTD	3C		4
CPL24STARTT	42		4
CPL24TOTSZ	34		3
CPL24VERS	5A		2
CPL24WSN	2E		3

CPLREC3C - Current plan occurrence record

Name : DCLCPR3C

Function:
This segment declares a current plan occurrence record, physically located in EQQCPXDS and EQQNCPDS.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC3C	OCCURRENCE RECORD
0	(0)	CHARACTER	302	CPLCOM3C	COMMON PART OF RECORD
0	(0)	CHARACTER	12	CPLKEYOC	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 03
2	(2)	CHARACTER	4	CPLNDXOC	KEY TO OCCURRENCE RECORD
2	(2)	UNSIGNED	3	CPLOC03C	OCC NUMBER
5	(5)	UNSIGNED	1	CPLOP03C	OP. NUMBER SET TO ZERO
6	(6)	CHARACTER	6	CPLKF03C	should contain blanks
12	(C)	CHARACTER	290	CPLBDYOC	BODY OF OCCURRENCE RECORD
12	(C)	CHARACTER	4	CPLEYE3C	EYECATCHER
16	(10)	UNSIGNED	1	CPLVERS3C	VERSION NUMBER
17	(11)	CHARACTER	8	CPLGROUP3C	AUTHORITY GROUP
25	(19)	CHARACTER	3	*	FREE CATC
28	(1C)	CHARACTER	16	CPLADIOC	APPLICATION ID
44	(2C)	CHARACTER	10	CPLIAOC	INPUT ARRIVAL FROM LTP
44	(2C)	CHARACTER	6	CPLIADOC	DATE
50	(32)	CHARACTER	4	CPLIATOC	TIME
54	(36)	CHARACTER	10	CPLIMOC	MODIFIED INPUT ARRIVAL OR IA FROM LTP
54	(36)	CHARACTER	6	CPLIMDOC	DATE
60	(3C)	CHARACTER	4	CPLIMTOC	TIME
64	(40)	SIGNED	2	CPLNETOC	NETWORK ID
66	(42)	CHARACTER	24	CPLTXTOC	APPL.DESCRIPTIVE TEXT
90	(5A)	CHARACTER	16	CPLOIDOC	OWNER ID
106	(6A)	CHARACTER	24	CPLOTXOC	OWNER DESCRIPTION
130	(82)	CHARACTER	10	CPLDLOC	DEADLINE
130	(82)	CHARACTER	6	CPLDLDOC	DATE
136	(88)	CHARACTER	4	CPLDLTOC	TIME
140	(8C)	CHARACTER	1	CPLPRIOC	PRIORITY
141	(8D)	CHARACTER	1	CPLSTAOC	S=STARTED,U=UNDECIDED, D=DELETED,W=WAITING C=COMPLET,E=ENDED IN ERROR P=PENDING PREDECESSOR M=MANDATORY PEND PRED
142	(8E)	CHARACTER	10	CPLAAOC	ACTUAL ARRIVAL BLANK
142	(8E)	CHARACTER	6	CPLAADOC	DATE
148	(94)	CHARACTER	4	CPLAATOC	TIME
152	(98)	CHARACTER	10	CPLACOC	ACTUAL COMPLETION BLANK
152	(98)	CHARACTER	6	CPLACDOC	DATE
158	(9E)	CHARACTER	4	CPLACTOC	TIME
162	(A2)	SIGNED	2	CPLCOPOC	NDX TO NON-COMPLETE OP WITH EARLIEST LATEST OUT
164	(A4)	UNSIGNED	4	CPLRDUOC	REMAINING DUR (sec) ON CURRENT CRITICAL PATH

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
168	(A8)	SIGNED	4	CPLROPOC	NUMBER OF REMAINING OPS ON CURRENT CRITICAL PATH
172	(AC)	SIGNED	4	CPLOPOC	NUMBER OF OPS IN OCC
172	(AC)	SIGNED	4	CPLsuccDep	NUMBER OF successors dependencies VALID ONLY FOR 'M' occ
176	(B0)	SIGNED	4	CPLOPCOC	NUMBER OF OPS COMPLETED
180	(B4)	SIGNED	4	CPLEROC	NBR OF OPS ENDED IN ERROR
184	(B8)	SIGNED	4	CPLUNOC	NUMBER OF OPS UNDECIDED
188	(BC)	SIGNED	4	CPLSTOC	NUMBER OF OPS STARTED
192	(C0)	CHARACTER	4	CPLERROC	ERROR CODE OF OCC
196	(C4)	SIGNED	2	CPLRSROC	INDEX OF REM.SLACK CHAIN
198	(C6)	SIGNED	2	CPLRSOOC	OFFSET IN REM SLACK REC.
200	(C8)	SIGNED	2	CPLUOROC	INDEX OF UNDECIDED CHAIN R
202	(CA)	SIGNED	2	CPLUOOOC	OFFSET IN UNDECIDED REC
204	(CC)	CHARACTER	1	CPLSWIOC	BIT SWITCHES
		1...		CPLREROC	1 MEANS RERUN REQUESTED
		.1..		CPLDEFOC	1 = DEFAULT RULE FOR PEND
		..1.		CPLEXPOC	1=CURR. VALUE IN AAOC FROM OP. WITH EXPL. INP. ARRIVAL
		...1		CPLMCPOC	1 MEANS ADDED BY MCP
		.... 1...		cpllltpoc	came from LTP at last DP
		.... .1..		CPLMONC	any monitored op in occ
		.... ..1.		CPLMPEND	original MPEND
		1		*	FREE
205	(CD)	CHARACTER	1	CPLADDOC	HOW ADDED TO CP D MEANS MCP-DIALOG E MEANS ETT P MEANS PIF A MEANS AR BLANK MEANS DP
206	(CE)	CHARACTER	16	cpljvtab	jcl var table name jcla
222	(DE)	CHARACTER	16	cplOCCgroup	occurrence group ref ADGA
238	(EE)	CHARACTER	16	cplcal	calendar name
254	(FE)	CHARACTER	8	cplluseroc	lastupdate user-id
262	(106)	CHARACTER	8	cplocctok	occ token TOD D90C
270	(10E)	CHARACTER	1	cplTWSocc	Y means tws occurrence
271	(10F)	UNSIGNED	1	cplettinfo	Extended ETT info area 0 = not allocated 1 = allocated
272	(110)	SIGNED	2	cplrunidc	ADR run cycle id
274	(112)	CHARACTER	2	*	free
276	(114)	SIGNED	4	cplxcoc	number of dum compl ops
280	(118)	CHARACTER	22	*	free CWHCCZSC
302	(12E)	CHARACTER	*	cplvar3c	variable part of record

ETT VARIABLES

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	159	cplettvar	ETT variables
0	(0)	CHARACTER	159	cplett3c	ETT entry
0	(0)	CHARACTER	44	cplettcrit	ETT criteria
44	(2C)	CHARACTER	1	cpletttyp	J or R
45	(2D)	CHARACTER	8	cplettjob	Jobname
53	(35)	CHARACTER	8	cplettjid	Jobid
61	(3D)	UNSIGNED	1	cplettgrootl	GDG root length

rootcplettevnam1coloncplettgrootl31A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
62	(3E)	CHARACTER	44	cplettevnam	Full ETT event name
106	(6A)	CHARACTER	53	*	Free

Cross reference

Name	Hex Offset	Hex Value	Level
CPLEROC	B4		4
CPLOPOC	AC		4
CPLSTOC	BC		4
CPLsuccDep	AC		5
CPLUNOC	B8		4
cplxcoc	114		4
CPLAADOC	8E		5
CPLAAOC	8E		4
CPLAATOC	94		5
CPLACDOC	98		5
CPLACOC	98		4
CPLACTOC	9E		5
CPLADDOC	CD		4
CPLADIOC	1C		4
CPLBDYOC	C		3
cplcal	EE		4
CPLCOM3C	0		2
CPLCOPOC	A2		4
CPLDEFOC	CC	40	5
CPLDLDOC	82		5
CPLDLOC	82		4
CPLDLTOC	88		5
CPLERROC	C0		4
cplettcrit	0		3
cplettevnam	3E		3
cplettgrootl	3D		3
cplettinfo	10F		4
cplettjid	35		3
cplettjob	2D		3
cpletttyp	2C		3
cplettvar	0		1
cplett3c	0		2
CPLEXPOC	CC	20	5
CPLEYE3C	C		4

Name	Hex Offset	Hex Value	Level
CPLGROUP3C	11		4
CPLIADOC	2C		5
CPLIAOC	2C		4
CPLIATOC	32		5
CPLIMDOC	36		5
CPLIMOC	36		4
CPLIMTOC	3C		5
cpljvtab	CE		4
CPLKEYOC	0		3
CPLKF03C	6		4
cplltpoc	CC	08	5
cplluseroc	FE		4
CPLMCPOC	CC	10	5
CPLMONC	CC	04	5
CPLMPEND	CC	02	5
CPLNDXOC	2		4
CPLNETOC	40		4
cplOCCgroup	DE		4
cplocctok	106		4
CPLOC03C	2		5
CPLOIDOC	5A		4
CPLOPCOC	B0		4
CPLOP03C	5		5
CPLOTXOC	6A		4
CPLPRIOC	8C		4
CPLRDUOC	A4		4
CPLREC3C	0		1
CPLREROC	CC	80	5
CPLROPOC	A8		4
CPLRSOOC	C6		4
CPLRSROC	C4		4
cplrunidc	110		4
CPLSTAOC	8D		4
CPLSWIOC	CC		4
cplTWSocc	10E		4
CPLTXTOC	42		4
CPLUOOOC	CA		4
CPLUOROC	C8		4
cplvar3c	12E		2
CPLVERS3C	10		4

CPLREC3P - Current plan operation record

Name: DCLCPR3P

Function:
This segment declares a current plan operation record, physically located in EQQCPXDS and EQQNCPDS.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC3P	OPERATION RECORD
0	(0)	CHARACTER	436	CPLCOM3P	
0	(0)	CHARACTER	12	CPLKEYOP	
0	(0)	CHARACTER	2	CPLKYP3P	TYPE IS ALWAYS 03
2	(2)	CHARACTER	4	CPLNDXOP	KEY TO OCCOP RECORD
2	(2)	UNSIGNED	3	CPLOC03P	OCCURENC NUMBER
5	(5)	UNSIGNED	1	CPLOP03P	OPERATION NUMBER
6	(6)	CHARACTER	6	CPLKF03P	SHOULD CONTAIN blanks
12	(C)	CHARACTER	424	CPLBDYOP	BODY OF OPERATION RECORD
12	(C)	CHARACTER	4	CPLEYE3P	EYECATCHER
16	(10)	UNSIGNED	1	CPLVERS3P	VERSION NUMBER
17	(11)	CHARACTER	8	CPLGROUP3P	AUTHORITY GROUP
25	(19)	CHARACTER	1	cplClnStat	Clean Up status D90C see dclrccon D90C blank = none D90A R = request OPInfo W = waiting OPInfo D90A O = avail OPInfo D90A I = initiated D90A S = started D90A C = completed D90A E = ended in error D90A
26	(1A)	CHARACTER	16	CPLADIOP	APPLICATION ID
42	(2A)	CHARACTER	10	CPLIAOP	APPL INP. ARRIVAL AFTER MCP
42	(2A)	CHARACTER	6	CPLIADOP	DATE
48	(30)	CHARACTER	4	CPLIATOP	TIME
52	(34)	SIGNED	2	CPLNETOP	NETWORK ID
54	(36)	CHARACTER	24	CPLTXTOP	OP TEXT DESCRIPTION
78	(4E)	CHARACTER	8	CPLJBNOP	OP OS JOBNAME BLANK
86	(56)	CHARACTER	8	CPLJIDOP	JOB ID
86	(56)	CHARACTER	4	CPLJIDOPP	Job id prefix for TWS
90	(5A)	SIGNED	4	CPLJIDOPN	Job id number(TWS)
94	(5E)	CHARACTER	6	CPLIDOP	OPERATION ID
94	(5E)	CHARACTER	4	CPLWSOP	WS NAME
98	(62)	SIGNED	2	CPLNUMOP	OPERATION CREATION NUMBER
100	(64)	CHARACTER	1	CPLJCLOP	JOBCLASS OR SYSOUTCLASS BLANK
101	(65)	BITSTRING	1	CPLFLG05	Flag byte 5
		1...		CPLDEFLT	Oper ia defaulted
		.1..		CPLPSUWAIT	1 = wait for PSU done D90C set by dialog when apply logic needed but cleanup = none
		..1.		cplCP15req	on = Ask opi CP15 in progress
		...1		cplmhld	on=manually held op HOPa
		.... 1...		cplnop	on=nop op HOPA
		.... .1..		cplexec	on=execute operation HOPC
		.... ..1.		cplclncm	on=clean up comm issued
		1		CPLSSEQ	CPLISOP used as SSEQ
102	(66)	CHARACTER	8	CPLFRMOP	FORMNUMBER BLANK
110	(6E)	CHARACTER	10	CPLPSOP	PLANNED START d88c

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
110	(6E)	CHARACTER	6	CPLPSDOP	DATE BLANK
116	(74)	SIGNED	4	CPLPSTOP	TIME 0 d88c
120	(78)	CHARACTER	10	CPLPEOP	PLANNED END
120	(78)	CHARACTER	6	CPLPEDOP	DATE BLANK
126	(7E)	SIGNED	4	CPLPETOP	TIME 0 d88c d88c
130	(82)	CHARACTER	10	CPLOIOP	OPERATION INPUT ARRIVAL
130	(82)	CHARACTER	6	CPLOIDOP	DATE BLANK
136	(88)	CHARACTER	4	CPLOITOP	TIME BLANK
140	(8C)	CHARACTER	10	CPLODOP	OPERATION DEADLINE
140	(8C)	CHARACTER	6	CPLODDOP	DATE BLANK
146	(92)	CHARACTER	4	CPLODTOP	TIME BLANK
150	(96)	CHARACTER	10	CPLLOOP	LATEST OUT FOR OP d88c
150	(96)	CHARACTER	6	CPLLODOP	DATE
156	(9C)	SIGNED	4	CPLLOTOP	TIME d88c
160	(A0)	CHARACTER	10	CPLASOP	ACTUAL START
160	(A0)	CHARACTER	6	CPLASDOP	DATE BLANK
166	(A6)	SIGNED	4	CPLASTOP	TIME 0 d88c d88c
170	(AA)	CHARACTER	10	CPLAAOP	ACTUAL ARRIVAL d88c
170	(AA)	CHARACTER	6	CPLAADOP	DATE BLANK
176	(B0)	SIGNED	4	CPLAATOP	TIME 0 d88c
180	(B4)	CHARACTER	10	CPLISOP	INTERMEDIATE START (IF OP
180	(B4)	CHARACTER	6	CPLISDOP	DATE BLANK WAS IRUPT)
180	(B4)	UNSIGNED	2	CPLSSEQOP	SUBMIT SEQUENCE
180	(B4)	CHARACTER	2	cplisdopf2	SUBMIT SEQUENCE
182	(B6)	CHARACTER	4	cplisdopl4	(WHEN JOB, STC OR WTO)
186	(BA)	SIGNED	4	CPLISTOP	TIME 0 d88c d88c
190	(BE)	CHARACTER	10	CPLAEOP	ACTUAL END d88c
190	(BE)	CHARACTER	6	CPLAEDOP	DATE BLANK
196	(C4)	SIGNED	4	CPLAETOP	DATE 0 d88c
200	(C8)	SIGNED	4	CPLEDUOP	ESTIMATED DURATION d88c
204	(CC)	UNSIGNED	4	CPLADUOP	ACTUAL DURATION d88c
208	(D0)	SIGNED	2	CPLOPIRET	ask opinfo retry counter
210	(D2)	SIGNED	2	CPLPSOP	PARALLEL SERVERS REQUIRED
212	(D4)	SIGNED	2	CPLR1OP	WS RESOURCES REQUIRED
214	(D6)	SIGNED	2	CPLR2OP	WS RESOURCES REQUIRED
216	(D8)	CHARACTER	1	CPLCSTOP	CURRENT STATE W - WAITING A - WAITING FOR ARRIVAL R - READY * - READY PREV WS NONREP S - STARTED I - INTERRUPTED U - UNDECIDED C - COMPLETED E - ENDED IN ERROR
217	(D9)	CHARACTER	4	CPLERROP	ERROR CODE
221	(DD)	CHARACTER	1	CPLAECOP	AUTOM. ERROR COMPL.(YESNO)
222	(DE)	CHARACTER	1	CPLPRIOP	PRIORITY
223	(DF)	CHARACTER	1	CPLXSTOP	EXTENDED STATUS
224	(E0)	SIGNED	2	CPLERKOP	INDEX TO ERROR RECORD 0
226	(E2)	SIGNED	2	CPLEROOP	OFFSET IN ERROR RECORD 0
228	(E4)	SIGNED	2	CPLSUOP	NUMBER OF SUCCESSORS
230	(E6)	SIGNED	2	CPLPROP	NUMBER OF PREDECESSORS
232	(E8)	SIGNED	2	CPLDEOP	NUMBER OF SUCCESSORS AND PREDECESSORS
234	(EA)	SIGNED	2	CPLPCOP	NUMBER OF PREDECESSORS COMPLETED
236	(EC)	SIGNED	2	CPLSROP	NO. OF SPECIAL RESOURCES

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
238	(EE)	BITSTRING	1	CPLFLGOP	FLAGS
		1...		CPLREROP	RERUN RECORD FOR THIS OP
		.1..		CPLEXTOP	ON=VALIDATION EXIT PASSED
		..1.		CPLASSOP	ON=OP. ASSUMED COMPLETED
		...1		CPLSPIA	ON=SPECIFIED IA FOR OP
		.... 1...		CPLSPDL	ON=SPECIFIED DL FOR OP
		.... .1..		CPLASUOP	ON=AUTO SUBMISSION OF JOB
		.... ..1.		CPLAJR	ON=AUTO HOLDRELEASE
		1		CPLLATEP	ON=LATE OP MSG ISSUED
239	(EF)	BITSTRING	1	CPLFLGO2	FLAGS
		1...		CPLSUBOP	ON=JOB SUBMITTED
		.1..		CPLTJTOP	ON=TIME JOB
		..1.		CPLPRPOP	ON=PREP WS EXISTS FOR THIS .. CPU, BUT OP IS .. NOT COMPLETE
		...1		CPLRELOP	ON=RELEASE AS SOON AS POSS.
		.... 1...		CPL2EVOP	ON=TYPE 2 EVENT READ
		.... .1..		CPLHRUOP	ON=HIGH RETCODE USED
		.... ..1.		CPLPENDP	A PENDING PREDECESSOR
		1		CPLLONGP	ON=LONG DUR MS ISSUED
240	(F0)	CHARACTER	8	CPLMCPUP	TIME OF LAST MCP UPDATE
248	(F8)	CHARACTER	1	CPLDEPTOP	DEPENDEN TYPE (PSRT) R - conditional predeces T - conditional success
249	(F9)	BITSTRING	1	CPLFLGO3	3rd flag byte WLRA
		1...		CPLRESTA	restartable operation WLRA
		.1..		CPLDIRES	inst parm restart WLRA
		..1.		CPLRERUT	rerouteable operation WLRA
		...1		CPLDIRER	inst parm reroute WLRA
		.... 1...		CPLWRER	op WAS rerouted WLRA
		.... .1..		CPLDWTO	DEADLINE WTO WANTED TNSA
		.... ..1.		CPLDWTOS	DEADLINE WTO REQ SENT TNSA
		1		CPLDWTOP	DEADLINE WTO REQ PROC TNSA
250	(FA)	UNSIGNED	2	CPLHRCOP	HIGHEST RETCODE NOT IN ERR
252	(FC)	SIGNED	2	CPLTMPOP	temporary save area
254	(FE)	CHARACTER	4	CPLALTWS	Alternate WS name WLRA
258	(102)	SIGNED	2	CPLMXLVL	Max nesting level JODA
258	(102)	SIGNED	2	CPLMPREDopno	- used only by ISPF dialogue to pass info in GETL,MSUCC buffer
260	(104)	CHARACTER	16	CPLudata	User field NVIA
276	(114)	BITSTRING	1	cplflag05	more flags
		1...		cplMoreRun	0 = First run D90C 1 = At least 1 run D90A set by AB3P event D90A
		.1..		cplJLInfo	0 = JLInfo to be req. D90A 1 = JLInfo requested. D90A set by AB3P event D90A reset when JlInfo D90A requested
		..1.		cplOpInfo	0 = last OPerInfo not D90A available D90A 1 = last OPerInfo D90A available
		...1		cplPSUdone	0 = PSU not compl. D90A 1 = PSU completed D90A
		.... 1...		cplUserSYS	1 = need user sysout D90A
		.... .1..		cplimmerr	1 = imm from error state (set by A3P if error D90A status) D90A
		.... ..1.		cplsyssfail	jes sub failed
		1		cplExpJCL	need Expanded JCL MVAD90C

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
277	(115)	CHARACTER	1	cplJRstat	Joblog retrieval staus ' ' = no action yet 'I' = initiated 'R' = Reinitiated 'S' = started (j0-evt) 'C' = complete(j2-evt) 'E' = error (j2-evt)
278	(116)	CHARACTER	2	cplwlm	WLM fields D63a
278	(116)	CHARACTER	1	cpljobcrt	Critical Job D63a
279	(117)	CHARACTER	1	cpljobpol	Late Job Policy D63a
280	(118)	CHARACTER	8	cpltoken_full	operation full token
280	(118)	CHARACTER	4	cpltoken_pre	prefix
284	(11C)	UNSIGNED	4	cpltoken	opctoken
288	(120)	CHARACTER	8	cplEXdest	Execution dest
296	(128)	SIGNED	2	CPLASIDJOB	OPC ASID D63a
298	(12A)	CHARACTER	1	cplClnTyp	Clean up Type= D90C A = automatic, D90A I = immediate, D90A M = manual, D90A N = none D90A
299	(12B)	CHARACTER	1	cpltwsjob	Y,S only in Sym,N d94a
300	(12C)	CHARACTER	8	cplstch	status was changed
300	(12C)	CHARACTER	4	cplstchd	yydddF
304	(130)	CHARACTER	4	cplstcht	hhmmssth
308	(134)	CHARACTER	8	cplluserop	lastupdate user-id
316	(13C)	CHARACTER	8	cplJLOdest	Job Log destination D54A
324	(144)	CHARACTER	4	cplPSUtoken	PSU request token D90C
328	(148)	UNSIGNED	1	cplOpEx03p	Operation num extension
329	(149)	CHARACTER	1	cplclnRes	clean up result
330	(14A)	CHARACTER	1	cplclnDia	clean up from Dialog
331	(14B)	CHARACTER	8	cplEQQCLjid	last EQQCLEAN jobid
331	(14B)	CHARACTER	8	cplsubtoken	HTTP job submission token
339	(153)	CHARACTER	1	cplflag06	free to use
		1...		cplmonp	oper monitoring flag
		.1..		cplexpjclUSED	last run used expanded BGDC
		..1.		cplsending	on=S-sending exstat
		...1		cplsuber	on=E-OSUF exstat
		.... 1...		cplerfail	on=E-FAIL exstat
		.... .1..		cplrecprmp	on=E-prmpt exstat
		.... ..1.		cplrecjob	on=E rec job running
		1		cplrecrun	on=E(R,S,C) not restart
340	(154)	CHARACTER	8	cplEQQCLJBN	last EQQCLEAN jobname
340	(154)	CHARACTER	8	cplDOAtime	time the op was started
348	(15C)	CHARACTER	1	cplflag07	
		1...		cplRec24Is	CP24 exists
		.1..		cplRecEnd	op. completed by continue recovery action
		..1.		cplscript	centralized script used
		...1		cplFromPLC	on = from PLC CCUA
		.... 1...		cplWASUJ	on = from WASUJ CCUA
		.... .1..		cplWaitSE	on = wait SE CCUA
		.... ..1.		cplWaitWQA	on = wait WS delay
		1		cplURGmsg	on = URG msg issued
349	(15D)	UNSIGNED	1	cplExtInf	Extended info area 25ACCUC 0 = not allocated 1 = allocated
350	(15E)	CHARACTER	1	cplTWSjbnm	criterion used to build BOGa the TWS jobname in sym BOGa
351	(15F)	CHARACTER	1	cplAutInf	26CBOGc CHWC N = not allocated CHWA Y = allocated CHWA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
352	(160)	SIGNED	4	cplftwRC	RC oper running on FTW
356	(164)	CHARACTER	8	cplRUSER	RACF user from EQQUX001 BUTA
364	(16C)	SIGNED	2	cpl3pSPLEX	sysplex id CCUA
366	(16E)	CHARACTER	1	cplflag08	
		1...		cplWLMprom	WLM failure msg flag
		.1..		cplWLMmsg	WLM promotion flag
		..1.		cplFTSAnop	job nopped on standard agent hosted by master
		...1		cplCritJobPred	Critical predecessor CRFA
		.... 1...		cpldprem	on = recovered by cond
		.... .1..		cplTZONE	on = TZONE applied CRPA
		.... ..1.		cplcondrjob	on = cond recovery job
		1		cplNoEval	on = do not eval cond succ
367	(16F)	BITSTRING	1	cplflag09	
		1...		cplstepdep	on = this 3p has conds with step dep
		.1..		cplStepSucc	on = this 3p has at least 1 csuc with step dep pointing to it
		..1.		cplzcentric	on = job on z-centric
		...1		cplRecovOn	on = recov set (not saved)
		.... 1...		cplnoerr	on = NOERROR occurred CZJA
		.... .1..		cplUnexpRC	on = unexpected RC occur
		.... ..1.		cplFakeSub	
		1		cplshadow	on = shadow job
368	(170)	CHARACTER	8	cplWLMclass	WLM Service Class
376	(178)	SIGNED	4	cplCritPath	Critical Path indicator for internal processing
380	(17C)	SIGNED	4	cplEarE	Earliest End for internal processing
384	(180)	CHARACTER	4	cplCrtPthX	Critical Path Origin Key
384	(180)	UNSIGNED	3	cplPthOc	Occ. Number
387	(183)	UNSIGNED	1	cplPthOp	Operation Number
388	(184)	CHARACTER	2	cplJESdest	Dest (x Mult Datastore) COBA
390	(186)	CHARACTER	8	cplVdest	last used Virtual Dest COBA
398	(18E)	SIGNED	2	cplTZOFF	TZOFF in mins CRPA CRPA
400	(190)	UNSIGNED	4	cplJobTblIdx	Job Table index CRFA
404	(194)	SIGNED	2	CPLCPROP	NUMBER OF COND PRED
406	(196)	SIGNED	2	CPLCSUOP	NUMBER OF COND SUCC
408	(198)	SIGNED	2	CPLCondTot	NUMBER OF CONDS T F U D
410	(19A)	SIGNED	2	CPLCond_T	NUMBER OF CONDS TRUE
412	(19C)	SIGNED	2	CPLCond_F	NUMBER OF CONDS FALSE
414	(19E)	SIGNED	2	CPLCond_U	NUMBER OF CONDS UNDEFINED
416	(1A0)	SIGNED	2	cplcdeop	NUMBER OF COND DEP
418	(1A2)	SIGNED	2	CPLPXOP	NUMBER OF PREDECESSORS DUMMY COMPLETED
420	(1A4)	CHARACTER	4	CPLOrigRC	job RC before emerh
424	(1A8)	SIGNED	2	CPLCondId	Cond id - Never stored, only used by GS to return extra info
426	(1AA)	UNSIGNED	1	cplRengInf	Remote engine job area
427	(1AB)	CHARACTER	1	cplrejst	Remote eng job substatus
428	(1AC)	SIGNED	2	cplbnd	num of bind requests
430	(1AE)	BITSTRING	1	cplflag10	
		1...		cpldyn	on = this 3p has conds
		.1..		cpldrecov	on=recovery in progress
		..1.		cplshoseq	on=shadow in oseq
		...1		cplMvDeadl	on=move deadl DCRA

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
		.... 1...		cplexit13rc	on=rc 12 from exit13 DGWA
		.... .1..		cplNoerrJT	NOERROR match JT DEWA
		.... ..1.		cplNoerrSkip	NOERROR Skip A3S in reappDEWA
		1		cplmpendp	- used only by ISPF,PIF dialogue to pass info in GETL,DEPC buffer
431	(1AF)	CHARACTER	1	cplTaskType	plugin task type
432	(1B0)	BITSTRING	1	cplflag11	
		1...		cplmppend	on if pend or mpend pred - used only by ISPF,PIF dialogue to pass info in GETL buffer
		.111 1111	*	free	
433	(1B1)	CHARACTER	3	*	added to make fix part multiple of 4. This becasue DDICT declare EXT3P as a subfield of a structure having a string of char with length = CP3p fix part len as first field

53A WARNING ATTENTION if you need to increase cp3p fix part make it always a multiple of 4

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
436	(1B4)	CHARACTER	*	CPLVAR03	VAR DATA

PREDECESSOR INFORMATION

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	12	CPLPREOP (*)	
0	(0)	CHARACTER	1	CPLPSWOP	PREDECESSOR SWITCHES
		1...		CPLPCOOP	PREDECESSOR COMPLETED
		.1..		CPLPNROP	PRED. WS WAS NON-REPORTING
		..1.		CPLPNDOP	PENDING PRED OCCURRENCE
		...1		cplCritPred	Critical Path indicator for internal processing
		.... 1...		CPLPMandPend	on is a mandatory pend.
		.... .111		*	RESERVED
1	(1)	CHARACTER	1	CPLPMatc	Originating matching criteria - C closest preceding - S same day - A absolute IVL - R relative IVL
2	(2)	SIGNED	2	CPLPTTOP	TRANSPORT TIME
4	(4)	CHARACTER	4	CPLPOP	PREDECESSOR OP'S KEY IN CP
4	(4)	UNSIGNED	3	CPLPOCOP	OCCURRENCE INDEX
7	(7)	UNSIGNED	1	CPLPOPOP	OPERATION NO.
8	(8)	CHARACTER	4	CPLPMatcIVL	resolution criteria IVL
8	(8)	SIGNED	2	CPLPIVLleft	- left side - min value -32500 means no left side
10	(A)	SIGNED	2	CPLPIVLright	- right side - min value 0 means same IA

SUCCESSOR INFORMATION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	6	CPLSUCOP (*)	
0	(0)	CHARACTER	1	CPLSSWOP	SUCCESSOR SWITCHES
		1...		CPLSCROP	CRITICAL PATH SUCCESSOR
		.111 1111		*	RESERVED
1	(1)	CHARACTER	1	*	RESERVED
2	(2)	CHARACTER	4	CPLSOP	SUCCESSOR OP'S KEY IN CP
2	(2)	UNSIGNED	3	CPLSOCOP	OCCURRENCE INDEX
5	(5)	UNSIGNED	1	CPLSOPOP	OPERATION NUMBER

SPECIAL RESOURCES

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	68	CPLSRSOP (*)	
0	(0)	CHARACTER	44	CPLSRNOP	NAME OF SPECIAL RESOURCE
44	(2C)	CHARACTER	1	CPLSRTOP	S = SHARED, X = EXCLUSIVE
45	(2D)	BITSTRING	1	*	flags
		1...		CPLSRAVOP	available
		.1..		CPLSRIUSOP	in use shared
		..1.		CPLSRIUXOP	in use exclusively
		...1		cplsrkept	was kept at error
		.... 1...		cplsrkeptx	in exclusive use
		.... .1..		cplrescnew	Q515W msg issued
		11		*	reserved
46	(2E)	CHARACTER	1	cplsrerr	on error flag
47	(2F)	UNSIGNED	1	cplsrwrs	reason for wait
48	(30)	SIGNED	4	cplsrnum	amount
52	(34)	SIGNED	4	cplsridx	resource index id
56	(38)	CHARACTER	8	cplAlcTime	allocation time
64	(40)	CHARACTER	1	cplsrAVAco	Set SR availaibility to NO Y , N , R or blank
65	(41)	CHARACTER	3	*	free

EXTENDED INFORMATION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	200	cpl3pExtInfo	
0	(0)	CHARACTER	54	cpl3pEXTname	Extended Job Name
54	(36)	CHARACTER	16	cpl3pSEname	Scheduling Env. Name CCUA
70	(46)	CHARACTER	130	*	free 25ACCUC

CONDITIONAL PREDECESSOR

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	8	CPLCPREOP (*)	
0	(0)	BITSTRING	1	CPLCPSWOP	flags
		1...		CPLCPNDOP	pending cond pred occ
		.1..		CPLCPCOOP	on = if external, CPRE not to be created
		..11 1111		*	free
1	(1)	CHARACTER	1	CPLCPMatc	Originating matching criteria - C closest preceding - S same day - A absolute IVL - R relative IVL
2	(2)	SIGNED	2	CPLCPOP_cid	Cond id
4	(4)	CHARACTER	4	CPLCPOP	Operation VSAM key
4	(4)	UNSIGNED	3	CPLCPOCOP	- Occ Number
7	(7)	UNSIGNED	1	CPLCPOPOP	- Opr Number

CONDITIONAL SUCCESSOR

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	8	CPLCSUCOP (*)	
0	(0)	BITSTRING	1	CPLCSSWOP	flags
1	(1)	BITSTRING	1	CPLCSUFLAG	
		1...		CPLCSUSTEP	on = cond succ with step dep pointing to this 3p
		.111 1111		*	free
2	(2)	SIGNED	2	CPLCSOP_cid	Cond id
4	(4)	CHARACTER	4	CPLCSOP	Operatiob VSAM key
4	(4)	UNSIGNED	3	CPLCSOCOP	- Occ Number
7	(7)	UNSIGNED	1	CPLCSOPOP	- Opr Number

EXTENDED INFORMATION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	200	cplRENG	
0	(0)	CHARACTER	16	cplRENG_jsname	ad id
16	(10)	CHARACTER	40	cplRENG_jobname	job name TWSd
16	(10)	CHARACTER	8	cplRENG_zjob	matched job name TWSz
56	(38)	CHARACTER	16	cplRENG_jsws	jobstream ws TWSd
56	(38)	CHARACTER	4	cplRENG_zws	matched workstation TWSz
72	(48)	CHARACTER	3	cplRENG_opno	operation number TWSz
75	(4B)	BITSTRING	1	cplRENG_flags	
		1...		cplRENG_comp	complete on bind err
		.111 1111		*	free
76	(4C)	CHARACTER	10	cplRENG_IA	matched input arrival
76	(4C)	CHARACTER	6	cplRENG_IAD	date
82	(52)	CHARACTER	4	cplRENG_IAT	time
82	(52)	CHARACTER	2	cplRENG_IATH	HH
84	(54)	CHARACTER	2	cplRENG_IATM	MM
86	(56)	CHARACTER	114	*	free

BIND REQUESTS INFORMATION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	16	CPLBNDOP (*)	
0	(0)	CHARACTER	8	CPLBND99K	XD99 KEY
8	(8)	BITSTRING	1	CPLBNDFLAG	
		1...		CPLBNDDEL	
		.111 1111		*	
9	(9)	CHARACTER	7	*	

Cross reference

Name	Hex Offset	Hex Value	Level
cplbnd	1AC		4
cplcdeop	1A0		4
CPLCond_F	19C		4
CPLCond_T	19A		4
CPLCond_U	19E		4
CPLCondTot	198		4
CPLCPROP	194		4
CPLCSUOP	196		4
CPLDEOP	E8		4
CPLPCOP	EA		4
CPLPROP	E6		4
CPLPSOP	D2		4
CPLPXOP	1A2		4
CPLR1OP	D4		4
CPLR2OP	D6		4
CPLSROP	EC		4
CPLSUOP	E4		4
CPLAADOP	AA		5
CPLAAOP	AA		4
CPLAATOP	B0		5
CPLADIOP	1A		4
CPLADUOP	CC		4
CPLAECOP	DD		4
CPLAEDOP	BE		5
CPLAEOP	BE		4
CPLAETOP	C4		5
CPLAJR	EE	02	5
cplAlcTime	38		2
CPLALTWS	FE		4
CPLASDOP	A0		5

Name	Hex Offset	Hex Value	Level
CPLASIDJOB	128		4
CPLASOP	A0		4
CPLASSOP	EE	20	5
CPLASTOP	A6		5
CPLASUOP	EE	04	5
cplAutInf	15F		4
CPLBDYOP	C		3
CPLBNDDEL	8	80	3
CPLBNDFLAG	8		2
CPLBNDOP	0		1
CPLBND99K	0		2
cplclncm	65	02	5
cplclnDia	14A		4
cplclnRes	149		4
cplClnStat	19		4
cplClnTyp	12A		4
CPLCOM3P	0		2
CPLCondId	1A8		4
cplcondrjob	16E	02	5
CPLCPCOOP	0	40	3
CPLCPMatc	1		2
CPLCPNDOP	0	80	3
CPLCPOCOP	4		3
CPLCPOP	4		2
CPLCPOP_cid	2		2
CPLCPOPOP	7		3
CPLCPREOP	0		1
CPLCPSWOP	0		2
cplCP15req	65	20	5
cplCritJobPred	16E	10	5
cplCritPath	178		4
cplCritPred	0	10	3
cplCrtPthX	180		4
CPLCSOCOP	4		3
CPLCSOP	4		2
CPLCSOP_cid	2		2
CPLCSOPOP	7		3
CPLCSSWOP	0		2
CPLCSTOP	D8		4
CPLCSUCOP	0		1
CPLCSUFLAG	1		2

Name	Hex Offset	Hex Value	Level
CPLCSUSTEP	1	80	3
CPLDEFLT	65	80	5
CPLDEPTOP	F8		4
CPLDIRER	F9	10	5
CPLDIRES	F9	40	5
cplDOAtime	154		5
cpldprem	16E	08	5
cpldrecov	1AE	40	5
CPLDWTO	F9	04	5
CPLDWTOP	F9	01	5
CPLDWTOS	F9	02	5
cpldyn	1AE	80	5
cplEarE	17C		4
CPLEDUOP	C8		4
cplEQQCLJBN	154		4
cplEQQCLjid	14B		4
cplerfail	153	08	5
CPLERKOP	E0		4
CPLEROOP	E2		4
CPLERROP	D9		4
cplEXdest	120		4
cplexec	65	04	5
cplexit13rc	1AE	08	5
cplExpJCL	114	01	5
cplexpjclUSED	153	40	5
cplExtInf	15D		4
CPLEXTOP	EE	40	5
CPLEYE3P	C		4
cplFakeSub	16F	02	5
cplflag05	114		4
cplflag06	153		4
cplflag07	15C		4
cplflag08	16E		4
cplflag09	16F		4
cplflag10	1AE		4
cplflag11	1B0		4
CPLFLGOP	EE		4
CPLFLGO2	EF		4
CPLFLGO3	F9		4
CPLFLG05	65		4
CPLFRMOP	66		4

Name	Hex Offset	Hex Value	Level
cplFromPLC	15C	10	5
cplFTSAnop	16E	20	5
cplftwRC	160		4
CPLGROUP3P	11		4
CPLHRCOP	FA		4
CPLHRUOP	EF	04	5
CPLIADOP	2A		5
CPLIAOP	2A		4
CPLIATOP	30		5
CPLIDOP	5E		4
cplimmerr	114	04	5
CPLISDOP	B4		5
cplisdopf2	B4		7
cplisdopl4	B6		6
CPLISOP	B4		4
CPLISTOP	BA		5
CPLJBNOP	4E		4
CPLJCLOP	64		4
cplJESdest	184		4
CPLJIDOP	56		4
CPLJIDOPN	5A		5
CPLJIDOPP	56		5
cplJLInfo	114	40	5
cplJLOdest	13C		4
cpljobcrt	116		5
cpljobpol	117		5
cplJobTblIdx	190		4
cplJRstat	115		4
CPLKEYOP	0		3
CPLKF03P	6		4
CPLKYP3P	0		4
CPLLATEP	EE	01	5
CPLLODOP	96		5
CPLLONGP	EF	01	5
CPLLOOP	96		4
CPLLOTOP	9C		5
cplluserop	134		4
CPLMCPUP	F0		4
cplmhld	65	10	5
cplmonp	153	80	5
cplMoreRun	114	80	5

Name	Hex Offset	Hex Value	Level
cplmpendp	1AE	01	5
cplmppend	1B0	80	5
CPLMPREDopno	102		5
cplMvDeadl	1AE	10	5
CPLMXLVL	102		4
CPLNDXOP	2		4
CPLNETOP	34		4
cplnoerr	16F	08	5
cplNoerrJT	1AE	04	5
cplNoerrSkip	1AE	02	5
cplNoEval	16E	01	5
cplnop	65	08	5
CPLNUMOP	62		5
CPLOC03P	2		5
CPLODDOP	8C		5
CPLODOP	8C		4
CPLODTOP	92		5
CPLOIDOP	82		5
CPLOIOP	82		4
CPLOITOP	88		5
cplOpEx03p	148		4
cplOpInfo	114	20	5
CPLOPIRET	D0		4
CPLOP03P	5		5
CPLOrigRC	1A4		4
CPLPCOOP	0	80	3
CPLPEDOP	78		5
CPLPENDP	EF	02	5
CPLPEOP	78		4
CPLPETOP	7E		5
CPLPIVLleft	8		3
CPLPIVLright	A		3
CPLPMandPend	0	08	3
CPLPMatc	1		2
CPLPMatcIVL	8		2
CPLPNDOP	0	20	3
CPLPNROP	0	40	3
CPLPOCOP	4		3
CPLPOP	4		2
CPLPOPOP	7		3
CPLPREOP	0		1

Name	Hex Offset	Hex Value	Level
CPLPRIOP	DE		4
CPLPRPOP	EF	20	5
CPLPSDOP	6E		5
CPLPSOP	6E		4
CPLPSTOP	74		5
cplPSUdone	114	10	5
cplPSUtoken	144		4
CPLPSUWAIT	65	40	5
CPLPSWOP	0		2
cplPthOc	180		5
cplPthOp	183		5
CPLPTTOP	2		2
cplRecEnd	15C	40	5
cplrecjob	153	02	5
cplRecovOn	16F	10	5
cplrecprmp	153	04	5
cplrecrun	153	01	5
cplRec24Is	15C	80	5
CPLREC3P	0		1
cplrejst	1AB		4
CPLRELOP	EF	10	5
cplRENG	0		1
cplRENG_comp	4B	80	3
cplRENG_flags	4B		2
cplRENG_IA	4C		2
cplRENG_IAD	4C		3
cplRENG_IAT	52		3
cplRENG_IATH	52		4
cplRENG_IATM	54		4
cplRENG_jobname	10		2
cplRENG_jsname	0		2
cplRENG_jsws	38		2
cplRENG_opno	48		2
cplRENG_zjob	10		3
cplRENG_zws	38		3
cplRengInf	1AA		4
CPLREROP	EE	80	5
CPLRERUT	F9	20	5
cplrescnew	2D	04	3
CPLRESTA	F9	80	5
cplRUSER	164		4

Name	Hex Offset	Hex Value	Level
cplscript	15C	20	5
CPLSCROP	0	80	3
cplsending	153	20	5
cplshadow	16F	01	5
cplshoseq	1AE	20	5
CPLSOCOP	2		3
CPLSOP	2		2
CPLSOPOP	5		3
CPLSPDL	EE	08	5
CPLSPIA	EE	10	5
cplsrAVAco	40		2
CPLSRAVOP	2D	80	3
cplsrerr	2E		2
cplsridx	34		2
CPLSRIUSOP	2D	40	3
CPLSRIUXOP	2D	20	3
cplsrkept	2D	10	3
cplsrkeptx	2D	08	3
CPLSRNOP	0		2
cplsrnum	30		2
CPLSRSOP	0		1
CPLSRTOP	2C		2
cplsrwrs	2F		2
CPLSSEQ	65	01	5
CPLSSEQOP	B4		6
CPLSSWOP	0		2
cplstch	12C		4
cplstchd	12C		5
cplstcht	130		5
cplstepdep	16F	80	5
cplStepSucc	16F	40	5
cplsuber	153	10	5
CPLSUBOP	EF	80	5
cplsubtoken	14B		5
CPLSUCOP	0		1
cplsyssfail	114	02	5
cplTaskType	1AF		4
CPLTJTOP	EF	40	5
CPLTMPOP	FC		4
cpltoken	11C		5
cpltoken_full	118		4

Name	Hex Offset	Hex Value	Level
cpltoken_pre	118		5
cplTWSjbnm	15E		4
cpltwsjob	12B		4
CPLTXTOP	36		4
cplTZOFF	18E		4
cplTZONE	16E	04	5
CPLudata	104		4
cplUnexpRC	16F	04	5
cplURGmsg	15C	01	5
cplUserSYS	114	08	5
CPLVAR03	1B4		2
cplVdest	186		4
CPLVERS3P	10		4
cplWaitSE	15C	04	5
cplWaitWQA	15C	02	5
cplWASUJ	15C	08	5
cplwlm	116		4
cplWLMclass	170		4
cplWLMmsg	16E	40	5
cplWLMprom	16E	80	5
CPLWRER	F9	08	5
CPLWSOP	5E		5
CPLXSTOP	DF		4
cplzcentric	16F	20	5
CPL2EVOP	EF	08	5
cpl3pExtInfo	0		1
cpl3pEXTname	0		2
cpl3pSEname	36		2
cpl3pSPLEX	16C		4

CPLREC3Q - Current plan operation condition record

Name : DCLCPR3Q

Function:
This segment declares a current plan operation condition record, physically located in EQQCPXDS and EQQNCPDS.

The structure of the operation record is as follows:

```
DCL
  1 CPLREC3Q
    2 common data
    2 condition dependency ( )
```

Common data segment

```
Offsets      Type     Length  Name           Description
   0    (0) STRUCTURE     *   CPLREC3Q       OPERATION CONDITION RECORD
   0    (0) CHARACTER   133   CPLCOM3Q
   0    (0) CHARACTER    12   CPLKEY3Q
   0    (0) CHARACTER     2   CPLKYP3Q       TYPE IS ALWAUS 03
   2    (2) CHARACTER     4   CPLNDX3Q       OPR VSAM KEY
   2    (2) UNSIGNED      3   CPLOC03Q       OCCUURENCE NUMBER
   5    (5) UNSIGNED      1   CPLOP03Q       OPERATION NUMBER
   6    (6) CHARACTER     2   CPLFIL13Q      ALWAYS '--' ( '6060'X )
   8    (8) SIGNED        2   CPLCOID3Q      COND ID
  10    (A) CHARACTER     2   CPLFIL23Q      SHOULD CONTAIN BLANKS
  12    (C) CHARACTER   121   CPLBDY3Q       BODY OF CONDITION RECORD
  12    (C) CHARACTER     4   CPLEYE3Q       EYE CATCHER: CP3Q
  16   (10) UNSIGNED      1   CPLVERS3Q      VERSION NUMBER
  17   (11) CHARACTER     3   CPLFIL33Q      SHOULD CONTAIN '00'X
  20   (14) SIGNED        2   CPLTYPE3Q      CONDITION TYPE:
                                             1 = ONE OF
                                             2 = AT LEAST N OF
                                             3 = ALL
  22   (16) SIGNED        2   CPL#SIMP3Q     NUMBER OF CONDITION DEPENDENCIES
  24   (18) SIGNED        2   CPLCOUNT3Q     COUNTER: - ONLY FOR TYPE 2
  26   (1A) BITSTRING     1   CPLFLAGS3Q     CONDITION FLAGS
            1...  ....        CPLEMPTY3Q     CONDITION EMPTY
            .1..  ....        CPLAURES3Q     CONDITION GENERATED BY SUCCESSOR RESOLUTION
            ..1.  ....        CPLSTCHG3Q     CONDITION STATUS CHANGE(NOT SAVED)
            ...1  ....        CPLSTEP3Q      CONTAINS STEP LEVEL CONDITION DEPENDENCIES
            ....  1111        *              FREE
  27   (1B) CHARACTER     1   *              FREE
  28   (1C) CHARACTER    24   CPLDESC3Q      DESCRIPTION
  52   (34) CHARACTER     1   CPLVALUE3Q     FINAL CONDITION STATUS:
                                             U = UNDECIDED
                                             T = TRUE
                                             F = FALSE
  53   (35) CHARACTER    16   CPLADID3Q      ADID OF THE OWNING OPERATION
  69   (45) CHARACTER    10   CPLIA3Q        OCCURRENCE IA OF THE OWNING OPERATION
  69   (45) CHARACTER     6   CPLIAD3Q       - DATE
  75   (4B) CHARACTER     4   CPLIAT3Q       - TIME
  79   (4F) CHARACTER     1   CPLXST3Q       EXTENDED STATUS (RERUN)
  80   (50) SIGNED        2   CPLREFID3Q     REF. CONDITION ID (SUCCESSOR RESOLUTION)
  82   (52) CHARACTER    51   *              FREE
 133   (85) CHARACTER     *   CPLVAR3Q
```

Condition dependency segment

The value of CPL#SIMP3Q in the common data segment describes the number of condition dependencies segments.

```
Offsets      Type     Length  Name             Description
   0    (0) STRUCTURE    82   CPLSIMP3Q(*)
   0    (0) CHARACTER     4   CPLSIMP_KEY      VSAM KEY OF CONDITIONAL PRED
   0    (0) UNSIGNED      3   CPLSIM3QOC
   3    (3) UNSIGNED      1   CPLSIM3QOP
   4    (4) CHARACTER     2   CPLSIMPTYP       CONDITION DEPENDENCY CHECK TYPE:
                                               ST = STATUS
                                               RC = RETURN CODE
   6    (6) CHARACTER     2   CPLSIMPLOG       OPERATOR: GE, GT, LE, LT, EQ, NE, RG
   8    (8) CHARACTER     4   CPLSIMPVALRC     RC VALUE
  12    (C) CHARACTER     4   CPLSIMPVALRC2    RC2 VALUE
  16   (10) CHARACTER     1   CPLSIMPVALST     ST VALUE
  17   (11) CHARACTER     1   CPLSIMPLVAL      SIMPLE COND STATUS:
                                               U = UNDECIDED
                                               T = TRUE
                                               F = FALSE
  18   (12) CHARACTER    36   CPLSIMP_EXTKEY   BLANK FOR INTERNAL PREDECESSOR
  18   (12) CHARACTER    16  CPLSIMPADI       APPLICATION ID
  34   (22) CHARACTER    10  CPLSIMPIA        APPLLICATION INPUT ARRIVAL
  34   (22) CHARACTER     6  CPLSIMPIAD       DATE
  40   (28) CHARACTER     4  CPLSIMPIAT       TIME
  44   (2C) CHARACTER    10  *                FREE
  54   (36) CHARACTER     1  *                FREE
  55   (37) CHARACTER     1  CPLSIMPDEP       I (INTERNAL) E (EXTERNAL)
  56   (38) BITSTRING     1  CPLSIMPFLAG      FLAGS
            1...  ....       CPLSIMPREM       ON: OPER REMOVED BY DP
            .1..  ....       CPLSIMSTCHG      ON: CONDITION DEP STATUS CHANGE
            ..1.  ....       CPLSIMPMISS      ON: STEP-END INFO MISSING
            ...1  ....       CPLSIMPMISSCHG   ON: MISSING EVENT SET
            ....  1...       CPLSIMPMISSNO    ON: STEP NOT MISSING BUT NOT EVALUATED FOR SUCCESSOR VIEW
                             CPLSIMMISSNOCHG  ON: MISSNO EV SET
            ....  ..11       *                FREE
  57   (39) CHARACTER     9  *                FREE
  66   (42) CHARACTER     8  CPLSIMPSTEP      PROCEDURE INVOCATION STEP NAME
  74   (4A) CHARACTER     8  CPLSIMPPSTEP     STEP NAME
```

Cross reference

```
                                             HEX         HEX
NAME                                        OFFSET      VALUE     LEVEL

____                                        ______     ________   _____
CPL#SIMP3Q                                      16                    4
CPLADID3Q                                       35                    4
CPLAURES3Q                                      1A           40       5
CPLBDY3Q                                         C                    3
CPLCOID3Q                                        8                    4
CPLCOM3Q                                         0                    2
CPLCOUNT3Q                                      18                    4
CPLDESC3Q                                       1C                    4
CPLEMPTY3Q                                      1A           80       5
CPLEYE3Q                                         C                    4
CPLFIL13Q                                        6                    4
CPLFIL23Q                                        A                    4
CPLFIL33Q                                       11                    4
CPLFLAGS3Q                                      1A                    4
CPLIAD3Q                                        45                    5
CPLIAT3Q                                        4B                    5
CPLIA3Q                                         45                    4
CPLKEY3Q                                         0                    3
CPLKYP3Q                                         0                    4
CPLNDX3Q                                         2                    4
CPLOC03Q                                         2                    5
CPLOP03Q                                         5                    5
CPLREC3Q                                         0                    1
CPLREFID3Q                                      50                    4
CPLSIMP_EXTKEY                                  12                    2
CPLSIMP_KEY                                      0                    2
CPLSIMPADI                                      12                    3
CPLSIMPDEP                                      37                    2
CPLSIMPFLAG                                     38                    2
CPLSIMPIA                                       22                    3
CPLSIMPIAD                                      22                    4
CPLSIMPIAT                                      28                    4
CPLSIMPLOG                                       6                    2
CPLSIMPLVAL                                     11                    2
CPLSIMPMISS                                     38           20       3
CPLSIMPMISSCHG                                  38           10       3
CPLSIMPMISSNO                                   38           08       3
CPLSIMPMISSNOCHG                                38           04       3
CPLSIMPPSTEP                                    4A                    2
CPLSIMPREM                                      38           80       3
CPLSIMPSTCHG                                    38           40       3
CPLSIMPSTEP                                     42                    2
CPLSIMPTYP                                       4                    2
CPLSIMPVALRC                                     8                    2
CPLSIMPVALRC2                                    C                    2
CPLSIMPVALST                                    10                    2
CPLSIMP3Q                                        0                    1
CPLSIMSTCHG                                     38           40       3
CPLSIM3QOC                                       0                    3
CPLSIM3QOP                                       3                    3
CPLSTCHG3Q                                      1A           20       5
```

CPLSTEP3Q	1A	10	5
CPLTYPE3Q	14		4
CPLVALUE3Q	34		4
CPLVAR3Q	85		2
CPLVERS3Q	10		4
CPLXST3Q	4F		4

CPLREC3R - Current plan operation user fields record

Name : DCLCPR3R

Function:
This segment declares a current plan operation user fields record, physically located in EQQCPXDS and EQQNCPDS.

The structure of the operation record is as follows:

```
DCL
  1 CPLREC3R
    2 common data
    2 user field ( )
```

Common data segment

```
Offsets       Type        Length  Name          Description
  0     (0)  STRUCTURE   *       CPLREC3R      OPR CONDITION RECORD
  0     (0)  CHARACTER   92      CPLCOM3R
  0     (0)  CHARACTER   12      CPLKEY3R
  0     (0)  CHARACTER   2       CPLKYP3R      TYPE IS ALWAYS 03
  2     (2)  CHARACTER   4       CPLNDX3R      OPR VSAM KEY
  2     (2)  UNSIGNED    3       CPLOC03R      OCC NUMBER
  5     (5)  UNSIGNED    1       CPLOP03R      OPR NUMBER
  6     (6)  CHARACTER   2       CPLFIL13R     ALWAYS '--' ( '6060'X )
  8     (8)  CHARACTER   4       CPLEYE3R      ALWAYS 'CP3R'
 12     (C)  CHARACTER   80      CPLBDY3R      BODY
 12     (C)  CHARACTER   4       CPLEYE3R1     ALWAYS 'CP3R'
 16    (10)  UNSIGNED    1       CPLVERS3R     VERSION
 17    (11)  CHARACTER   4       CPLFIL33R     ALWAYS '00'X
 20    (14)  SIGNED      2       CPLUFN3R      NUMBER OF USER FIELDS
 22    (16)  CHARACTER   1       CPLVALUE3R    EQUAL TO 'D' MEANS DELETED RECORD
 23    (17)  CHARACTER   1       *             FREE
 24    (18)  CHARACTER   16      CPLADID3R     ADID OF THE OWNING OPERATION
 40    (28)  CHARACTER   10      CPLIA3R       OCC IA  OF THE OWNING OPR
 40    (28) CHARACTER    6       CPLIAD3R      DATE
 46    (2E) CHARACTER    4       CPLIAT3R      TIME
 50    (32) CHARACTER    42      *             FREE
 92    (5C) CHARACTER    *       CPLVAR3R
```

User Field segment

The value of CPL#UFN3R in the common data segment describes the number of user fields segments.

```
Offsets       Type        Length  Name          Description
    0    (0) STRUCTURE   70      CPLBODY3R(*)
    0    (0) CHARACTER   16      CPLUFNM3R     USER FIELD NAME
   16   (10) CHARACTER   54      CPLUFV3R      USER FIELD VALUE
```

Cross reference

Name	Hex Offset	Hex Value	Level
CPLADID3R	18		4
CPLBDY3R	C		3

Name	Hex Offset	Hex Value	Level
CPLBODY3R	0		1
CPLCOM3R	0		2
CPLEYE3R	8		4
CPLEYE3R1	C		4
CPLFIL13R	6		4
CPLFIL33R	11		4
CPLIAD3R	28		5
CPLIAT3R	2E		5
CPLIA3R	28		4
CPLKEY3R	0		3
CPLKYP3R	0		4
CPLNDX3R	2		4
CPLOC03R	2		5
CPLOP03R	5		5
CPLREC3R	0		1
CPLUFNM3R	0		2
CPLUFN3R	14		4
CPLUFV3R	10		2
CPLVALUE3R	16		4
CPLVAR3R	5C		2
CPLVERS3R	10		4

CPLREC33 - Current plan system automation info occurrence record

Name : DCLCPR33

Function:
This segment declares a current plan system automation information occurrence record.

Common data segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC33	SYSTEM AUTOMATION INFO (SAI) RECORD
0	(0)	CHARACTER	12	CPL33KEY	KEY OF RECORD
0	(0)	CHARACTER	2	CPLTYP33	RECORD TYPE (ALWAYS 33)
2	(2)	UNSIGNED	3	CPLOCC33	OCCURENCE NUMBER IN CP
5	(5)	UNSIGNED	1	CPLOPR33	OPERATION NUMBER
6	(6)	CHARACTER	6	CPLKF033	CONTAINS BLANKS
12	(C)	CHARACTER	404	CPL33BDY	BODY OF SYSTEM AUTOMATION INFO RECORD
12	(C)	CHARACTER	4	CPL33EYE	EYE CATCHER (ALWAYS CP33)

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
16	(10)	UNSIGNED	1	CPL33VER	RECORD VERSION
17	(11)		1	*	RESERVED
18	(12)	CHARACTER	1	CPLFIL33	ALIGNMENT WITH CP VSAM KEY
19	(13)	CHARACTER	336	CPL33AUTINFO	SYSTEM AUTOMATION INFO SECTION
19	(13)	CHARACTER	256	CPL33COMTXT	SYSTEM AUTOMATION COMMAND TEXT
19	(13)	CHARACTER	64	CPL33COMTXT1	SA COMMAND TEXT, LINE 1
83	(53)	CHARACTER	64	CPL33COMTXT2	SA COMMAND TEXT, LINE 2
147	(93)	CHARACTER	64	CPL33COMTXT3	SA COMMAND TEXT, LINE 3
211	(D3)	CHARACTER	64	CPL33COMTXT4	SA COMMAND TEXT, LINE 4
275	(113)	CHARACTER	8	CPL33AUTOP	SA AUTOMATED FUNCTION (FOR OPERATION)
283	(11B)	CHARACTER	8	CPL33SECELEM	SA SECURITY ELEMENT
291	(123)	CHARACTER	64	CPL33CINFO	SA COMPLETION INFO
355	(163)	CHARACTER	61	*	FREE
416	(1A0)	CHARACTER	31	CPL33VAR	VAR DATA

Cross reference

Name	Hex Offset	Hex Value	Level
CPLFIL33	12		3
CPLKF033	6		3
CPLOCC33	2		3
CPLOPR33	5		3
CPLREC33	0		1
CPLTYP33	0		3
CPL33AUTINFO	13		3
CPL33AUTOP	113		4
CPL33BDY	C		2
CPL33CINFO	123		4
CPL33COMTXT	13		4
CPL33COMTXT1	13		5
CPL33COMTXT2	53		5
CPL33COMTXT3	93		5
CPL33COMTXT4	D3		5
CPL33EYE	C		3
CPL33KEY	0		2
CPL33SECELEM	11B		4
CPL33VAR	1A0		2
CPL33VER	10		3

CPLREC44 - Current plan step name change record

Name : DCLCPR44

Function:
This segment declares the step name change record.
The structure of this record is the following:

```
DCL 1 CPLREC44
        2 common data
        2 run info
        2 step entries
```

Common data segment

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	52	CPLREC44	STEP NAME CHANGE TABLE
0	(0)	STRUCTURE	52	CPLFIX44	
0	(0)	CHARACTER	16	CPLKEY44	
0	(0)	CHARACTER	2	CPLTYP44	ALWAYS 44
2	(2)	CHARACTER	4	CPLKOP44	KEY TO RECORD 44
2	(2)	CHARACTER	3	CPLOCC44	OCCURRENCE NUMBER
5	(5)	UNSIGNED	3	CPLOPR44	OPERATION NUMBER
6	(6)	SIGNED	2	CPLSEQ44	ALWAYS 0
8	(8)	CHARACTER	8	CPLKF44P	RESERVED
16	(10)	STRUCTURE	36	CPLBDY44	BODY OF RECORD 44
16	(10)	CHARACTER	4	CPL44EYE	EYE CATCHER
20	(14)	CHARACTER	1	CPL44VER	VERSION NUMBER
21	(15)	CHARACTER	1	*	FREE
22	(16)	CHARACTER	16	CPL44ADID	APPLICATION ID
38	(26)	CHARACTER	10	CPL44IA	INPUT ARRIVAL TIME
38	(26)	CHARACTER	6	CPL44IAD	IA DATE
44	(2C)	CHARACTER	4	CPL14IAT	IA TIME
48	(30)	SIGNED	4	CPL44RUN	RUN NUMBER
52	(34)	CHARACTER	*	CPLVAR44	VAR DATA

Run information segment

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	4	CPLENT44	RUN INFO
0	(0)	SIGNED	4	CPLNUM44	STEP NUMBER
4	(4)	CHARACTER	*	CPLVARE44	VAR DATA

Step entry segment

The value of CPLNUM44 in the run information segment describes the number of step entry segments.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	4	CPLSTE44	STEP ENTRY
0	(0)	SIGNED	1	LAST44	LAST RUN FLAG Y/N
1	(1)	CHARACTER	3	*	FREE
4	(4)	CHARACTER	8	STEP44	STEP NAME
12	(C)	CHARACTER	8	PROCSTEP44	PROCEDURE STEP NAME
20	(14)	CHARACTER	8	NEWNAME44	NEW STEP NAME

Cross reference

Name	Hex Offset	Hex Value	Level
CPL14IAT	2C		5
CPL44ADID	16		4
CPL44EYE	10		4
CPL44IA	26		4
CPL44IAD	26		5
CPL44RUN	30		4
CPL44VER	14		4
CPLBDY44	10		3
CPLENT44	0		1
CPLFIX44	0		2
CPLKEY44	0		3
CPLKF44P	8		4
CPLKOP44	2		4
CPLNUM44	0		2
CPLOCC44	2		5
CPLOPR44	5		5
CPLREC44	0		1
CPLSEQ44	6	80	4
CPLSTE44	0		1
CPLTYP44	0		4
CPLVAR44	34		2
CPLVARE44	4		2
LAST44	0		2
NEWNAME44	14		2
PROCSTEP44	C		2
STEP44	4		2

CPR55 - AD ID INDEX RECORD

Name: DCLCPR55

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLREC55	AD ID INDEX RECORD
0	(0)	CHARACTER	19	CPLKEY55	
0	(0)	CHARACTER	2	*	TYPE IS ALWAYS 55
2	(2)	CHARACTER	16	CPLAID55	APPLICATION
18	(12)	UNSIGNED	1	CPLSEQ55	CP55 SEQUENCE NUMBER 0 FOR SPECIAL CP55 REC
19	(13)	CHARACTER	*	CPLBDY55	BODY OF AD ID INDEX RECORD
19	(13)	CHARACTER	4	CPLEYE55	EYECATCHER
23	(17)	UNSIGNED	1	CPLVERS55	VERSION NUMBER
24	(18)	SIGNED	2	CPLOC55	NUMBER OF OCCURRENCES FOR INDEX RECORD (CPLSEQ55X=0) CONTAINS THE NUMBER OF CP55 RECORDS IN THE CHAIN
26	(1A)	CHARACTER	6	*	FREE
32	(20)	CHARACTER	*	CPLVAR55	VARIABLE PART OF RECORD

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLOCS55	OCCURRENCES DATA
0	(0)	CHARACTER	32	CPLOCC55 (*)	OCCURRENCE DATA
0	(0)	CHARACTER	10	CPLIA55	INPUT ARRIVAL
0	(0)	CHARACTER	6	CPLIAD55	DATE
6	(6)	CHARACTER	4	CPLIAT55	TIME
10	(A)	CHARACTER	10	CPLADL55	ACTUAL COMPLETIONDELETION
10	(A)	CHARACTER	6	CPLDLD55	DATE BLANK
16	(10)	CHARACTER	4	CPLDLT55	TIME 0000
20	(14)	UNSIGNED	3	CPLOCX55	OCCURRENCE NUMBER
23	(17)	SIGNED	2	CPLNET55	NETWORK ID
25	(19)	CHARACTER	1	CPLSTA55	OCCURRENCE STATUS U = UNDECIDED W = WAITING S = STARTED E = ENDED IN ERROR C = COMPLETED D = DELETED P = PENDING PREDECESSOR M = Mandatory pending
26	(1A)	1...		CPLCLT55	1 MEANS ALREADY MARKED.. COMPLETE ON LTP
		.1..		CPLMCP55	1 MEANS ADDED BY MCP
		..1.		CPLIAM55	1 MEANS IA MODIFIED BY MCP
		...1		CPLCFR55	1 MEANS COPIED FOR REPORT
		.... 1...		CPLMPEND55	1 MEANS ORIG MPEND
		.... .111		*	FREE
27	(1B)	CHARACTER	1	CPLOCA55	D MEANS ADDED BY MCP- DIALOG E MEANS ADDED BY ETT P MEANS ADDED BY PIF A MEANS ADDED BY AR BLANK MEANS DP
28	(1C)	CHARACTER	1	CPLOCDB2	Y = ADDED TO DB2
29	(1D)	CHARACTER	1	CPLCP16AD	Y = ADDED TO CP16
30	(1E)	CHARACTER	2	*	FREE

ENTRIES OF INDEX RECORD CPLSEQ550

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	CPLIDX55	INDEX RECORD ENTRY
0	(0)	CHARACTER	32	CPLOCCX55 (*)	SINGLE CP55 RECORDS IX
0	(0)	CHARACTER	10	CPLFIA55	FIRST INPUT ARRIVAL
0	(0)	CHARACTER	6	CPLFIAD55	DATE
6	(6)	CHARACTER	4	CPLFIAT55	TIME
10	(A)	CHARACTER	10	CPLLIA55	LAST INPUT ARRIVAL
10	(A)	CHARACTER	6	CPLLIAD55	DATE
16	(10)	CHARACTER	4	CPLLIAT55	TIME
20	(14)	SIGNED	2	CPLOCN55	OCCURRENCE NUMBER
22	(16)	CHARACTER	10	*	RESERVED

Cross reference

Name	Hex Offset	Hex Value	Level
CPLOC55	18		3
CPLADL55	A		3
CPLAID55	2		3
CPLBDY55	13		2
CPLCFR55	1A	10	3
CPLCLT55	1A	80	3
CPLCP16AD	1D		3
CPLDLD55	A		4
CPLDLT55	10		4
CPLEYE55	13		3
CPLFIAD55	0		4
CPLFIAT55	6		4
CPLFIA55	0		3
CPLIAD55	0		4
CPLIAM55	1A	20	3
CPLIAT55	6		4
CPLIA55	0		3
CPLIDX55	0		1
CPLKEY55	0		2
CPLLIAD55	A		4
CPLLIAT55	10		4
CPLLIA55	A		3
CPLMCP55	1A	40	3
CPLMPEND55	1A	08	3
CPLNET55	17		3
CPLOCA55	1B		3
CPLOCCX55	0		2

Name	Hex Offset	Hex Value	Level
CPLOCC55	0		2
CPLOCDB2	1C		3
CPLOCN55	14		3
CPLOCS55	0		1
CPLOCX55	14		3
CPLREC55	0		1
CPLSEQ55	12		3
CPLSTA55	19		3
CPLVAR55	20		3
CPLVERS55	17		3

CPLREC66 - Critical job index record

Name : DCLCPR66

Function:
This segment declares a critical job index record, physically located in EQQCPxDS and EQQNCPDS. The structure of this record is the following:
DCL 1 CPLREC66
2 common data
2 critical job entries ( )

Common data segment

```
Offsets
Dec   Hex Type         Len   Name       Description
 0    (0) STRUCTURE     40  CPLREC66   CRITICAL JOBS INDEX RECORD
 0    (0) CHARACTER     19  CPLKEY66
 0    (0) CHARACTER    2  *              TYPE IS ALWAYS 66
 2    (2) CHARACTER      2  *              BLANK
 4    (4) SIGNED          4  CPLSEQ66       SEQUENCE NUMBER
 8    (8) CHARACTER     11  CPLKF66        FILLER, CONTAINS BLANKS
19   (13) CHARACTER     21  CPLBDY66       BODY OF CRIT JOBS INDEX REC
19   (13) CHARACTER      4  CPLEYE66       EYECATCHER
23   (17) CHARACTER      1  CPLVERS66      VERSION
24   (18) CHARACTER      4  *              RESERVED
28   (1C) SIGNED         4  CPLCRITNUM     NUMBER OF CRIT JOBS IN REC
32   (20) CHARACTER      8  *              RESERVED
40   (28) CHARACTER         CPLVAR66       VARIABLE PART OF RECORD
```

Critical job entry index segment

```
Offsets
Dec   Hex Type      Len   Name            Description
 0    (0) STRUCTURE     *    CPLCRIT66       CRIT JOBS SECTION
 0    (0) CHARACTER    28    CPLCRITJOB66(*) CRIT JOB ENTRY
 0    (0) UNSIGNED      3    CPLOCX66        OCCUR NUMBER
 3    (3) UNSIGNED      1    CPLOPR66        OPERATION NUMBER
 4    (4) SIGNED        4    CPLJTIDX        JOB TABLE INDEX
 8    (8) SIGNED        4    CPLCRPATHREC#   CNT FOR CRIT PATH RECALC
12    (C) CHARACTER     3    *               FREE
15    (F) CHARACTER     1    CPLCRPATHTRG    CRIT PATH RECALC TRIGGER
16   (10) CHARACTER     2    *               CRIT PATH RECALC TRIGGER
18   (12) CHARACTER    10    CPLCRPATHREC   CRIT PATH RECALC
18   (12) CHARACTER     6    CPLCRPATHRECD    CRIT PATH RECALC DATE
24   (18) CHARACTER     4    CPLCRPATHRECT    CRIT PATH RECALC TIME
```

Cross reference

NAME	HEX OFFSET	HEX VALUE	LEVEL
CPLBDY66	13		2
CPLCRITJOB66	0		2
CPLCRITNUM	1C		3
CPLCRIT66	0		1
CPLCRPATHREC	12		3
CPLCRPATHREC#	8		3
CPLCRPATHRECD	12		4
CPLCRPATHRECT	18		4
CPLCRPATHTRG	F		3
CPLEYE66	13		3
CPLJTIDX	4		3
CPLKEY66	0		2
CPLKF66	8		3
CPLOCX66	0		3
CPLOPR66	3		3
CPLREC66	0		1
CPLSEQ66	4		3
CPLVAR66	28		3
CPLVERS66	17		3

CPLREC99 - Current plan occurrence index record

Name : DCLCPLRECR99

Function:
Describes layout of current plan occurrence index record.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	43	CPLREC99	STAND-ALONE CLEAN UP
0	(0)	CHARACTER	19	CPL99KEY	KEY OF RECORD
0	(0)	CHARACTER	2	CPL99TYP	RECORD TYPE (99)
2	(2)	CHARACTER	8	CPL99TOK	OCCURRENCE TOKEN=TOD
10	(A)	CHARACTER	9	*	FREE
19	(13)	CHARACTER	24	CPL99BDY	
19	(13)	CHARACTER	4	CPL99EYE	EYE CATCHER=CP99
23	(17)	CHARACTER	1	CPL99VER	VERSION
24	(18)	UNSIGNED	3	CPL99OCC	OCCURRENCE NUMBER IN CP
27	(1B)	CHARACTER	16	*	FREE

Cross reference

Name	Hex Offset	Hex Value	Level
CPLREC99	0		1
CPL99BDY	13		2
CPL99EYE	13		3
CPL99KEY	0		2
CPL99OCC	18		3
CPL99TOK	2		3
CPL99TYP	0		3

Name	Hex Offset	Hex Value	Level
CPL99VER	17		3

CSUM - Summary of completed/deleted applications report in DP

Name : DCLCSUM

Function:
Describes layout of summary of completed/ deleted occurrence records. There is one record per Priority and day. The records have the standard recsize and keysize of dpin records.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	158	CSUAREA	PARALLEL SERV|R1|R2 REC
0	(0)	CHARACTER	114	CSUKEY	KEY
0	(0)	SIGNED	2	CSUTYPE	ALWAYS TYPE 4
2	(2)	CHARACTER	34	*	ADJUST KEY
36	(24)	CHARACTER	6	CSUDATE	DATE IN ISO FORMAT
42	(2A)	SIGNED	2	CSURPRI	REVERSED PRIORITY PRI 9 -> 10 - 9 = 1 PRI 8 -> 10 - 8 = 2 ETC. TOTAL -> 10 - 0 = 10
44	(2C)	CHARACTER	70	*	TO ADD UP THE KEY
114	(72)	CHARACTER	44	CSUDATA	DATA PART OF CSU RECORD
114	(72)	CHARACTER	1	CSUPRI	OCCURENCE PRIORITY
115	(73)	CHARACTER	6	CSUCOMP	NUMBER OF COMPLETED OCC. CSURPRI=10 COMPLET+DELETED
121	(79)	CHARACTER	5	CSURERUN	NUMBER OF RERUN OCC.
126	(7E)	CHARACTER	5	CSUDELET	NUMBER OF DELETED OCC.
131	(83)	CHARACTER	5	CSULATEI	LATE ARRIVAL OCC.
136	(88)	CHARACTER	4	CSUI	AVERAGE LATE ARRIVAL
136	(88)	CHARACTER	2	CSUIHH	(HH)
138	(8A)	CHARACTER	2	CSUIMM	(MM)
140	(8C)	CHARACTER	5	CSULATEO	LATE COMPLETION
145	(91)	CHARACTER	4	CSUO	AVERAGE LATE COMPLETION
145	(91)	CHARACTER	2	CSUOHH	(HH)
147	(93)	CHARACTER	2	CSUOMM	(MM)
149	(95)	CHARACTER	5	CSUEARLO	EARLY COMPLETION
154	(9A)	CHARACTER	4	CSUE	AVERAGE EARLY COMPLETION
154	(9A)	CHARACTER	2	CSUEHH	(HH)
156	(9C)	CHARACTER	2	CSUEMM	(MM)

Cross reference

Name	Hex Offset	Hex Value	Level
CSUAREA	0		1
CSUCOMP	73		3
CSUDATA	72		2
CSUDATE	24		3
CSUDELET	7E		3
CSUE	9A		3

Name	Hex Offset	Hex Value	Level
CSUEARLO	95		3
CSUEHH	9A		4
CSUEMM	9C		4
CSUI	88		3
CSUIHH	88		4
CSUIMM	8A		4
CSUKEY	0		2
CSULATEI	83		3
CSULATEO	8C		3
CSUO	91		3
CSUOHH	91		4
CSUOMM	93		4
CSUPRI	72		3
CSURERUN	79		3
CSURPRI	2A		3
CSUTYPE	0		3

DAIOP - Daily operating plan report record in DP

Name : DCLDAIOP

Function:
Describes layout of daily_op records representing report data on operations (and occurrences) in the New current plan. There is one daily op record for each operation and additionally one for each Predecessor after the 1st predecessor, e.g. operations with 1,2,3,4 preds give 1,1,2,3 daiop records. The records have the standard recsize and keysize of dpin records.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	264	DAIAREA	DAILY_OP RECORD
0	(0)	CHARACTER	115	DAIKEY	KEY
0	(0)	SIGNED	2	DAITYPE	TYPE IS ALWAYS 2
2	(2)	CHARACTER	2	*	TO ADJUST TO ADIDOFF
4	(4)	CHARACTER	34	*	TO ADJUST TO ADIDOFF
38	(26)	CHARACTER	26	DAIOCCID	OCC IDENTIFICATION
38	(26)	CHARACTER	16	DAIADID	ADID OF OCC OF OPERATION
54	(36)	CHARACTER	10	DAIOCIA	OCCURRENCE IA® (AFTER MCP)
54	(36)	CHARACTER	6	DAIOCIAD	DAY (YYMMDD)
60	(3C)	CHARACTER	4	DAIOCIAT	TIME (HHMM)
60	(3C)	CHARACTER	2	DAIOCIAH	TIME (HH)
62	(3E)	CHARACTER	2	DAIOCIAM	TIME (MM)
64	(40)	CHARACTER	10	DAIOPRS	OPERATION PLANNED START
64	(40)	CHARACTER	6	DAIOPRSD	DAY (YYMMDD)
70	(46)	CHARACTER	4	DAIOPRST	TIME (HHMM)
70	(46)	CHARACTER	2	DAIOPRSH	TIME (HH)
72	(48)	CHARACTER	2	DAIOPRSM	TIME (MM)
74	(4A)	CHARACTER	1	DAIOPPRI	10 - OPERATION PRIORITY
75	(4B)	CHARACTER	7	DAIOPID	OPERATION ID

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
75	(4B)	CHARACTER	4	DAIWSID	WORKSTATION ID OF OPER.
79	(4F)	CHARACTER	3	DAIOPNO	OPERATION NUMBER
82	(52)	CHARACTER	33	DAIPRE	OPERATION PRED¦OR BLANK
82	(52)	CHARACTER	16	DAIPREOC	PREDECESSOR ADID
98	(62)	CHARACTER	10	DAIPREIA	PREDECESSOR IA ¦BLANK
98	(62)	CHARACTER	6	DAIPREID	YYMMDD ¦BLANK
104	(68)	CHARACTER	2	DAIPREIH	HH ¦BLANK
106	(6A)	CHARACTER	2	DAIPREIM	MM ¦BLANK
108	(6C)	CHARACTER	4	DAIPREWS	PREDECESSOR WSID ¦BLANK
112	(70)	CHARACTER	3	DAIPREOP	PREDECESSOR OP NO ¦BLANK
115	(73)	CHARACTER	149	DAIDATA	DATA PART OF DAILY_OP RCD
115	(73)	CHARACTER	10	DAIADIA	OCC ORIGINAL IA
115	(73)	CHARACTER	6	DAIADIAD	DAY (YYMMDD)
121	(79)	CHARACTER	4	DAIADIAT	TIME (HHMM)
121	(79)	CHARACTER	2	DAIADIAH	TIME (HH)
123	(7B)	CHARACTER	2	DAIADIAM	TIME (MM)
125	(7D)	CHARACTER	10	DAIOCDL	OCCURRENCE DEADLINE
125	(7D)	CHARACTER	6	DAIOCDLD	DAY (YYMMDD)
131	(83)	CHARACTER	4	DAIOCDLT	TIME (HHMM)
131	(83)	CHARACTER	2	DAIOCDLH	TIME (HH)
133	(85)	CHARACTER	2	DAIOCDLM	TIME (MM)
135	(87)	CHARACTER	1	DAIOCPRI	OCCURRENCE PRIORITY
136	(88)	CHARACTER	1	DAIOCSTA	OCC STATUS S¦A¦W¦U
137	(89)	CHARACTER	1	DAIOCRER	Y=OCC RERUN, ELSE N
138	(8A)	CHARACTER	16	DAIOCOWI	APPLICATION OWNER
154	(9A)	CHARACTER	24	DAIOCTXT	OCCURRENCE TEXT
178	(B2)	CHARACTER	8	DAIOPJOB	OPERATION JOBNAME
186	(BA)	CHARACTER	24	DAIOPTXT	OPERATION TEXT
210	(D2)	CHARACTER	2	DAIOPSPR	OPERATION SPECIAL RESOURCE
212	(D4)	CHARACTER	4	DAIOPDUR	OPERATION DURATION HHMM REMAINING
216	(D8)	CHARACTER	10	DAIOPDL	OPERATION SPEC DEADLINE
216	(D8)	CHARACTER	6	DAIOPDLD	DAY (YYMMDD) ¦BLANK
222	(DE)	CHARACTER	4	DAIOPDLT	TIME (HHMM) ¦BLANK
222	(DE)	CHARACTER	2	DAIOPDLH	TIME (HH)
224	(E0)	CHARACTER	2	DAIOPDLM	TIME (MM)
226	(E2)	CHARACTER	1	DAIOPST	OPERATION STATUS
227	(E3)	CHARACTER	1	DAIOPRER	Y=OPR PLANNED TO RERUN
228	(E4)	CHARACTER	1	DAIOPMON	Y=OPERATION IS MONITORED
229	(E5)	CHARACTER	35	*	TO ADD UP TO STD REC SIZE

Cross reference

Name	Hex Offset	Hex Value	Level
DAIADIA	73		3
DAIADIAD	73		4
DAIADIAH	79		5
DAIADIAM	7B		5
DAIADIAT	79		4
DAIADID	26		4

Name	Hex Offset	Hex Value	Level
DAIAREA	0		1
DAIDATA	73		2
DAIKEY	0		2
DAIOCCID	26		3
DAIOCDL	7D		3
DAIOCDLD	7D		4
DAIOCDLH	83		5
DAIOCDLM	85		5
DAIOCDLT	83		4
DAIOCIA	36		4
DAIOCIAD	36		5
DAIOCIAH	3C		6
DAIOCIAM	3E		6
DAIOCIAT	3C		5
DAIOCOWI	8A		3
DAIOCPRI	87		3
DAIOCRER	89		3
DAIOCSTA	88		3
DAIOCTXT	9A		3
DAIOPDL	D8		3
DAIOPDLD	D8		4
DAIOPDLH	DE		5
DAIOPDLM	E0		5
DAIOPDLT	DE		4
DAIOPDUR	D4		3
DAIOPID	4B		3
DAIOPJOB	B2		3
DAIOPMON	E4		3
DAIOPNO	4F		4
DAIOPPRI	4A		3
DAIOPRER	E3		3
DAIOPRS	40		3
DAIOPRSD	40		4
DAIOPRSH	46		5
DAIOPRSM	48		5
DAIOPRST	46		4
DAIOPSPR	D2		3
DAIOPST	E2		3
DAIOPTXT	BA		3
DAIPRE	52		3
DAIPREIA	62		4

Name	Hex Offset	Hex Value	Level
DAIPREID	62		5
DAIPREIH	68		5
DAIPREIM	6A		5
DAIPREOC	52		4
DAIPREOP	70		4
DAIPREWS	6C		4
DAITYPE	0		3
DAIWSID	4B		4

DOA - Dynamic operation area

Name : DCLDOA

Function:
This segment declares the dynamic operation area. A DOA is either on the DOA free chain or it is chained to a WQA. A DOA chained to a WQA represents an operation in progress at the workstation defined by that WQA. The information in the DOA is used to build the ready list row for an operation in a workstation ready list display and by the workstation analyzer in order to find the next operation to be scheduled.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	248	doa	Dynamic Operation Area
0	(0)	CHARACTER	4	doadesc	block descriptor = 'DOA '
4	(4)	CHARACTER	2	doaver	block mapping version
6	(6)	BITSTRING	2	doaflags	doa flags
		1...		doaclate	1 cancel oper if late
		.1..		doaurg	1 oper is urgent
		..1.		doastamr	1 status set manually
		...1		doatjt	1 time job operation
		.... 1...		doasfail	1 submitrelease failed
		.... .1..		doaarset	1 oper was auto reset
		.... ..1.		doaevent	1 a jt event was found
		1		doastart	1 job has started exec
7	(7)	1...		doajsub	1 auto submit this job
		.1..		doajhrc	1 job level high rc def
		..1.		doadurms	1 max dur exceeded
		...1		doaorer	1 oper is rerouteable
		.... 1...		doaores	1 oper is restartable
		.... .1..		doawrer	1 oper was rerouted WLRA
		.... ..1.		doarmsg	1 rer faild msg issued
		1		doanhld	1 shouldn't be released
8	(8)	CHARACTER	1	doastate	operation status
9	(9)	CHARACTER	1	doaclass	job or sysout class
10	(A)	CHARACTER	1	doaoist	opi status (yn0)
11	(B)	BITSTRING	1	doaflag2	flag byte TNSC
		1...		doadwto	deadline WTO wanted TNSA
		.1..		doadwtos	deadline WTO sent TNSA
		..1.		doadwtop	deadline WTO processed
		...1		doaremov	set oper to C or E DCRA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.... 1...		doamhld	on=manually held op HOPA
		.... .1..		doanop	on=nop op HOPA
		.... ..1.		doaexec	on=execute operation HOPA
		1		doaclncm	on=clean up comm issued
12	(C)	CHARACTER	4	doacpptr	index to operation record
12	(C)	UNSIGNED	3	doaoccx	occurrence index
15	(F)	UNSIGNED	1	doaopno	operation number
16	(10)	CHARACTER	8	doajobn	jobname (for cpu oper)
24	(18)	ADDRESS	4	doanxt	ptr to next doa
28	(1C)	ADDRESS	4	doapre	ptr to previous doa
32	(20)	ADDRESS	4	doanxttj	next time job in tjt
36	(24)	ADDRESS	4	doapretj	previous time job
40	(28)	ADDRESS	4	doawqap	addr of wqa for this doa
44	(2C)	ADDRESS	4	doadrap	addr of DRA for this doa
48	(30)	ADDRESS	4	doaaws_p	alternate WS address WLRA
52	(34)	UNSIGNED	2	doasseq	submit sequence number
54	(36)	1...		doaNoerrJT	NOERR match JT reapply DEWA
		.1..		doaNoerrSkip	NOERR skip A3S evts 9TCDEWA
54	(36)	BITSTRING	1	*	reserved 9TCDEWC
56	(38)	SIGNED	4	doadlmin	deadl. this op. in mins
60	(3C)	CHARACTER	8	doaform	form number
68	(44)	UNSIGNED	1	doascrrt	CentrScript send retry
69	(45)	CHARACTER	1	doarejst	remote eng job bind status
70	(46)	SIGNED	2	doahirc	highest non-error retcode
72	(48)	CHARACTER	8	doatime	last change (tod format)
80	(50)	SIGNED	2	doanet	network id for operation
82	(52)	SIGNED	2	doapserv	num parallel servers reqd
84	(54)	SIGNED	2	doar1	amount of r1 required
86	(56)	SIGNED	2	doar2	amount of r2 required
88	(58)	SIGNED	2	doaTZoff	TZONE returned by exit CRPC
90	(5A)	SIGNED	2	doapri	priority
92	(5C)	CHARACTER	10	doast	submit date time
102	(66)	CHARACTER	10	doalo	latest start for this op d88c
102	(66)	CHARACTER	6	doalod	date d88a
108	(6C)	SIGNED	4	doalot	time d88a
112	(70)	CHARACTER	10	doaia	input arrival for this occ
122	(7A)	CHARACTER	10	doaps	planned start f this op d88c
122	(7A)	CHARACTER	6	doapsd	date d88a
128	(80)	SIGNED	4	doapst	time d88a
132	(84)	SIGNED	4	doarelft	time 1st rlse 1.048 s
136	(88)	UNSIGNED	1	doarel	number of release done
137	(89)	CHARACTER	4	doaaltws	alternate WS WLRA
141	(8D)	CHARACTER	1	doaClnfla	Clean up flag D90C
		1...		doaMoreRun	0 = First run D90C 1 = At least 1 run D90A set by AB3P event D90A
		.1..		doaJLInfo	1 = JLInfo requested D90A 0 = JLInfo to be req. D90A set by JL info requestor D90A reset when resub () D90A
		..1.		doaOpInfo	0 = last OPerInfo not D90A available D90A 1 = last OPerInfo D90A available
		...1		doaPSUdone	0 = PSU not compl. D90A 1 = PSU completed D90A
		.... 1...		doaUserSYS	1 = need user sysout D90A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.... .1..		doaimmerr	1 = imm from error st D90A (set by A3P if error D90A status) D90A
		.... ..1.		doaCLmsg	1 = message issued D90C
		1		doaPSUwait	1 = wait for PSU done D90C set by dialog when apply logic needed but cleanup = none
142	(8E)	CHARACTER	1	doaClnStat	Clean Up status D90C see dclrccon D90A blank = none D90A S = started D90A C = completed D90A E = ended in error D90A
143	(8F)	BITSTRING	1	doaflag3	flag byte HOPA
		1...		doacmd_fail	op command fail flag HOPA
		.1..		doaresub	resend jcl for oper
		..1.		doaosi	osi operation
		...1		doasyssfail	jes sub fail
		.... 1...		doaevher	error in event handler
		.... .1..		doasrkep	keep on error
		.... ..1.		doasenq	second DLF enq done
		1		doaexpJCL	use expanded JCL D90C
144	(90)	CHARACTER	1	doaJRstat	joblog retrieval status ' ' = no action yet 'R' = requested Opinfo 'W' = waiting Opinfo 'O' = Opinfo available 'I' = initiated 'S' = started (j0-evt) 'C' = complete(j2-evt) 'E' = error (j2-evt)
145	(91)	BITSTRING	1	doaflag4	D63A
		1...		doaWLMr	WLM Already Request D63A
		.1..		doacscr	on=has centralized script
		..1.		doasending	on=S-sending exstat
		...1		doasuber	on=E-OSUF exstat
		.... 1...		doaerfail	on=E-FAIL exstat
		.... .1..		doarecprmp	on=E-prmpt exstat
		.... ..1.		doarecjob	on=E rec job running
		1		doarecrun	on=E(R,S,C) not restar
146	(92)	CHARACTER	2	doawlm	WLM fields D63A
146	(92)	CHARACTER	1	doajcrt	WLM critical indicat. D63A
147	(93)	CHARACTER	1	doajpol	WLM policy D63A
148	(94)	CHARACTER	2	*	reserved D63CLUC
150	(96)	CHARACTER	10	doadl	deadline for this opr TNSA
160	(A0)	CHARACTER	8	doatoken_full	operation full token
160	(A0)	CHARACTER	4	doatoken_pre	prefix
164	(A4)	UNSIGNED	4	doatoken	opctoken
168	(A8)	SIGNED	4	doamaxd	max duration in mins
172	(AC)	UNSIGNED	4	doadur	duration in 100th of sec
176	(B0)	ADDRESS	4	doaPSUjcl	pointer to JCL D90A

2 doaPSUtoken char4 PSU request token $BLBD

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
180	(B4)	ADDRESS	4	doaOldJCL	pointer to old JCL BLBA
184	(B8)	CHARACTER	1	doaClnTyp	Clean up Type= D90C A = automatic, D90A I = immediate, D90A M = manual, D90A N = none D90A
185	(B9)	BITSTRING	1	doaflag5	Bit battery

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		1...		doaRecEnd	op completed by recovery
		.1..		doaexpjclUSED	doaexpjcl contains BGDA really expanded jcl BGDA
		..1.		doafromPLC	job from PLC path CCUA
		...1		doaWASUJ	job from WASUJ path CCUA
		.... 1...		doawaiTSE	job wait SE CCUA
		.... .1..		doawaitWQA	wait delay
		.... ..1.		doaURGmsg	on = URG msg sent
		1		doaFTSAnop	on = nopped on standard agent hosted by master
186	(BA)	BITSTRING	1	doaflag6	reserved 32CSJC22c
		1...		doaWLMprom	on = WLM promotion done
		.1..		doaWLMmsg	on = WLM fail msg issued
		..1.		doaRdSent	reserved 34a
		...1		doanoerr	CPWA
		.... 1...		DoaCond	On = COND exists
		.... .1..		doaTZONE	on = TZONE applied CRPA
		.... ..1.		doaNoEval	On = do not eval cond succ
		1		doastepsucc	mapped on CPLSTEPSUCC
187	(BB)	BITSTRING	1	doaflag7	
		1...		doashadow	on = job is a shadow job
		.1..		doacfbind	on = complete if bind fail
		..1.		doae113	e113 issued CZ1A
		...1		doaUnexpRC	ON = Unexpected RC
		.... 1...		doadrecov	on=recovery in progress
		.... .1..		doashoseq	on=shadow in oseq
		.... ..1.		doaexit13rc	on=rc 12 from exit13 DGWA
		1		*	62CDGWC
188	(BC)	ADDRESS	4	doaseinfop	dseinfo ptr CCUA
192	(C0)	SIGNED	4	doaCRITpath	Critical path counter
196	(C4)	CHARACTER	8	doaWLMclass	WLM service class
204	(CC)	CHARACTER	4	doaCrtPthX	Critical Path Origin Key
204	(CC)	UNSIGNED	3	doaPthOc	Occ. Number
207	(CF)	UNSIGNED	1	doaPthOp	Operation Number
208	(D0)	ADDRESS	4	doasainfop	dsainfo ptr CHWA
212	(D4)	CHARACTER	8	doaVdest	last used submit dest COBA
220	(DC)	SIGNED	4	doaMaxdSec	max duration in sec CHWACVIA
224	(E0)	CHARACTER	4	doaErrCode	error code (ERRRES)
228	(E4)	CHARACTER	8	doasubtoken	zCentric submission token
236	(EC)	ADDRESS	4	doaSviewP	Sview ptr
240	(F0)	CHARACTER	1	doaTaskType	reserved
241	(F1)	CHARACTER	3	*	reserved
244	(F4)	CHARACTER	4	*	reserved
248	(F8)	CHARACTER		doaend	end of doa

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	4	*	layout of time in doa
0	(0)	CHARACTER	2	doahour	hour
2	(2)	CHARACTER	2	doaminut	minute

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	6	*	layout of date in doa
0	(0)	CHARACTER	2	doayear	year
2	(2)	CHARACTER	2	doamonth	month
4	(4)	CHARACTER	2	doaday	day

SE NAME variable part $CCUA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	30	dseinfo	CCUA
0	(0)	CHARACTER	4	dseEYE	DSEI CCUA
4	(4)	CHARACTER	2	dseVer	01 CCUA
6	(6)	CHARACTER	2	*	free CCUA
8	(8)	ADDRESS	4	dsenext	CCUA
12	(C)	CHARACTER	16	dseSEname	Scheduling Environment CCUA Name CCUA
28	(1C)	SIGNED	2	dseSysplexId	Sysplex Id CCUA

$CHWA
SAI System Automation Info variable part $CHWA
$CHWA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	348	dsainfo	CHWA
0	(0)	CHARACTER	4	dsaEYE	DSAI CHWA
4	(4)	CHARACTER	2	dsaVer	01 CHWA
6	(6)	CHARACTER	2	*	free CHWA
8	(8)	ADDRESS	4	dsanext	CHWA
12	(C)	CHARACTER	336	dsaAutoInfo	System Automation Info CHWA
12	(C)	CHARACTER	256	dsaCommText	SA op command text CHWA
12	(C)	CHARACTER	64	dsaCommText1	op com line 1 CHWA
76	(4C)	CHARACTER	64	dsaCommText2	op com line 2 CHWA
140	(8C)	CHARACTER	64	dsaCommText3	op com line 3 CHWA
204	(CC)	CHARACTER	64	dsaCommText4	op com line 4 CHWA
268	(10C)	CHARACTER	8	dsaAutoOper	SA automated operator CHWA
276	(114)	CHARACTER	8	dsaSecElem	SA security element CHWA
284	(11C)	CHARACTER	64	dsaCompInfo	SA completion info CHWA

Cross reference

Name	Hex Offset	Hex Value	Level
doa	0		1
doarel	88		2
doaaltws	89		2
doaarset	6	04	3
doaaws_p	30		2
doacfbind	BB	40	3

Name	Hex Offset	Hex Value	Level
doaclass	9		2
doaclate	6	80	3
doaCLmsg	8D	02	3
doaclncm	B	01	3
doaClnfla	8D		2
doaClnStat	8E		2
doaClnTyp	B8		2
doacmd_fail	8F	80	3
DoaCond	BA	08	3
doacpptr	C		2
doaCRITpath	C0		2
doaCrtPthX	CC		2
doacscr	91	40	3
doaday	4		2
doadesc	0		2
doadl	96		2
doadlmin	38		2
doadrap	2C		2
doadrecov	BB	08	3
doadur	AC		2
doadurms	7	20	3
doadwto	B	80	3
doadwtop	B	20	3
doadwtos	B	40	3
doaend	F8		2
doaerfail	91	08	3
doaErrCode	E0		2
doaevent	6	02	3
doaevher	8F	08	3
doaexec	B	02	3
doaexit13rc	BB	02	3
doaexpJCL	8F	01	3
doaexpjclUSED	B9	40	3
doae113	BB	20	3
doaflags	6		2
doaflag2	B		2
doaflag3	8F		2
doaflag4	91		2
doaflag5	B9		2
doaflag6	BA		2
doaflag7	BB		2

Name	Hex Offset	Hex Value	Level
doaform	3C		2
doafromPLC	B9	20	3
doaFTSAnop	B9	01	3
doahirc	46		2
doahour	0		2
doaia	70		2
doaimmerr	8D	04	3
doajcrt	92		3
doajhrc	7	40	3
doaJLInfo	8D	40	3
doajobn	10		2
doajpol	93		3
doaJRstat	90		2
doajsub	7	80	3
doalo	66		2
doalod	66		3
doalot	6C		3
doamaxd	A8		2
doaMaxdSec	DC		2
doamhld	B	08	3
doaminut	2		2
doamonth	2		2
doaMoreRun	8D	80	3
doanet	50		2
doanhld	7	01	3
doanoerr	BA	10	3
doaNoerrJT	36	80	2
doaNoerrSkip	36	40	2
doaNoEval	BA	02	3
doanop	B	04	3
doanxt	18		2
doanxttj	20		2
doaoccx	C		3
doaoist	A		2
doaOldJCL	B4		2
doaOpInfo	8D	20	3
doaopno	F		3
doaorer	7	10	3
doaores	7	08	3
doaosi	8F	20	3
doapre	1C		2

Name	Hex Offset	Hex Value	Level
doapretj	24		2
doapri	5A		2
doaps	7A		2
doapsd	7A		3
doapserv	52		2
doapst	80		3
doaPSUdone	8D	10	3
doaPSUjcl	B0		2
doaPSUwait	8D	01	3
doaPthOc	CC		3
doaPthOp	CF		3
doaRdSent	BA	20	3
doaRecEnd	B9	80	3
doarecjob	91	02	3
doarecprmp	91	04	3
doarecrun	91	01	3
doarejst	45		2
doarelft	84		2
doaremov	B	10	3
doaresub	8F	40	3
doarmsg	7	02	3
doar1	54		2
doar2	56		2
doasainfop	D0		2
doascrrt	44		2
doaseinfop	BC		2
doasending	91	20	3
doasenq	8F	02	3
doasfail	6	08	3
doashadow	BB	80	3
doashoseq	BB	04	3
doasrkep	8F	04	3
doasseq	34		2
doast	5C		2
doastamr	6	20	3
doastart	6	01	3
doastate	8		2
doastepsucc	BA	01	3
doasuber	91	10	3
doasubtoken	E4		2
doaSviewP	EC		2

Name	Hex Offset	Hex Value	Level
doasyssfail	8F	10	3
doaTaskType	F0		2
doatime	48		2
doatjt	6	10	3
doatoken	A4		3
doatoken_full	A0		2
doatoken_pre	A0		3
doaTZoff	58		2
doaTZONE	BA	04	3
doaUnexpRC	BB	10	3
doaurg	6	40	3
doaURGmsg	B9	02	3
doaUserSYS	8D	08	3
doaVdest	D4		2
doaver	4		2
doawaiTSE	B9	08	3
doawaitWQA	B9	04	3
doaWASUJ	B9	10	3
doawlm	92		2
doaWLMclass	C4		2
doaWLMmsg	BA	40	3
doaWLMprom	BA	80	3
doaWLMr	91	80	3
doawqap	28		2
doawrer	7	04	3
doayear	0		2
dsaAutoInfo	C		2
dsaAutoOper	10C		3
dsaCommText	C		3
dsaCommText1	C		4
dsaCommText2	4C		4
dsaCommText3	8C		4
dsaCommText4	CC		4
dsaCompInfo	11C		3
dsaEYE	0		2
dsainfo	0		1
dsanext	8		2
dsaSecElem	114		3
dsaVer	4		2
dseEYE	0		2
dseinfo	0		1

Name	Hex Offset	Hex Value	Level
dsenext	8		2
dseSEname	C		2
dseSysplexId	1C		2
dseVer	4		2

DPH - Daily plan report header record

Name : DCLDPH

Function:
Describes layout of a header record written in all DP plans to describe general data about the planning run. The record has the standard recsize and keysize of DPIN records.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	273	DPHAREA	HEADER RECORD
0	(0)	CHARACTER	114	DPHKEY	KEY
0	(0)	SIGNED	2	DPHTYPE	ALWAYS TYPE 1
2	(2)	SIGNED	2	DPHSTYPE	SUBTYPE 1=DPREC, 2=DPRES
4	(4)	CHARACTER	34	*	ADJUST FOR ADID
38	(26)	CHARACTER	76	*	TO ADD UP TO STD KEYSIZE
114	(72)	CHARACTER	159	DPHDATA	DATA PART OF DPH REC
114	(72)	CHARACTER	1	DPHPTYPE	TYPE OF PLANNING: 1 = TRIAL NEXT 2 = TRIAL REPLAN 3 = TRIAL FUTURE 4 = NEXT 5 = REPLAN 6 = NONREPORTING 7 = CP RESULTS
115	(73)	CHARACTER	1	DPHNCP	NEW NCP Y|N
116	(74)	CHARACTER	1	DPHFEEDB	MISSED FEEDBACK REPORT Y|N
117	(75)	CHARACTER	10	DPHOCP	USED OCP IDENTIFYER
117	(75)	CHARACTER	6	DPHOCPD	DATE (YYMMDD)
123	(7B)	CHARACTER	4	DPHOCPT	TIME (HHMM)
127	(7F)	CHARACTER	6	DPHLTP	USED LTP IDENTIFYER
127	(7F)	CHARACTER	6	DPHLTPD	DATE (YYMMDD)
133	(85)	CHARACTER	10	DPHSALG	PLANNING START (ALGORITHM)
133	(85)	CHARACTER	6	DPHSALGD	DATE (YYMMDD)
139	(8B)	CHARACTER	4	DPHSALGT	TIME (HHMM)
143	(8F)	CHARACTER	10	DPHSUSE	PLANNING START (USER)
143	(8F)	CHARACTER	6	DPHSUSED	DATE (YYMMDD)
149	(95)	CHARACTER	4	DPHSUSET	TIME (HHMM)
153	(99)	CHARACTER	10	DPHPLNT	TAIL PERIOD END
153	(99)	CHARACTER	6	DPHPLNTD	DATE (YYMMDD)
159	(9F)	CHARACTER	4	DPHPLNTT	TIME (HHMM)
163	(A3)	CHARACTER	10	DPHSREP	REPORT PERIOD START (RES)
163	(A3)	CHARACTER	6	DPHSREPD	DATE (YYMMDD)
169	(A9)	CHARACTER	4	DPHSREPT	TIME (HHMM)
173	(AD)	CHARACTER	10	DPHEREP	REPORT PERIOD END (RES)
173	(AD)	CHARACTER	6	DPHEREPD	DATE (YYMMDD)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
179	(B3)	CHARACTER	4	DPHEREPT	TIME (HHMM)
179	(B3)	CHARACTER	2	DPHEREPH	HH
181	(B5)	CHARACTER	2	DPHEREPM	MM
183	(B7)	CHARACTER	10	DPHEUSE	PLANNING END (USER)
183	(B7)	CHARACTER	6	DPHEUSED	DATE (YYMMDD)
189	(BD)	CHARACTER	4	DPHEUSET	TIME (HHMM)
193	(C1)	CHARACTER	6	DPHROPT	OPTIONS
193	(C1)	CHARACTER	1	DPHWSACT	1|0
194	(C2)	CHARACTER	1	DPHDAIOP	1|0
195	(C3)	CHARACTER	1	DPHALLWS	1|0
196	(C4)	CHARACTER	1	DPH1STWS	1|0
197	(C5)	CHARACTER	1	DPHNONRE	1|0
198	(C6)	CHARACTER	1	DPHCPRES	1|0
199	(C7)	CHARACTER	2	DPHROPTRU	OPTIONS, RES UTILIZATION
199	(C7)	CHARACTER	1	DPHPLNRU	PLANNED RES UTILIZ 1|0
200	(C8)	CHARACTER	1	DPHACTRU	ACTUAL RES UTILIZ 1|0
201	(C9)	CHARACTER	1	DPHCRIT	Y=REPORT PRESENT
202	(CA)	CHARACTER	8	DPH#MSG	DPIN MESSAGE COUNTERS
202	(CA)	SIGNED	2	DPH#MSGE	ERROR
204	(CC)	SIGNED	2	DPH#MSGW	WARNING
206	(CE)	SIGNED	2	DPH#MSGI	INFORMATION
208	(D0)	SIGNED	2	DPH#MSGU	UNRECOVERABLE
210	(D2)	UNSIGNED	4	DPH#OCCS	NUMBER OF PLANNED OCC
214	(D6)	UNSIGNED	4	DPH#OPS	NUMBER OF PLANNED OPS
218	(DA)	CHARACTER	55	*	ADD UP TO STD REC SIZE

Cross reference

Name	Hex Offset	Hex Value	Level
DPH#MSG	CA		3
DPH#MSGE	CA		4
DPH#MSGI	CE		4
DPH#MSGU	D0		4
DPH#MSGW	CC		4
DPH#OCCS	D2		3
DPH#OPS	D6		3
DPHACTRU	C8		4
DPHALLWS	C3		4
DPHAREA	0		1
DPHCPRES	C6		4
DPHCRIT	C9		3
DPHDAIOP	C2		4
DPHDATA	72		2
DPHEREP	AD		3
DPHEREPD	AD		4
DPHEREPH	B3		5

Name	Hex Offset	Hex Value	Level
DPHEREPM	B5		5
DPHEREPT	B3		4
DPHEUSE	B7		3
DPHEUSED	B7		4
DPHEUSET	BD		4
DPHFEEDB	74		3
DPHKEY	0		2
DPHLTP	7F		3
DPHLTPD	7F		4
DPHNCP	73		3
DPHNONRE	C5		4
DPHOCP	75		3
DPHOCPD	75		4
DPHOCPT	7B		4
DPHPLNRU	C7		4
DPHPLNT	99		3
DPHPLNTD	99		4
DPHPLNTT	9F		4
DPHPTYPE	72		3
DPHROPT	C1		3
DPHROPTRU	C7		3
DPHSALG	85		3
DPHSALGD	85		4
DPHSALGT	8B		4
DPHSREP	A3		3
DPHSREPD	A3		4
DPHSREPT	A9		4
DPHSTYPE	2		3
DPHSUSE	8F		3
DPHSUSED	8F		4
DPHSUSET	95		4
DPHTYPE	0		3
DPHWSACT	C1		4
DPH1STWS	C4		4

DPIN or DPUT - Report record in DP

Name : DCLDPIN

Function:
Describes standard of layout for all daily planner report (including message) records. There are several record types but they all have the same key size and record size. The sort step that follows step1 of a daily planner run sorts the record on that key.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	558	DPINREC	DPIN RECORD
0	(0)	CHARACTER	114	DPIKEY	KEY OF DPIN RECORD
0	(0)	SIGNED	2	DPITYPE	
					1 = HEADER RECORD SUBTYPE (DPH) 1 = DPREC HDR SUBTYPE 2 = DPRES HDR 2 = DAILY_OP RECORD (DAIOP) 3 = WS PLAN RECORDS SUBTYPE (WSP) 1 = WSP_OP RCD SSTYPE 1 = OPENTIME SSTYPE 2 = OPERATION SUBTYPE 2 = WSP_FOP RCD SUBTYPE 3 = WSN RCD SUBTYPE 4 = WSN_R1 SUBTYPE 5 = WSN_R2 4 = SUMMARY TABLES RECORD (CSUM) 5 = COMPLETED OCCURRENCE|OP RECORD SUBTYPE (CAPP) 1 = OLD 24-H SUBTYPE 2 = CURR 24-H 6 = OPER IN ERROR REC (OPERR) 7 = WS ACTIVITY RECORD SUBTYPE (WSN) 1 = WSNA RCD SUBTYPE 2 = WSN_R1 SUBTYPE 3 = WSN_R2 8 = COMPLETED OCCURRENCE ERR REC. SUBTYPE (ERRS) 1 = ERROR SUM SUBTYPE 2 = TOTAL SUM 9 = MISSED FEEDBACK (MFB) 13 = PLANNED UTIL HISTO (PRUH) 17 = PLANNED UTIL REPORT (PRUT) 20 = ACTUAL UTIL REPORT SUBTYPE (ARU) 1 = INTERVAL SUBTYPE 5 = DETAIL 30 = ERROR MESSAGE REC (DPMSG) 40 = CRITICAL PATH (CRIRP)
2	(2)	CHARACTER	78	*	TO ADD UP TO STD KEYSIZE
80	(50)	CHARACTER	34	*	TO ADD UP TO NEW KZ
114	(72)	CHARACTER	444	DPIDATA	DATA PART OF DPIN RECORD
114	(72)	CHARACTER	444	*	TO ADD UP TO STD REC SIZE

DQE - Data queue element

Name : DCLDQE

Function:
This segment maps queue elements for several queues. The mapping of the DQEDATA field varies depending on the value of DQETYPE.

If data buffers are used (indicated by dqebptr ^= 0), they are always allocated in subpool 2 by queue adders, and are freed when no longer needed by queue servers.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	152	dqe	data router q element
0	(0)	CHARACTER	4	dqedesc	block descriptor, DQE
4	(4)	CHARACTER	2	dqever	version number, 01
6	(6)	BITSTRING	2	*	reserved flags
8	(8)	CHARACTER	3	dqetype	data type
11	(B)	BITSTRING	1	dqeflags	flags
		1...		dqeflres	reset seqds
		.1..		dqecmexs	cm execution suspended for this element
		..1.		dqenckpt	don't chkpt this submit
		...1		dqepace	pace the event flood
		.... 1...		dqenoedp	no edp chkpt for this ev
		.... .1..		*	reserved
		.... ..1.		dqe23psus	flag for A2A3P suspANEABKLC
		1		dqeSPIN	on = SPIN enabled off= SPIN disabled
12	(C)	ADDRESS	4	dqeecbp	addr of service requestor ecb
16	(10)	ADDRESS	4	dqebptr	external data buffer ptr
20	(14)	SIGNED	4	dqeblen	size of external buffer
24	(18)	CHARACTER	8	dqedest	destination id
32	(20)	SIGNED	4	dqermax	max of recs per cyc in eds
36	(24)	SIGNED	4	dqebltot	total size of external buffer
40	(28)	SIGNED	2	dqebsnbr	buffer sequence number
42	(2A)	SIGNED	2	dqebstot	buffer sequence tot nbr in seq
44	(2C)	CHARACTER	100	dqedata	local data buffer
144	(90)	SIGNED	4	dqeadder	Additional recs w buffer
148	(94)	BITSTRING	1	dqeflags1	additional flags
		1...		dqesp131	ext buf in sp 131
		.111 1111		*	reserved
149	(95)	CHARACTER	1	*	reserved
150	(96)	UNSIGNED	2	dqeevts	current evds submit seq
152	(98)	CHARACTER		dqeend	end of dqe

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	100	dqeevt	event data buffer mapping
0	(0)	SIGNED	2	dqennum	opc node number
2	(2)	SIGNED	2	dqerdrn	event reader number in node
4	(4)	CHARACTER	8	dqepos	event ds position
4	(4)	SIGNED	4	dqecyc	write cycle number
8	(8)	SIGNED	4	dqerec	record number in cycle
12	(C)	CHARACTER	8	dqereld	release destination or blank
20	(14)	CHARACTER	80	dqeexr	exit record, see dclexr

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	6	dqeclj	a DASD tracker needs BTVA
0	(0)	CHARACTER	5	dqecljob	clnjob prefix and BTVA
5	(5)	CHARACTER	1	dqedsclas	datastore class if JCC BTVA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	20	dqerfw	rfw data
0	(0)	CHARACTER	8	dqerfwdest	destination name
8	(8)	CHARACTER	8	dqeedp	edp information
8	(8)	SIGNED	4	dqeedpwcy	write cycle number
12	(C)	SIGNED	4	dqeedprec	record of last record
16	(10)	SIGNED	4	dqerfwnnum	node number

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	48	dqerel	release job mapping
0	(0)	CHARACTER	8	dqerjbnm	job name
8	(8)	CHARACTER	8	dqerjbid	job number
16	(10)	CHARACTER	8	dqercnje	current nje node
24	(18)	CHARACTER	8	dqeronje	origin nje node
32	(20)	UNSIGNED	2	dqeaseq	submit sequence
34	(22)	CHARACTER	4	dqeawsid	work station id
38	(26)	UNSIGNED	2	*	reserved
40	(28)	SIGNED	4	dqerojid	original job number
44	(2C)	CHARACTER	4	dqeassnam	controller subsystem name

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	100	dqesub	submit data mapping
0	(0)	CHARACTER	4	dqerus	**RU or blank
4	(4)	CHARACTER	8	dqetso	Tso user id or blank
12	(C)	CHARACTER	8	dqejobn	jobSTC name
20	(14)	SIGNED	4	dqenrec	no of jcl card images
24	(18)	CHARACTER	4	dqewsid	work station id
28	(1C)	CHARACTER	16	dqeadid	application ID
44	(2C)	CHARACTER	10	dqeocia	occurrence input arrival
54	(36)	SIGNED	2	dqeopnum	operation number
56	(38)	CHARACTER	4	dqessnam	controller subsystem name
60	(3C)	UNSIGNED	2	dqesseq	submit sequence
62	(3E)	CHARACTER	8	dqejid	job of last subbed job
70	(46)	SIGNED	2	dqeasid	job asid D90C
		1...		dqeckp	dqeasid not used. 1 bit CXHA
		.1..		dqepromote	promoted job
70	(46)	BITSTRING	1	*	used for DASD connection CXHA
72	(48)	CHARACTER	8	dqetoken_full	
72	(48)	CHARACTER	4	dqetoken_pre	operation token prefix
72	(48)	CHARACTER	4	DQEOCCtok1	OCC token part 1 D90A
76	(4C)	UNSIGNED	4	dqetoken	operation token
76	(4C)	CHARACTER	4	DQEOCCtok2	OCC token part 2 D90A
80	(50)	ADDRESS	4	dqeeoptp	e2e option user area
84	(54)	CHARACTER	8	dqenetid	APPC netid (final dest)
84	(54)	CHARACTER	8	dqesubtoken	HTTP submission token
92	(5C)	CHARACTER	8	dqenetLU	APPC netLU (final dest)
92	(5C)	CHARACTER	8	dqeSdest	submit dest COBA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	28	dqersseq	sseq sync request

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	CHARACTER	4	dqerwsid	work station id
4	(4)	CHARACTER	8	dqecreat	request evt creation time9TA
4	(4)	SIGNED	4	dqedate	date format (00yydddf)
8	(8)	SIGNED	4	dqetime	time format (secs*100)
12	(C)	CHARACTER	4	dqerssnam	controller subsystem name
16	(10)	UNSIGNED	1	dqerseq	request sequence from w
17	(11)	BITSTRING	1	dqerflg1	flagbyte byte 1
		1...		dqerask	req for curr evds sseq
		.1..		dqercold	cold start the ws
		..1.		dqerdlte	remove the ws from evds
		...1		dqerset	req to set evds to wseq
		 1111		*	reserved
18	(12)	UNSIGNED	2	dqewseq	submit sequence from ws
20	(14)	CHARACTER	8	dqeJdest	dqedest COBA

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	18	dqecmini	initialize CM for operCATA
0	(0)	CHARACTER	8	dqecmjob	jobSTC name CATA
8	(8)	SIGNED	4	dqecmocc	occurrence number CATA
12	(C)	SIGNED	4	dqecmopr	operation number CATA
16	(10)	CHARACTER	2	dqecmrt	TASK ID requestor CATA EM = Event Manager CATA AR = Automatic Recovery CATA GS = General Service CATA

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	100	dqerfp	RODM request for parms
0	(0)	CHARACTER	8	dqerfpd	requestor destination
8	(8)	CHARACTER	92	*	reserved

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	100	dqersl	RODM subsystem lost
0	(0)	CHARACTER	8	dqersldn	requestor destination
8	(8)	CHARACTER	4	dqerslssn	subsystem name
12	(C)	CHARACTER	88	*	reserved

$CRFA
Dynamic Critical Path $CRFA
dqedata containing data sent to critical path handler $CRFA
when an MCP action is performed on a critical predecessor $CRFA
and the dataspace has to be updated consequently $CRFA
$CRFA

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	100	dqecrt	Crit Path recalc CRFA
0	(0)	CHARACTER	4	dqews	workstation name CRFA
4	(4)	CHARACTER	1	dqestatus	operation status CRFA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
5	(5)	CHARACTER	1	dqeexstat	oper extended status CRFA
6	(6)	BITSTRING	1	dqeflags2	flags CRFA
		1...		dqenoped	noped operation CRFA
		.1..		dqemanheld	oper manually held CRFA
		..1.		dqemheldch	first dqe of a seq CRFA
		...1		dqenopch	last dqe of a seq CRFA
		.... 1...		dqetmdch	time dep chg CWWA
		.... .1..		dqetmdep	time dep CWWA
		.... ..1.		dqeopiach	oper ia changed CWWA
		1		*	free CWWC
7	(7)	BITSTRING	1	dqeflags3	free CRFA
		1...		dqeisfirst	first dqe of a seq CRFA
		.1..		dqeaddjob	add job to dataspace CRFA
		..1.		dqeislast	last dqe of a seq CRFA
		...1 1111		*	free CRFA
8	(8)	CHARACTER	8	dqejobname	job name CRFA
16	(10)	SIGNED	4	dqejobenix	job table entry indexCRFA
20	(14)	CHARACTER	1	dqejobprty	job priority CRFA
21	(15)	CHARACTER	1	dqereqtype	MCP request type CRFA S - status change CRFA A - Add operationdepCRFA D - Delete (oper, depCRFA M - Modify CRFA P - Critical indicat CRFA
22	(16)	CHARACTER	2	*	free CRFA
24	(18)	SIGNED	4	dqeindpred	pred j table entry ixCRFA
28	(1C)	CHARACTER	1	dqeoldstat	operation old status CRFA
29	(1D)	CHARACTER	1	dqecritind	critical indicator CRFA
30	(1E)	CHARACTER	10	dqeinparr	input arrival time CRFA
30	(1E)	CHARACTER	6	dqeinparrd	date CRFA
36	(24)	CHARACTER	4	dqeinparrt	time CRFA
40	(28)	CHARACTER	2	*	free CRFA
42	(2A)	CHARACTER	10	dqedeadl	deadline CRFA
42	(2A)	CHARACTER	6	dqedeadld	date CRFA
48	(30)	CHARACTER	4	dqedeadlt	time CRFA
52	(34)	CHARACTER	2	*	free CRFA
54	(36)	CHARACTER	10	dqeastart	actual start time CRFA
54	(36)	CHARACTER	6	dqeastartd	date CRFA
60	(3C)	SIGNED	4	dqeastartt	time CRFA
64	(40)	CHARACTER	2	*	free CRFA
66	(42)	CHARACTER	10	dqeaend	actual end time CRFA
66	(42)	CHARACTER	6	dqeaendd	date CRFA
72	(48)	SIGNED	4	dqeaendt	time CRFA
76	(4C)	SIGNED	4	dqeduration	duration CRFA
80	(50)	SIGNED	4	dqeactdur	actual duration CRFA
84	(54)	CHARACTER	4	dqeoprkey	operation index CRFA
84	(54)	UNSIGNED	3	dqeoccidx	occ number CRFA CRFA
87	(57)	UNSIGNED	1	dqeopridx	oper number CRFA
88	(58)	CHARACTER	4	dqeerrcode	job error code CRFA
92	(5C)	CHARACTER	4	dqeprekey	pred oper index CRFA
92	(5C)	UNSIGNED	3	dqepoccidx	occ number CRFA CRFA
95	(5F)	UNSIGNED	1	dqepopridx	oper number CRFA
96	(60)	CHARACTER	4	*	free CRFA

$CRFA
External buffer for dqecrt $CRFA
It contains WLM data Policy and Service Class $CRFA
$CRFA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	40	dqecrtBUF	DQE Type CRT CRFA
0	(0)	CHARACTER	1	dqewlmpol	WLM policy CRFA
1	(1)	CHARACTER	5	*	free CRFA
6	(6)	CHARACTER	8	dqewlmclass	WLM Service Class CRFA
14	(E)	CHARACTER	10	dqelstart	latest start CRFA
14	(E)	CHARACTER	6	dqelstartd	latest start date CRFA
20	(14)	SIGNED	4	dqelstartt	ltst start time CRFACWWC
24	(18)	CHARACTER	16	dqeopIA	operation IA CWWA
24	(18)	CHARACTER	6	dqeopIAD	date CWWA
30	(1E)	CHARACTER	4	dqeopIAT	time CWWA
34	(22)	CHARACTER	6	*	CWWA

$CRFA
Dynamic Critical Path $CRFA
dqedata containing data sent to critical path handler $CRFA
when a status change EM occurs or when the job is late or $CRFA
long running WA $CRFA
$CRFA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	100	dqecr1	Crit Path recalc CRFA
0	(0)	CHARACTER	1	dqestatu1	operation status CRFA
1	(1)	CHARACTER	1	dqeexstat1	oper extended status CRFA
2	(2)	CHARACTER	2	*	free CRFA
4	(4)	SIGNED	4	dqejobeni1	job table entry indexCRFA
8	(8)	CHARACTER	1	dqereqtyp1	MCP request type CRFA S - status change CRFA L - Late CRFA R - Long Running CRFA
9	(9)	CHARACTER	1	dqeoldsta1	operation old status CRFA
10	(A)	BITSTRING	1	dqeflags4	flags CRFA
		1...		dqeurgch1	doa urgent queue chngCRFA
		.1..		dqedoaur1	doa urgent queue flagCRFA
		..1.		dqewlmpro1	WLM promotion flag CRFA
		...1 1111		*	free CRFA
11	(B)	CHARACTER	3	*	free CRFA
14	(E)	CHARACTER	10	dqeastar1	actual start time CRFA
14	(E)	CHARACTER	6	dqeastar1d	date CRFA
20	(14)	SIGNED	4	dqeastar1t	time CRFA
24	(18)	CHARACTER	2	*	free CRFA
26	(1A)	CHARACTER	10	dqeaen1	actual end time CRFA
26	(1A)	CHARACTER	6	dqeaen1d	date CRFA
32	(20)	SIGNED	4	dqeaen1t	time CRFA
36	(24)	SIGNED	4	dqeactdu1	actual duration CRFA
40	(28)	CHARACTER	4	dqeoprke1	operation index CRFA
40	(28)	UNSIGNED	3	dqeoccid1	occ number CRFA CRFA
43	(2B)	UNSIGNED	1	dqeoprid1	oper number CRFA
44	(2C)	CHARACTER	4	dqeerrcod1	job error code CRFA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
48	(30)	CHARACTER	52	*	free CRFA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	100	dqearc	DQE Type OPC Data Sto D90C
0	(0)	CHARACTER	4	dqearctyp	Service Required D90C LOG = get MVS JobLog D90C SLO = get Struct. Log D90A OPI = ask Oper_info D90A SDEL= start delete D90A of old entries D90A

D90A common part D90A
D90A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
4	(4)	CHARACTER	24	dqearcJRkey	Job Log retrieval key D90C
4	(4)	CHARACTER	8	dqearcJobId	Job Log Id D90C
12	(C)	CHARACTER	8	dqearcJobNa	Job Log Name D90C
20	(14)	SIGNED	4	dqearcRdrD	Job Start Rdr Date D90C
24	(18)	SIGNED	4	dqearcRdrT	Job Start Rdr Time D90C
28	(1C)	CHARACTER	8	dqearcdest	output DEST D90C
36	(24)	CHARACTER	16	dqearcADID	Application Name D90C
52	(34)	CHARACTER	10	dqearcIA	IA Date and Time D90C
52	(34)	CHARACTER	6	dqearcIADate	IA Date D90C
58	(3A)	CHARACTER	4	dqearcIATime	IA Time D90C
62	(3E)	CHARACTER	9	dqearcOPkey	Operation VSAM key D90C
62	(3E)	CHARACTER	8	dqearcOCC	OCC token D90C
70	(46)	UNSIGNED	1	dqearcOPR	Opr num D54A
71	(47)	CHARACTER	2	dqearccaller	EM=Event manager D90A WA=WorkStat.Analyzer D90A AR=Automatic Recover D90A GS=General Server D90A

D90A OPI type only D90A
D90A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
73	(49)	CHARACTER	1	dqearcfla1	flags (Ask Oper_Info) D90A
		1...		ARC_PSUpost	1 = post PSU D90A
		.1..		ARC_CleanUp	1 = Stand Alone CLNUP D90A
		..1.		ARC_SL	1 = StepList required D90A
		...1		ARC_DS	1 = DSList required D90A
		.... 1...		ARC_ExpJCL	1 = use expanded JCL D90A
		.... .1..		ARC_BestStep	1 = start from BSTEP BQBA
		.... ..1.		ARC_noask	1 = cp14nostr ON CAEA
		1		ARC_AskSimGDG	1 = GDG sim required CLZA
74	(4A)	CHARACTER	8	*	free BUTA
82	(52)	CHARACTER	8	dqearcUSER	original job userid BUTA

D90A Clean Up only D90A
D90A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
90	(5A)	CHARACTER	8	dqearcEXdest	execution destination

01A AR restart only

01A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
98	(62)	UNSIGNED	1	dqearcopiRet	retry counter
99	(63)	UNSIGNED	1	dqearcARste	AR restart step 01ABVZC (currently not used) BVZA

$BUTA
AR restart only $BUTA
structure mapped into DQEARC buffer $BUTA
$BUTA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	16	dqearcBUF	DQE Type OPC Data Sto BUTA
0	(0)	CHARACTER	8	dqearcSTEPN	Stepname (AR) BUTA
8	(8)	CHARACTER	8	dqearcPSTEPN	Proc Step Name (AR) BUTA

$CRFA
Dynamic Critical Path $CRFA
external buffer containing part of the critical predecessor $CRFA
network entry point and its predecessor See DCLJOBEN $CRFA
$CRFA

NOTE
For the dqearc SDEL type the information are all contained
in the buffer pointed by dqebptr
dqebptr CP16 record layout

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	100	dqePSU	DQE Type OPC Data Sto D90A
0	(0)	CHARACTER	4	dqePSUtyp	Service Required D90A APL = Apply logic D90A CLN = Stand Alone CLN D90A REM = resume suspendedD90A DEL = delete request D90A

D90A common part D90A
D90A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
4	(4)	CHARACTER	45	dqePSUcommon	D90A
4	(4)	CHARACTER	2	dqePSUcaller	FL=Fetch JobLog D90A GS=General Server D90A EM=Event manager D90A
6	(6)	CHARACTER	8	dqePSUJobNa	Job Name D90A
14	(E)	CHARACTER	9	dqePSUOPkey	Operation VSAM key D90A
14	(E)	CHARACTER	8	dqePSUOCC	Occ token D90A
22	(16)	UNSIGNED	1	dqePSUOPR	Opr num D90A
23	(17)	CHARACTER	16	dqePSUADID	Application Name D90A
39	(27)	CHARACTER	10	dqePSUIA	IA Date and Time D90A
39	(27)	CHARACTER	6	dqePSUIADate	IA Date D90A
45	(2D)	CHARACTER	4	dqePSUIATime	IA Time D90A

D90A GS only D90A
D90A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
49	(31)	CHARACTER	8	dqePSUuser	TSO userid D90A
57	(39)	CHARACTER	4	dqePSUtoken	GS request token D90A

D90A FL for AR only D90A
D90A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
61	(3D)	CHARACTER	8	dqeStepname	AR restart step D90A
69	(45)	CHARACTER	8	dqePStepName	AR restart proc step D90A

D90A CLN type only D90A
D90A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
77	(4D)	CHARACTER	8	dqePSUexdest	execution DEST D90A

D90A APL type only D90A
D90A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
85	(55)	CHARACTER	1	dqePSUfla1	D90A
		1...		PSU_ExpJCL	1 = use expanded JCL D90A
		.1..		PSU_Suspend	1 = suspend DQE D90A
		..1.		PSU_Operinfo	1 = SL from buffer D90A
		...1		PSU_UserSys	1 = add usersys DD
		.... 1...		PSU_SimGDG	1 = simulate GDG BLBA
		.... .1..		PSU_Root	1 = GDG root list DRIV13A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.... ..1.		PSU_BestStep	1 = start from BSTEP BQBA
		1		PSU_AskSimGDG	1 = req simulation CLZC

```
2 char2   free
2 dqePSULen
```

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
86	(56)	UNSIGNED	2	PSU_SLlen	SL length BLBCBUUC BUUA
88	(58)	UNSIGNED	3	PSU_DSlen	DL length BLBCBUUC BUUA
91	(5B)	UNSIGNED	3	PSU_GDGlen	Sim. GDG info BLBABUUC BUUA
94	(5E)	UNSIGNED	2	PSU_Rootlen	Root len DRIV13CBUUC BUUA
96	(60)	SIGNED	4	PSU_JCLlen	JCL length D90ABUUC BUUA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	dqeWLM	sub WLM reset request D63A
0	(0)	CHARACTER	8	dqetsoW	Tso user id or blank D63A
8	(8)	CHARACTER	8	dqejobnW	jobSTC name D63A
16	(10)	CHARACTER	8	dqeSVCnW	WLM Hi perf service class
24	(18)	CHARACTER	4	dqewsidW	work station id D63A
28	(1C)	CHARACTER	16	dqeadidW	application ID D63A
44	(2C)	CHARACTER	10	dqeociaW	occurrence input arrival
44	(2C)	CHARACTER	6	dqeociaD	occurence IA date D63A
50	(32)	CHARACTER	4	dqeociaT	occurence IA time D63A
54	(36)	SIGNED	2	dqeopnumW	operation number D63A
56	(38)	CHARACTER	4	dqessnamW	controller subsystem name
60	(3C)	UNSIGNED	2	dqesseqW	submit sequence D63A

```
2 dqejidW char8    job of last subbed job
2 char2    reserved D63D
```

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
62	(3E)	SIGNED	2	dqeasidW	job asid D63A

```
2 dqetokenfullW    operation token D63d
  3 dqetokenpreW char4    operation token prefix
  3 dqetokenW unsigned bin32   operation token
2 dqeaccmpW ptr31   acc meth parms Q0A
2 dqenetidW char8   APPC netid final dest
2 dqenetLUW char8 semi APPC netLU final dest
```

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
64	(40)	SIGNED	4	dqeretcW	WLM promot request RC
68	(44)	SIGNED	2	dqersncW	WLM promot request RSN
70	(46)	CHARACTER	10	*	reserved D63A

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	8	dqeTcpIp	tcpip evt mapping CMDa
0	(0)	ADDRESS	4	dqeSocketIdPtr	CMDa
4	(4)	CHARACTER	4	dqeSocketDomain	CMDa

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	18	dqeConfFile	Conf File mapping 41a
0	(0)	CHARACTER	1	dqeRequestType	41a
1	(1)	CHARACTER	3	*	41a
4	(4)	CHARACTER	4	dqeRequestCRC	41a
8	(8)	CHARACTER	8	dqeCRCOwnerDest	41a
16	(10)	SIGNED	2	dqeDtbDestIndex	41a

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	44	dqeHTTPaction	HTTP Joblog retrieval Kill Query job status
0	(0)	CHARACTER	8	dqeHTocctoken	HTTP occurence token
8	(8)	CHARACTER	8	dqeHTsubtoken	HTTP submission token
16	(10)	CHARACTER	8	dqeHTjobname	HTTP job name
24	(18)	CHARACTER	4	dqeHTwsname	HTTP ws name
28	(1C)	CHARACTER	4	dqeHTssname	HTTP subsystem name
32	(20)	SIGNED	2	dqeHTopnum	HTTP operation number
34	(22)	SIGNED	2	*	reserved
36	(24)	CHARACTER	8	dqeHTuser	HTTP joblog req user

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	100	dqeHTnotify	HTTP notify
0	(0)	CHARACTER	56	dqeHNalias	alias
56	(38)	CHARACTER	4	dqeHNerrc	error code (stat EF)
60	(3C)	CHARACTER	6	dqeHNstartd	start date
66	(42)	CHARACTER	6	dqeHNendd	end date
72	(48)	SIGNED	4	dqeHNstartt	start time
76	(4C)	SIGNED	4	dqeHNendt	end time
80	(50)	CHARACTER	8	dqeHNxdtoken	XD99 key
88	(58)	SIGNED	4	dqeHNadur	duration in seconds
92	(5C)	CHARACTER	1	dqeHNstatus	status (sSCEF)
93	(5D)	CHARACTER	7	*	free

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	100	dqeHTinfo	HTTP bind infofailed
0	(0)	CHARACTER	56	dqeHIalias	alias
56	(38)	CHARACTER	8	dqeHIjobname	matched jobname
64	(40)	CHARACTER	10	dqeHIia	matched ia
64	(40)	CHARACTER	6	dqeHIiad	&period..IA date
70	(46)	CHARACTER	4	dqeHIiat	&period..IA time
74	(4A)	CHARACTER	1	dqeHItype	I=info, F=failed
75	(4B)	CHARACTER	1	dqeHIflags	flags
		1...		dqeHIbcp	&period..ON bin dis in CP

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
		.111 1111		*	&period..free
76	(4C)	CHARACTER	8	dqeHIxdtoken	XD99 key
84	(54)	CHARACTER	4	dqeHIwsname	matched ws name
88	(58)	CHARACTER	12	*	free

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	100	dqeHTsubscr	HTTP subscribe
0	(0)	CHARACTER	35	dqeHSrkeyz	remote job key
0	(0)	CHARACTER	16	dqeHSadid	&period..remote adidjsname (DZ)
16	(10)	CHARACTER	16	dqeHSjsws	&period..remote jsws (D)
32	(20)	CHARACTER	3	dqeHSopno	&period..remote opno (Z)
35	(23)	CHARACTER	1	dqeHSflags	flags
		1...		dqeHSzos	&period..ON remote eng type is Z
		.1..		dqeHSresub	&period..ON resub for sync
		..11 1111		*	&period..free
36	(24)	CHARACTER	2	*	free
38	(26)	CHARACTER	10	dqeHSia	IA for match
38	(26)	CHARACTER	6	dqeHSiad	&period..IA date
44	(2C)	CHARACTER	4	dqeHSiat	&period..IA time
48	(30)	CHARACTER	52	dqeHSalias	data for alias
48	(30)	CHARACTER	8	dqeHSocctoken	&period..occurence token
56	(38)	CHARACTER	8	dqeHSsubtoken	&period..submission token
64	(40)	CHARACTER	8	dqeHSjobname	&period..job name
72	(48)	CHARACTER	4	dqeHSwsname	&period..ws name
76	(4C)	CHARACTER	4	dqeHSssname	&period..subsystem name
80	(50)	SIGNED	2	dqeHSopnum	&period..operation number
82	(52)	CHARACTER	18	*	free

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	100	dqeHTincident	HTTP open incident
0	(0)	CHARACTER	16	dqeHTIncAdid	jobstream
16	(10)	CHARACTER	4	dqeHTIncWs	workstation
20	(14)	CHARACTER	4	dqeHTIncOpno	oper number
24	(18)	CHARACTER	10	dqeHTIncIA	IA for match
24	(18)	CHARACTER	6	dqeHTIncIad	&period..IA date
30	(1E)	CHARACTER	4	dqeHTIncIat	&period..IA time
34	(22)	CHARACTER	8	dqeHTIncJobname	job name
42	(2A)	CHARACTER	8	dqeHTIncJobnum	job number
50	(32)	CHARACTER	4	dqeHTIncErrCode	Error code
54	(36)	CHARACTER	4	dqeHTIncSSName	subsystem name
58	(3A)	BITSTRING	1	dqeHTIncFlags	flags
		1...		dqeHTIncE2E	use E2E error code
		.111 1111		*	reserved
59	(3B)	CHARACTER	1	*	reserved
60	(3C)	SIGNED	4	dqeHTIncE2ECode	E2E error code
64	(40)	CHARACTER	24	dqeHTIncOpText	operation text
88	(58)	CHARACTER	12	*	free

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	40	dqeHSbuff	DQE Type HTS
0	(0)	CHARACTER	40	dqeHSrjobnm	remote jobname

Cross reference

Name	Hex Offset	Hex Value	Level
ARC_AskSimGDG	49	01	3
ARC_BestStep	49	04	3
ARC_CleanUp	49	40	3
ARC_DS	49	10	3
ARC_ExpJCL	49	08	3
ARC_noask	49	02	3
ARC_PSUpost	49	80	3
ARC_SL	49	20	3
dqe	0		1
dqeactdur	50		2
dqeactdu1	24		2
dqeadder	90		2
dqeaddjob	7	40	3
dqeadid	1C		2
dqeadidW	1C		2
dqeaend	42		2
dqeaendd	42		3
dqeaendt	48		3
dqeaen1	1A		2
dqeaen1d	1A		3
dqeaen1t	20		3
dqearc	0		1
dqearcADID	24		2
dqearcARste	63		2
dqearcBUF	0		1
dqearccaller	47		2
dqearcdest	1C		2
dqearcEXdest	5A		2
dqearcfla1	49		2
dqearcIA	34		2
dqearcIADate	34		3
dqearcIATime	3A		3
dqearcJobId	4		3
dqearcJobNa	C		3
dqearcJRkey	4		2

Name	Hex Offset	Hex Value	Level
dqearcOCC	3E		3
dqearcopiRet	62		2
dqearcOPkey	3E		2
dqearcOPR	46		3
dqearcPSTEPN	8		2
dqearcRdrD	14		3
dqearcRdrT	18		3
dqearcSTEPN	0		2
dqearctyp	0		2
dqearcUSER	52		2
dqeaseq	20		2
dqeasid	46		2
dqeasidW	3E		2
dqeassnam	2C		2
dqeastart	36		2
dqeastartd	36		3
dqeastartt	3C		3
dqeastar1	E		2
dqeastar1d	E		3
dqeastar1t	14		3
dqeawsid	22		2
dqeblen	14		2
dqebltot	24		2
dqebptr	10		2
dqebsnbr	28		2
dqebstot	2A		2
dqeckp	46	80	3
dqeclj	0		1
dqecljob	0		2
dqecmexs	B	40	3
dqecmini	0		1
dqecmjob	0		2
dqecmocc	8		2
dqecmopr	C		2
dqecmrt	10		2
dqeConfFile	0		1
dqeCRCOwnerDest	8		2
dqecreat	4		2
dqecritind	1D		2
dqecrt	0		1
dqecrtBUF	0		1

Name	Hex Offset	Hex Value	Level
dqecr1	0		1
dqecyc	4		3
dqedata	2C		2
dqedate	4		3
dqedeadl	2A		2
dqedeadld	2A		3
dqedeadlt	30		3
dqedesc	0		2
dqedest	18		2
dqedoaur1	A	40	3
dqedsclas	5		2
dqeDtbDestIndex	10		2
dqeduration	4C		2
dqeecbp	C		2
dqeedp	8		2
dqeedprec	C		3
dqeedpwcy	8		3
dqeend	98		2
dqeeoptp	50		2
dqeerrcode	58		2
dqeerrcod1	2C		2
dqeevt	0		1
dqeevts	96		2
dqeexr	14		2
dqeexstat	5		2
dqeexstat1	1		2
dqeflags	B		2
dqeflags1	94		2
dqeflags2	6		2
dqeflags3	7		2
dqeflags4	A		2
dqeflres	B	80	3
dqeHIalias	0		2
dqeHIbcp	4B	80	3
dqeHIflags	4B		2
dqeHIia	40		2
dqeHIiad	40		3
dqeHIiat	46		3
dqeHIjobname	38		2
dqeHItype	4A		2
dqeHIwsname	54		2

Name	Hex Offset	Hex Value	Level
dqeHIxdtoken	4C		2
dqeHNadur	58		2
dqeHNalias	0		2
dqeHNendd	42		2
dqeHNendt	4C		2
dqeHNerrc	38		2
dqeHNstartd	3C		2
dqeHNstartt	48		2
dqeHNstatus	5C		2
dqeHNxdtoken	50		2
dqeHSadid	0		3
dqeHSalias	30		2
dqeHSbuff	0		1
dqeHSflags	23		2
dqeHSia	26		2
dqeHSiad	26		3
dqeHSiat	2C		3
dqeHSjobname	40		3
dqeHSjsws	10		3
dqeHSocctoken	30		3
dqeHSopno	20		3
dqeHSopnum	50		3
dqeHSresub	23	40	3
dqeHSrjobnm	0		2
dqeHSrkeyz	0		2
dqeHSssname	4C		3
dqeHSsubtoken	38		3
dqeHSwsname	48		3
dqeHSzos	23	80	3
dqeHTIncAdid	0		2
dqeHTIncErrCode	32		2
dqeHTIncE2E	3A	80	3
dqeHTIncE2ECode	3C		2
dqeHTIncFlags	3A		2
dqeHTIncIA	18		2
dqeHTIncIad	18		3
dqeHTIncIat	1E		3
dqeHTincident	0		1
dqeHTIncJobname	22		2
dqeHTIncJobnum	2A		2
dqeHTIncOpno	14		2

Name	Hex Offset	Hex Value	Level
dqeHTIncOpText	40		2
dqeHTIncSSName	36		2
dqeHTIncWs	10		2
dqeHTinfo	0		1
dqeHTjobname	10		2
dqeHTnotify	0		1
dqeHTocctoken	0		2
dqeHTopnum	20		2
dqeHTssname	1C		2
dqeHTsubscr	0		1
dqeHTsubtoken	8		2
dqeHTTPaction	0		1
dqeHTuser	24		2
dqeHTwsname	18		2
dqeindpred	18		2
dqeinparr	1E		2
dqeinparrd	1E		3
dqeinparrt	24		3
dqeisfirst	7	80	3
dqeislast	7	20	3
dqeJdest	14		2
dqejid	3E		2
dqejobenix	10		2
dqejobeni1	4		2
dqejobn	C		2
dqejobname	8		2
dqejobnW	8		2
dqejobprty	14		2
dqelstart	E		2
dqelstartd	E		3
dqelstartt	14		3
dqemanheld	6	40	3
dqemheldch	6	20	3
dqenckpt	B	20	3
dqenetid	54		2
dqenetLU	5C		2
dqennum	0		2
dqenoedp	B	08	3
dqenopch	6	10	3
dqenoped	6	80	3
dqenrec	14		2

Name	Hex Offset	Hex Value	Level
dqeoccidx	54		3
dqeoccid1	28		3
DQEOCCtok1	48		4
DQEOCCtok2	4C		4
dqeocia	2C		2
dqeociaD	2C		3
dqeociaT	32		3
dqeociaW	2C		2
dqeoldstat	1C		2
dqeoldsta1	9		2
dqeopIA	18		2
dqeopiach	6	02	3
dqeopIAD	18		3
dqeopIAT	1E		3
dqeopnum	36		2
dqeopnumW	36		2
dqeopridx	57		3
dqeoprid1	2B		3
dqeoprkey	54		2
dqeoprke1	28		2
dqepace	B	10	3
dqepoccidx	5C		3
dqepopridx	5F		3
dqepos	4		2
dqeprekey	5C		2
dqepromote	46	40	3
dqePStepName	45		2
dqePSU	0		1
dqePSUADID	17		3
dqePSUcaller	4		3
dqePSUcommon	4		2
dqePSUexdest	4D		2
dqePSUfla1	55		2
dqePSUIA	27		3
dqePSUIADate	27		4
dqePSUIATime	2D		4
dqePSUJobNa	6		3
dqePSUOCC	E		4
dqePSUOPkey	E		3
dqePSUOPR	16		4
dqePSUtoken	39		2

Name	Hex Offset	Hex Value	Level
dqePSUtyp	0		2
dqePSUuser	31		2
dqerask	11	80	3
dqercnje	10		2
dqercold	11	40	3
dqerdlte	11	20	3
dqerdrn	2		2
dqerec	8		3
dqerel	0		1
dqereld	C		2
dqereqtype	15		2
dqereqtyp1	8		2
dqeRequestCRC	4		2
dqeRequestType	0		2
dqeretcW	40		2
dqerflg1	11		2
dqerfp	0		1
dqerfpd	0		2
dqerfw	0		1
dqerfwdest	0		2
dqerfwnnum	10		2
dqerjbid	8		2
dqerjbnm	0		2
dqermax	20		2
dqerojid	28		2
dqeronje	18		2
dqerseq	10		2
dqerset	11	10	3
dqersl	0		1
dqersldn	0		2
dqerslssn	8		2
dqersncW	44		2
dqersseq	0		1
dqerssnam	C		2
dqerus	0		2
dqerwsid	0		2
dqeSdest	5C		3
dqeSocketDomain	4		2
dqeSocketIdPtr	0		2
dqeSPIN	B	01	3
dqesp131	94	80	3

Name	Hex Offset	Hex Value	Level
dqesseq	3C		2
dqesseqW	3C		2
dqessnam	38		2
dqessnamW	38		2
dqestatus	4		2
dqestatu1	0		2
dqeStepname	3D		2
dqesub	0		1
dqesubtoken	54		3
dqeSVCnW	10		2
dqeTcpIp	0		1
dqetime	8		3
dqetmdch	6	08	3
dqetmdep	6	04	3
dqetoken	4C		3
dqetoken_full	48		2
dqetoken_pre	48		3
dqetso	4		2
dqetsoW	0		2
dqetype	8		2
dqeurgch1	A	80	3
dqever	4		2
dqeWLM	0		1
dqewlmclass	6		2
dqewlmpol	0		2
dqewlmpro1	A	20	3
dqews	0		2
dqewseq	12		2
dqewsid	18		2
dqewsidW	18		2
dqe23psus	B	02	3
PSU_AskSimGDG	55	01	3
PSU_BestStep	55	02	3
PSU_DSlen	58		2
PSU_ExpJCL	55	80	3
PSU_GDGlen	5B		2
PSU_JCLlen	60		2
PSU_Operinfo	55	20	3
PSU_Root	55	04	3
PSU_Rootlen	5E		2
PSU_SimGDG	55	08	3

Name	Hex Offset	Hex Value	Level
PSU_SLlen	56		2
PSU_Suspend	55	40	3
PSU_UserSys	55	10	3

EMP - Event manager parameter area

Name : DCLEMP

Function:
This segment declares an Event Manager Parameter area. This control block is built, initialized, and freed by the Event Manager subtask and by callers of the event handler subroutine.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	432	emp	event manager parameters
0	(0)	CHARACTER	4	empdesc	block descriptor = 'EMP '
4	(4)	CHARACTER	2	empver	block mapping version
6	(6)	BITSTRING	2	empflags	event manager flags
6	(6)	BITSTRING	1	empflag1	event manager flag1
		1...		empjntw	jnt will be replaced
		.1..		empjntc	jnt will be reset
		..1.		emplate	no more delay for cur evt
		...1		empmanr	a manual request
		.... 1...		empmcp	mcp built this block
		.... .1..		empdelop	corresp oper deleted
		.... ..1.		emprstrt	job has been restrted
		1		empb5skp	b5 event skipped
7	(7)	BITSTRING	1	empflag2	event manager flag2
		1...		empjrpfl	JNR=Y triggert
		.1..		emp3Prep	3P event reprocessed
		..1.		empsrasc	sr avail status change
		...1		empjrnfl	JNR=N trigger
		.... 1...		empusev	user event flag
		.... .1..		empsrava	sr avail is yes
		.... ..1.		empnolog	don't log
		1		emp3P	3P event D54A
8	(8)	BITSTRING	4	emptmecb	timer ecb
12	(C)	ADDRESS	4	empmcap	mca address
16	(10)	ADDRESS	4	empdtop	1st delayed exit record
20	(14)	ADDRESS	4	empjntp	jnt buffer address
24	(18)	ADDRESS	4	empoprp	cp oper record address
28	(1C)	ADDRESS	4	empoccp	cp occ record address
32	(20)	ADDRESS	4	empwrkp	cp work record address
36	(24)	ADDRESS	4	empdoap	current doa ptr or 0
40	(28)	ADDRESS	4	empnmmb	normal mode mgr params
44	(2C)	ADDRESS	4	empperf	ev mgr performance
48	(30)	ADDRESS	4	empEdata	address of ext data NVIA
52	(34)	CHARACTER	4	empSTIM	timer ID
56	(38)	UNSIGNED	4	empdur	event duration (sec) d88c
60	(3C)	CHARACTER	8	empstamp	event time stamp
60	(3C)	SIGNED	4	empdate	event date (00yydddf)
64	(40)	SIGNED	4	emptime	event time (secs*100)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
68	(44)	SIGNED	4	empbsize	current plan buffer size
72	(48)	SIGNED	4	empEsize	ext data size NVIA
76	(4C)	ADDRESS	4	empdqep	address of curr DQE
80	(50)	ADDRESS	4	empEQEp	address of eqe
84	(54)	ADDRESS	4	empexrp	ptr to exit record
88	(58)	CHARACTER	10	empclock	event time (yymmddhhmm)
88	(58)	CHARACTER	6	empyymmdd	event date (yymmdd)
94	(5E)	CHARACTER	4	emphhmm	event time (hhmm)
98	(62)	CHARACTER	4	empindex	operation index
98	(62)	UNSIGNED	3	empoccn	occurrence number
101	(65)	UNSIGNED	1	empoprn	operation number
102	(66)	SIGNED	2	empnetid	current net number
104	(68)	CHARACTER	8	empuser	dialog user name
112	(70)	CHARACTER	1	empsign	return code sign
113	(71)	BITSTRING	1	empflag3	event manager flag3 AQBC
		1...		emp23psus	A2A3P susp ETT ANEA BKLC
		.1..		empopcsub	submitted by OPC AQBA
		..1.		empztwe	call ztwe from EM BSUA
		...1		empflush0eve	flush '0' events BRDC
		.... 1...		empTWSNoReleaseOcc	NO release OCC in TWS
		.... .1..		empJ4received	IJ4 processing CCUA
		.... ..1.		empdoaUrg	to be used by MCP ON old DOA was URGENT
		1		empemusr8	ON emusr RC=8
114	(72)	CHARACTER	4	empcode	operation error code
118	(76)	CHARACTER	1	empstat	new operation status
119	(77)	CHARACTER	1	empcom	jes command character
120	(78)	CHARACTER	8	empjobn	current job name
128	(80)	CHARACTER	8	empjnum	current job number
136	(88)	CHARACTER	64	empexit	timer exit
200	(C8)	BITSTRING	1	empflag4	event manager flag4 CLIC
		1...		empNoEtt	NO add a new ETT CLIC
		.1..		empWTO	CPQA
		..1.		empLOGGED	CPQA
		...1		empshadowb	on=ready bind requested
		.... 1...		empParallelOper	Paralle Operation
		.... .1..		empzce2ejob	
		.... ..1.		empPTA3J	DFSA
		1		empPTA3P	DFSC
201	(C9)	BITSTRING	1	empflag5	event manager flag5
		1...		empEMUSR	caller is emusr
		.111 1111		*	caller is emusr
202	(CA)	CHARACTER	10	empdttm	date and time d88a
202	(CA)	CHARACTER	6	empdt	date yymmdd d88a
208	(D0)	SIGNED	4	emptm	time 100th of sec d88a
212	(D4)	SIGNED	4	empcodefix	TWS numeric errc d88a
216	(D8)	ADDRESS	4	empcndp	addr work cond
220	(DC)	CHARACTER	2	*	free
222	(DE)	CHARACTER	8	empettjobn	ETT job name
230	(E6)	CHARACTER	1	empetttyp	ETT type (J or R)
231	(E7)	CHARACTER	44	empettcrit	ETT criteria
275	(113)	CHARACTER	44	empettevnam	ETT full event name
319	(13F)	UNSIGNED	1	empettgdgl	Length of GDG DS root

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
320	(140)	CHARACTER	8	empstrtime	Start time for delay
328	(148)	ADDRESS	4	empMCPtrlP	ptr to cond trl chain-MCP
332	(14C)	ADDRESS	4	empCNDLp	ptr to cond id chain
336	(150)	ADDRESS	4	empcodptr	MSG E031I CSQA
340	(154)	ADDRESS	4	empEmtrlP	ptr to cond trl chain(EM)
344	(158)	ADDRESS	4	empusrfp	addr work usrf
348	(15C)	ADDRESS	4	empxd02p	addr work xd02
352	(160)	ADDRESS	4	empOUCp	ptr to OUC comm area
356	(164)	CHARACTER	76	*	free
432	(1B0)	CHARACTER		empend	end of emp

Cross reference

Name	Hex Offset	Hex Value	Level
emp	0		1
empbsize	44		2
empb5skp	6	01	4
empclock	58		2
empCNDLp	14C		2
empcndp	D8		2
empcode	72		2
empcodefix	D4		2
empcodptr	150		2
empcom	77		2
empdate	3C		3
empdelop	6	04	4
empdesc	0		2
empdoap	24		2
empdoaUrg	71	02	3
empdqep	4C		2
empdt	CA		3
empdtop	10		2
empdttm	CA		2
empdur	38		2
empEdata	30		2
empEmtrlP	154		2
empEMUSR	C9	80	3
empemusr8	71	01	3
empend	1B0		2
empEQEp	50		2
empEsize	48		2
empettcrit	E7		2
empettevnam	113		2

Name	Hex Offset	Hex Value	Level
empettgdgl	13F		2
empettjobn	DE		2
empetttyp	E6		2
empexit	88		2
empexrp	54		2
empflags	6		2
empflag1	6		3
empflag2	7		3
empflag3	71		2
empflag4	C8		2
empflag5	C9		2
empflush0eve	71	10	3
emphhmm	5E		3
empindex	62		2
empjntc	6	40	4
empjntp	14		2
empjntw	6	80	4
empjnum	80		2
empjobn	78		2
empjrnfl	7	10	4
empjrpfl	7	80	4
empJ4received	71	04	3
emplate	6	20	4
empLOGGED	C8	20	3
empmanr	6	10	4
empmcap	C		2
empmcp	6	08	4
empMCPtrlP	148		2
empnetid	66		2
empnmmb	28		2
empNoEtt	C8	80	3
empnolog	7	02	4
empoccn	62		3
empoccp	1C		2
empopcsub	71	40	3
empoprn	65		3
empoprp	18		2
empOUCp	160		2
empParallelOper	C8	08	3
empperf	2C		2
empPTA3J	C8	02	3

Name	Hex Offset	Hex Value	Level
empPTA3P	C8	01	3
emprstrt	6	02	4
empshadowb	C8	10	3
empsign	70		2
empsrasc	7	20	4
empsrava	7	04	4
empstamp	3C		2
empstat	76		2
empSTIM	34		2
empstrtime	140		2
emptime	40		3
emptm	D0		3
emptmecb	8		2
empTWSNoReleaseOcc	71	08	3
empuser	68		2
empusev	7	08	4
empusrfp	158		2
empver	4		2
empwrkp	20		2
empWTO	C8	40	3
empxd02p	15C		2
empyymmdd	58		3
empzce2ejob	C8	04	3
empztwe	71	20	3
emp23psus	71	80	3
emp3P	7	01	4
emp3Prep	7	40	4

EPR - Event data set positions

Name : DCLEPR

Function:
This segment declares the scheduler event data set position record. The third record on the first track of the checkpoint data set is always the EPR for the scheduler host system. The following records are EPR records for remote nodes. The order of the EPR records is the same as the order of the scheduler node names in the NNN record.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	152	EPR	THE SCHEDULER EVENT DS POSITIONS
0	(0)	CHARACTER	4	EPRDESC	BLOCK DESCRIPTOR = 'EPR '
4	(4)	CHARACTER	2	EPRVER	BLOCK MAPPING VERSION

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
6	(6)	BITSTRING	2	*	RESERVED
8	(8)	CHARACTER	8	EPRNODE	NODE NAME (HOST = 0)
16	(10)	SIGNED	4	* (2)	RESERVED (INIT TO ZERO)
24	(18)	CHARACTER	128	EPRTABLE	DATA SET POSITION TABLE
24	(18)	CHARACTER	8	EPRPOS (16)	MAX 16 EVENT DS PER NODE
24	(18)	SIGNED	4	EPRWCY#	WRITE CYCLE NUMBER
28	(1C)	SIGNED	4	EPRREC#	RECORD # OF LAST RECORD
152	(98)	CHARACTER		EPREND	END OF EPR

Cross reference

Name	Hex Offset	Hex Value	Level
EPR	0		1
EPRDESC	0		2
EPREND	98		2
EPRNODE	8		2
EPRPOS	18		3
EPRREC#	1C		4
EPRTABLE	18		2
EPRVER	4		2
EPRWCY#	18		4

EQE - Event queue element

Name : DCLEQE

Function:
This segment declares a scheduler event queue element. EQEs are chained to the event manager queue by event readers and by the host task. EQEs are dequeued by the event manager (host) or by ncf (remote). The EQE is included in TRL24 records.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	120	EQE	EVENT QUEUE ENTRY
0	(0)	CHARACTER	4	EQEDESC	BLOCK DESCRIPTOR = 'EQE '
4	(4)	CHARACTER	2	EQEVER	BLOCK MAPPING VERSION
6	(6)	BITSTRING	2	*	RESERVED
8	(8)	SIGNED	4	EQENODE	NODE NUMBER
12	(C)	SIGNED	4	EQERDRN	EVENT READER NUM IN NODE
16	(10)	CHARACTER	8	EQEPOS	EVENT DS POSITION
16	(10)	SIGNED	4	EQECYC#	WRITE CYCLE NUMBER
20	(14)	SIGNED	4	EQEREC#	RECORD NUMBER IN CYCLE
24	(18)	CHARACTER	8	EQERELDD	SU/RE DDNAME OR ZERO
32	(20)	CHARACTER	80	EQEEXR	EXIT RECORD, SEE DCLEXR
112	(70)	ADDRESS	4	EQECMIP	ADDRESS OF CAT MGT
116	(74)	SIGNED	4	EQECMIL	LENGTH OF STG POINTED TO BY EQECMIP
120	(78)	CHARACTER		EQEEND	END OF EQE

Cross reference

Name	Hex Offset	Hex Value	Level
EQE	0		1
EQECMIL	74		2
EQECMIP	70		2
EQECYC#	10		3
EQEDESC	0		2
EQEEND	78		2
EQEEXR	20		2
EQENODE	8		2
EQEPOS	10		2
EQERDRN	C		2
EQEREC#	14		3
EQERELDD	18		2
EQEVER	4		2

ERR - Error statistics control block

Name : DCLERR

Function:
This segment describes the error statistics control block.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	72	ERR	
0	(0)	ADDRESS	4	ERRFRWD	ERR CB'
4	(4)	CHARACTER	4	ERRCODE	ERROR CODE TOTAL=HEX’0000’
8	(8)	SIGNED	4	ERRDUR	OCC ERROR DURATION
12	(C)	SIGNED	4	ERRTDUR	TOTAL ERROR DURATION
16	(10)	SIGNED	4	ERRRDUR	TOTAL RERUN DURATION
20	(14)	SIGNED	2	ERR#	NO OF ERRORS
22	(16)	CHARACTER	1	ERRST	BIT SWITCHES
				ERREX	ERROR CODE FOR OCC EXIST
				*	NOT USED
23	(17)	CHARACTER	1	*	FREE
24	(18)	CHARACTER	6	ERRWSID	WORKSTATION ID
24	(18)	CHARACTER	4	ERRWSIDN	WORKSTATION ID
28	(1C)	SIGNED	2	ERRWSIDO	WORKSTATION OP NUMBER
30	(1E)	CHARACTER	24	ERROPTXT	OPERATION TEXT
54	(36)	CHARACTER	8	ERRJBNM	JOB NAME
62	(3E)	CHARACTER	10	*	RESERVED
72	(48)	CHARACTER		*	END OF ERROR TABLE

Cross reference

Name	Hex Offset	Hex Value	Level
ERR	0		1
ERR#	14		2
ERRCODE	4		2
ERRDUR	8		2
ERREX	16	80	2
ERRFRWD	0		2
ERRJBNM	36		2
ERROPTXT	1E		2
ERRDUR	10		2
ERRST	16		2
ERRTDUR	C		2
ERRWSID	18		2
ERRWSIDN	18		3
ERRWSIDO	1C		3

ERRS - Error statistics report record in DP

Name : DCLERRS

Function:
Describes layout of error statistics report record. Representing report data on occurrences which had one/more restarted operations before completion. The records have the standard recsize and keysize of DPIN records.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	270	ERSAREA	ERS RECORD
0	(0)	CHARACTER	114	ERSKEY	KEY
0	(0)	SIGNED	2	ERSTYPE	TYPE IS ALWAYS 8
2	(2)	CHARACTER	34	*	
36	(24)	CHARACTER	4	ERSCODE	ERROR CODE
40	(28)	CHARACTER	26	ERSOCC	OCCURANCE IDENTIFICATION
40	(28)	CHARACTER	16	ERSADID	APPLICATION IDENTIFICATION
40	(28)	CHARACTER	1	ERSSUM	SUM PER ERROR: 1 AND
41	(29)	CHARACTER	15	*	TOTAL SUM : 9
56	(38)	CHARACTER	10	ERSOPIA	PLANNED OCC INPUT ARRIVAL
56	(38)	CHARACTER	6	ERSOPIAD	DAY (YYMMDD)
56	(38)	CHARACTER	4	ERSOPIAY	
60	(3C)	CHARACTER	2	ERSOPIAE	(DD)
62	(3E)	CHARACTER	4	ERSOPIAT	TIME (HHMM)
62	(3E)	CHARACTER	2	ERSOPIAH	TIME (HH)
64	(40)	CHARACTER	2	ERSOPIAM	TIME (MM)
66	(42)	CHARACTER	14	*	SPARE
80	(50)	CHARACTER	16	ERSOWID	OWNER ID
96	(60)	CHARACTER	18	*	TO FILL UP THE KEY
114	(72)	CHARACTER	81	ERSDATA	DATA PART OF ERS RECORD

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
114	(72)	CHARACTER	24	ERSOTXT	APPLICATION TEXT
138	(8A)	CHARACTER	1	ERSOPRI	PRIORITY
139	(8B)	CHARACTER	1	*	FREE
140	(8C)	CHARACTER	6	ERSWSID	WSID OF FAILING OPERATION
140	(8C)	CHARACTER	4	ERSWSIDS	WORKSTATION NAME
144	(90)	SIGNED	2	ERSWSIDN	OPERATION NUMBER
146	(92)	CHARACTER	24	ERSOPTXT	OPERATION TEXT
170	(AA)	CHARACTER	8	ERSJBNM	JOBNAME
178	(B2)	CHARACTER	6	ERSEDUR	OPER. ERROR DURATION
178	(B2)	CHARACTER	4	ERSEDURH	TIME (HHHH)
182	(B6)	CHARACTER	2	ERSEDURM	(MM)
184	(B8)	CHARACTER	6	ERSRDUR	OPER. RERUN DURATION
184	(B8)	CHARACTER	4	ERSRDURH	TIME (HHHH)
188	(BC)	CHARACTER	2	ERSRDURM	(MM)
190	(BE)	CHARACTER	5	ERSERRNM	NUMBER OF ERROR
195	(C3)	CHARACTER	75	*	FILLER TO RECSIZE

Cross reference

Name	Hex Offset	Hex Value	Level
ERSADID	28		4
ERSAREA	0		1
ERSCODE	24		3
ERSDATA	72		2
ERSEDUR	B2		3
ERSEDURH	B2		4
ERSEDURM	B6		4
ERSERRNM	BE		3
ERSJBNM	AA		3
ERSKEY	0		2
ERSOCC	28		3
ERSOPIA	38		4
ERSOPIAD	38		5
ERSOPIAE	3C		6
ERSOPIAH	3E		6
ERSOPIAM	40		6
ERSOPIAT	3E		5
ERSOPIAY	38		6
ERSOPRI	8A		3
ERSOPTXT	92		3
ERSOTXT	72		3
ERSOWID	50		3
ERSRDUR	B8		3
ERSRDURH	B8		4

Name	Hex Offset	Hex Value	Level
ERSRDURM	BC		4
ERSSUM	28		5
ERSTYPE	0		3
ERSWSID	8C		3
ERSWSIDN	90		4
ERSWSIDS	8C		4

ESP - Event split record

Name : DCLESP

Function:
This segment declares a split event record. Event types 2, 3J, and 3P can be sent as a split event.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	ESP	ESP head or continuation
0	(0)	CHARACTER	1	ESP_exrtyp	event type = N
1	(1)	CHARACTER	1	ESP_type	type of ESP F = first head ESP N = continued ESP
2	(2)	CHARACTER	1	ESP_syst	contained event type blank === above MUST match DCLEXR ==
3	(3)	CHARACTER	3	ESP_dqetype	actual contained event type
6	(6)	SIGNED	2	ESP_seqF	ESP F-type sequence number
8	(8)	SIGNED	4	ESP_seqN	ESP N-type sequence number
12	(C)	SIGNED	4	ESP_seqT	ESP seq totals
16	(10)	ADDRESS	4	ESP_extptr	address of data buffer NULL
20	(14)	SIGNED	4	ESP_datasize	Total size in buffer when ESP on WRTQ, and for F-type. Data size for N-types.
24	(18)	UNSIGNED	1	ESP_sp	subpool number for ext buffer
25	(19)	CHARACTER	1	ESP_flags	a flag byte
		1...		ESP_app	APP extension buffer
		.111 1111		*	reserved
26	(1A)	CHARACTER	50	ESP_data	actual event data
76	(4C)	CHARACTER	4	ESP_id	event identification (offs 76)
80	(50)	CHARACTER		ESP_xdata0	eXtended data portion plachold

Cross reference

Name	Hex Offset	Hex Value	Level
ESP	0		1
ESP_app	19	80	3
ESP_data	1A		2
ESP_datasize	14		2
ESP_dqetype	3		2
ESP_exrtyp	0		2
ESP_extptr	10		2

Name	Hex Offset	Hex Value	Level
ESP_flags	19		2
ESP_id	4C		2
ESP_seqF	6		2
ESP_seqN	8		2
ESP_seqT	C		2
ESP_sp	18		2
ESP_syst	2		2
ESP_type	1		2
ESP_xdata0	50		2

ETC - ETT criteria record

Name : DCLETC

Function:
This segment declares the Event-triggered-tracking criteria record physically located in the side-information file (EQQSIDS).

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	128	ETC	ETT TRACKING CRITERIA REC
0	(0)	CHARACTER	64	ETCKEY	KEY
0	(0)	CHARACTER	1	ETCTYPE	RECORD TYPE = EVENT TYPE 2 -> JOB, 3 -> RESOURCE
1	(1)	CHARACTER	44	ETCNAME	NAME OF TRIGGERING EVENT
45	(2D)	CHARACTER	19	*	RESERVED
64	(40)	CHARACTER	64	ETCDATA	DATA
64	(40)	CHARACTER	4	ETCDESC	RECORD DESCRIPTOR = 'ETC '
68	(44)	CHARACTER	2	ETCVERS	RECORD VERSION
70	(46)	CHARACTER	1	*	RESERVED
71	(47)	CHARACTER	16	ETCAPPL	CORRESPONDING APPLICATION
87	(57)	CHARACTER	1	ETCJREP	JOB REPLACE: Y=YES, N=NO
88	(58)	CHARACTER	8	ETCLUSER	USER THAT LAST UPDATED
96	(60)	CHARACTER	6	ETCLDATE	DATE OF LAST UPDATE
102	(66)	CHARACTER	4	ETCLTIME	TIME OF LAST UPDATE
106	(6A)	CHARACTER	8	ETCGROUP	AUTHORITY GROUP
114	(72)	CHARACTER	1	ETCDEPR	DEP RESOLUTION: Y=YES,N=NO
115	(73)	CHARACTER	1	ETCASSW	AVAIL STATUS: Y=YES,N=NO
116	(74)	CHARACTER	4	ETCSPARE	RESERVED
120	(78)	CHARACTER	8	ETCLUTS	LAST UPDATE TIMESTAMP
128	(80)	CHARACTER		ETCEND	END OF ETC

Cross reference

Name	Hex Offset	Hex Value	Level
ETC	0		1
ETCAPPL	47		3
ETCASSW	73		3

Name	Hex Offset	Hex Value	Level
ETCDATA	40		2
ETCDEPR	72		3
ETCDESC	40		3
ETCEND	80		2
ETCGROUP	6A		3
ETCJREP	57		3
ETCKEY	0		2
ETCLDATE	60		3
ETCLTIME	66		3
ETCLUSER	58		3
ETCLUTS	78		3
ETCNAME	1		3
ETCSPARE	74		3
ETCTYPE	0		3
ETCVERS	44		3

EVE - Tivoli Workload Scheduler events

```
Name : DCLEVE
Function:
```

This segment declares the Tivoli Workload Scheduler events record.

This layout maps the records of the EQQTWSOU and EQQTWSIN files.

In the EQQTWSOU and EQQTWSIN data sets, the first record is only and always the header event. It is mapped with the EVEHEADER structure. In the EVEHEADER structure, the EVESPECIALOFFSET and EVEJOBSUBMISSION fields are present only in the EQQTWSOU file.

From the second to the last record, the EVERECORD structure is used; it maps the first 24 bytes. From the 25th byte onwards, the record is mapped using the DCLEX0 layout.

Note: If a Symphony renew batch job fails and message EQQ3096E is displayed, check in the TWSOU file that the value stored in the EVENEXTTOREAD field is greater than the value stored in the EVENEXTTOWRITE field. If not, the error is due to a timing problem between the controller and server tasks. Wait for a while before running the Symphony renew batch job again.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	8	EVEGENERIC	GENERIC DS RECORD
0	(0)	CHARACTER	8	EVERECORDKEY	RECORD KEY
0	(0)	CHARACTER	4	EVERECTYPE	RECORD TYPE
4	(4)	SIGNED	4	EVERECNUM	PHYSICAL RECORD NUMBER (0=HDR)
8	(8)	CHARACTER		EVERECDEP	RECORD DEPENDENT DATA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	32	EVEHEADER	HEADER RECORD
0	(0)	SIGNED	4	*(2)	RECORD KEY
8	(8)	SIGNED	4	EVEWRTCYCNUM	LAST WRITE CYCLE NUMBER
12	(C)	SIGNED	4	EVEREADCYCNUM	LAST READ CYCLE NUMBER
16	(10)	SIGNED	4	EVEMAXNUM	MAX RECORD NUMBER EXCL HEADER
20	(14)	SIGNED	4	EVETRCAP	TRACK CAPACITY IN RECORDS
24	(18)	SIGNED	4	EVENEXTTOWRITE	NEXT RECORD TO WRITE
28	(1C)	SIGNED	4	EVENEXTTOREAD	NEXT RECORD TO READ
32	(20)	SIGNED	4	EVESPECIALOFFSET	SPECIAL EVENT OFFSET
36	(24)	CHARACTER	1	EVEJOBSUBMISSION	JOB SUBMISSION (A/D)
37	(25)	CHARACTER	1	EVEDSNAMETYPE	DATASET NAME ID (I/O)
38	(26)	CHARACTER	1	EVEREFRESHOUCDEST	REFRESH OUC DESTS (Y/N)
39	(27)	CHARACTER	1	*	FREE
40	(28)	SIGNED	4	EVEEMSYNCNEXTT OREAD	SYNC RECORD
44	(2C)	SIGNED	4	EVEEMSYNCREADC YCNUM	SYNC CYCLE NUMBER

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	EVERECORD	EVENT RECORD
0	(0)	CHARACTER	24	EVERECORD COMMONPART	EVENT RECORD COMMON PART
0	(0)	SIGNED	4	*(2)	RECORD KEY
8	(8)	SIGNED	4	EVERECWRTCYCNUM	RECORD WRITE CYCLE NUMBER
12	(C)	UNSIGNED	2	EVESEQNUMBER	SEQUENCE NUMBER WITHIN JOBLOG: 0=NO JOBLOG
14	(E)	UNSIGNED	2	EVELOGSIZE	SIZE OF JOBLOG
16	(10)	CHARACTER	8	EVESTMPTOD	TIME OF DAY IN MILLISECS
24	(18)	CHARACTER	*	EVERECORDDATA	RECORD DEPENDENT DATA

Cross reference

Name	Hex Offset	Hex Value	Level
EVEEMSYNCNEXTTOREAD	28		2
EVEEMSYNCREADCYCNUM	2C		2
EVEGENERIC	0		1
EVEHEADER	0		1
EVEJOBSUBMISSION	24		2
EVELOGSIZE	E		3
EVEMAXNUM	10		2
EVENEXTTOREAD	1C		2
EVENEXTTOWRITE	18		2
EVEREADCYCNUM	C		2
EVERECDEP	8		3

Name	Hex Offset	Hex Value	Level
EVERECNUM	4		3
EVERECORD	0		1
EVERECORDCOMMONPART	0		2
EVERECORDDATA	18		2
EVERECORDKEY	0		2
EVERECTYPE	0		3
EVERECWRTCYCNUM	8		3
EVESEQNUMBER	C		3
EVESPECIALOFFSET	20		2
EVESTMPTOD	10		3
EVETRCAP	14		2
EVEWRTCYCNUM	8		2

EVT - Event and SUBREL data set record

Name : DCLEVT

Function:
This segment declares the layout of the event data set and submit release data set.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	820	EVT	EVENT RECORD
0	(0)	CHARACTER	20	EVTKEY	EVENT RECORD KEY
0	(0)	SIGNED	4	EVTREC#	PHYSICAL RECORD NO (0=HDR)
4	(4)	SIGNED	4	*	USED ONLY IN EV DS 2 HDR
4	(4)	SIGNED	2	EVT2REC#	EV DS 2 READ RECORD NUMBER
6	(6)	SIGNED	2	EVT2CYC#	EV DS 2 READ CYCLE NUMBER
8	(8)	SIGNED	4	EVTCYC#	WRITE CYCLE NO 0|1|... = 0 FOR HEADER
12	(C)	SIGNED	4	EVTMAX#	MAX #RCDS EXCL HDR = 0 FOR NON-HEADER
16	(10)	SIGNED	4	EVTTRCAP	TRACK CAPACITY IN RCDS = 0 FOR NON-HEADER
20	(14)	CHARACTER	80	EVTEXR (10)	EXIT RECORD (SEE DCLEXR)
20	(14)	CHARACTER	80	EVTSUR	SUBMIT RECORD (SEE DCLSUR)
20	(14)	CHARACTER	80	EVTREL	RELEASE RECORD(SEE DCLREL) BLANK FOR HEADER & CYCLE0

Event data set header record

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	*	*	EVDS HEADER RECORD KEY
0	(0)	CHARACTER	*	EVTCKPT	CKPT RECORD (REC 0)
0	(0)	BITSTRING	1	EVTFLAGS	RESERVED
1	(1)	UNSIGNED	1	EVTRSSEQ#	REQUEST #
2	(2)	CHARACTER	6	EVTCKPTE (*)	CHECKPOINT ENTRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
2	(2)	CHARACTER	4	EVTWSNM	WORKSTATION NAME
6	(6)	UNSIGNED	2	EVTSSEQ#	CURRENT SUBMIT SEQ#

Submit/release header record

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	20	*	SU/RE HEADER RECORD KEY
0	(0)	CHARACTER	20	EVTSRH	SU/RE HEADER RECORD KEY
0	(0)	SIGNED	4	*	RECORD NUMBER = 0
4	(4)	SIGNED	4	EVTLRCYC	LATEST READ CYCLE NUMBER
8	(8)	SIGNED	4	EVTLRREC	LATEST READ RECORD NUMBER
12	(C)	SIGNED	4	*	MAX #RCDS EXCL HDR
16	(10)	SIGNED	4	*	TRACK CAPACITY IN RCDS

Submit/release record

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	4	*	SUB/REL HEADER REC EXT
0	(0)	BITSTRING	4	EVTSRFLG	EXTENSION TO HEADER
		1...		EVTID	OK TO SEND ID EVENT
0	(0)	BITSTRING	3	*	RESERVED

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	20	*	SU/RE RECORD KEY
0	(0)	CHARACTER	20	EVTSR	SU/RE RECORD KEY
0	(0)	SIGNED	4	*	RECORD NUMBER
4	(4)	UNSIGNED	2	EVTSEQ#B	SEQ. NO WITHIN JCLBATCH 0 FOR RELEASE RECORD
6	(6)	UNSIGNED	2	EVTSIZEB	SIZE OF JCL BATCH 0 FOR RELEASE RECORD
8	(8)	SIGNED	4	*	WRITE CYCLE NUMBER
12	(C)	CHARACTER	3	EVTDQETYP	DQE DATA TYPE
15	(F)	CHARACTER	5	EVTSTMP	EVENT TIME STAMP
15	(F)	UNSIGNED	1	EVTSTMPY	YEAR 0-99
16	(10)	UNSIGNED	1	EVTSTMPM	MONTH 1-12
17	(11)	UNSIGNED	1	EVTSTMPD	DAY 1-31
18	(12)	SIGNED	2	EVTSTMPT	TIME IN MINUTES IN HH * 60 + MM FORM

Cross reference

Name	Hex Offset	Hex Value	Level
EVT	0		1
EVTCKPT	0		2
EVTCKPTE	2		3
EVTCYC#	8		3

Name	Hex Offset	Hex Value	Level
EVTDQETYP	C		3
EVTEXR	14		2
EVTFLAGS	0		3
EVTID	0	80	3
EVTKEY	0		2
EVTLRCYC	4		3
EVTLRREC	8		3
EVTMAX#	C		3
EVTREC#	0		3
EVTREL	14		4
EVTRSSEQ#	1		3
EVTSEQ#B	4		3
EVTSIZEB	6		3
EVTSR	0		2
EVTSRFLG	0		2
EVTSRH	0		2
EVTSSEQ#	6		4
EVTSTMP	F		3
EVTSTMPD	11		4
EVTSTMPM	10		4
EVTSTMPT	12		4
EVTSTMPY	F		4
EVTSUR	14		3
EVTTRCAP	10		3
EVTWSNM	2		4
EVT2CYC#	6		4
EVT2REC#	4		4

EX0 - Topology In event data

Name: DCLEX0

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	ex0	topology In event data the length of this field must be minor or equal to Exr block length
0	(0)	CHARACTER	1	ex0Syst	'0' type event MAESTRO
1	(1)	CHARACTER	1	ex0Type	Type indicator
2	(2)	CHARACTER	22	*	reserved
24	(18)	CHARACTER	52	ex0InputEvent	INPUT Event in EVE CB
76	(4C)	CHARACTER	4	ex0Opcid	event id 'EQQ0'

Input Event Definition
The LENGTH of this block must be minor or equal to

everlen minus eveRecordCommonPart
Moreover must be equal to LENGTH of ex0InputEvent now 52
then the structure must be the following

Offsets Dec	 Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	52	ex0Event	
0	(0)	SIGNED	4	ex0Cdat	creation date (00yydddf)
4	(4)	SIGNED	4	ex0Ctim	creation time (sec*100)
8	(8)	SIGNED	2	ex0Gmtof	gmt offset (minutes)
10	(A)	CHARACTER	1	ex0flags	BRDA
		1...		ex0suspd	suspended event BRDA (mirror to exrsuspd) BRDA
		.111 1111		*	BRDA
11	(B)	CHARACTER	1	*	reserved BRDC
12	(C)	CHARACTER	40	ex0BODY	This is Input event body To map this field see the following Event Dependent data

Output Event Definition
The LENGTH of this block must be minor or equal to
everlen minus eveRecordCommonPart now 1202496
then the structure must be the following
dcl
$BOGa
The LENGTH of this block must be minor or equal to $BOGa
everlen minus eveRecordCommonPart now 16024136 $BOGa00
then the structure must be the following $BOGa
$BOGa

Offsets Dec	 Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	136	ex0Tws	tplgy Out event data BOGa0
0	(0)	CHARACTER	136	ex0TwsEvent	OUTPUT event in EVE CB BOGa0
0	(0)	CHARACTER	4	ex0TwsCommon	Common Part BOGa0
0	(0)	CHARACTER	1	ex0TwsSyst	'0' type event MAESTRO BOGa0
1	(1)	CHARACTER	1	ex0TwsType	indicator BOGa0
2	(2)	CHARACTER	1	ex0TwsExType	Extended type indicator BOGa0
3	(3)	CHARACTER	1	*	free BOGa
4	(4)	CHARACTER	132	ex0TwsBody	BOGa0 Topology Output event body To map this field see the following Event Dependent data

Event Dependent Data

Workstation Event mapping for ex0Body and ex0TwsBody

Offsets Dec	 Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	28	ex0WS	workstation event body
0	(0)	CHARACTER	4	ex0WSID	WS name
4	(4)	CHARACTER	1	ex0WSlnk	WS link status L linked U unlinked
5	(5)	CHARACTER	1	ex0WSsta	WS status BSQC A active O offline
6	(6)	CHARACTER	1	ex0WSflk	WS full link CHRCBSQA
7	(7)	CHARACTER	1	ex0WStyp	node type CHRA
8	(8)	CHARACTER	16	ex0WSdomain	domain name CHRA F fully linked BSQA U unfully linked BSQA
24	(18)	CHARACTER	1	ex0WSflags	
		1...		ex0WSChgLimit	Change WS Limit

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
		.111 1111		*	
25	(19)	CHARACTER	1	*	avoid compiler msg
26	(1A)	UNSIGNED	2	ex0WSLimit	New WS limit

Workstation Event switch dommgr mapping for ex0Body ex0TwsBody

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	20	ex0WSMGR	workstation event body
0	(0)	CHARACTER	4	ex0WSN	WS name
4	(4)	CHARACTER	16	ex0WSDOM	WS Domain manager

Operation Event mapping for ex0Body only

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	40	ex0JOB	tracker event record
0	(0)	CHARACTER	8	ex0JOBocctk	OCC token
8	(8)	SIGNED	2	ex0JOBnum	OPR number
10	(A)	CHARACTER	4	ex0JOBws	WS name
14	(E)	CHARACTER	1	ex0JOBsta	OPR status E Error S Started C Complited D Deleted U Undecided T Terminated
15	(F)	CHARACTER	1	ex0JOBflags	
		1111 111.		*	reserved not used for compatibil. with 8.1.0 events
		1		ex0JOBisRecJob	event is for recovery job
16	(10)	CHARACTER	24	ex0JOBbuff	JOB event buffer To map this field see the ex0JOBmaps and ex0JOBmapt structures

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	24	ex0JOBmaps	JOB event buffer map for status STARTED
0	(0)	CHARACTER	4	*	for align with old event
4	(4)	CHARACTER	8	ex0JobId	job number string format
12	(C)	SIGNED	4	ex0JobIdNum	job number numeric
16	(10)	CHARACTER	8	*	free

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	25	ex0JOBmapt	JOB event buffer map for status TERMPROMPT25a
0	(0)	SIGNED	4	ex0JOBerrc	error code
4	(4)	CHARACTER	4	ex0JobRecWs	recov job WS
8	(8)	UNSIGNED	2	ex0JobMsgNo	recov prmpt num 25aGDMc
10	(A)	SIGNED	2	ex0JobMsgLen	recov prmpt len
12	(C)	CHARACTER	1	ex0JobRecType	recovery type S - Stop C - Continue R - Rerun
13	(D)	CHARACTER	1	ex0JobPrmptAns	prompt answer YN
14	(E)	BITSTRING	1	ex0JobRecCond	recovery conditions
		1...		ex0JobRecovery	has been recovered
		.1..		ex0JobRecJob	has recovery job
		..1.		ex0JobRecMsg	wait on message
		...1		ex0JobFail	on if status=E Fail

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		 1111		*	free
15	(F)	CHARACTER	10	*	free

Change Operation Event mapping for ex0TwsBody Only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	110	ex0TwsJOB	tracker event record
0	(0)	CHARACTER	8	ex0TJBOccToken	Occurrence Token
8	(8)	SIGNED	2	ex0TJBOpNumber	Operation Number
10	(A)	CHARACTER	16	ex0TJBAdName	Application Name
26	(1A)	CHARACTER	6	ex0TJBWS	WS Information
26	(1A)	CHARACTER	4	ex0TJBWsName	WS name
30	(1E)	CHARACTER	1	ex0TJBWsType	WS Type
31	(1F)	CHARACTER	1	*	free
32	(20)	BITSTRING	2	ex0TJBCommands	Commands Flags
		1...		ex0TJBExecute	Execute Job
34	(22)	BITSTRING	2	ex0TJBActionsFlags	Actions Flags
		1...		ex0TJBChgStatus	Status Changed
		.1..		ex0TJBChgIA	Input Arrival Changed
		..1.		ex0TJBChgUntilTm	Until Time Changed CANC
		...1		ex0TJBChgHoldRel	HoldRelease Changed
		.... 1...		ex0TJBChgTimeDep	Time Dependency Changed
		.... .1..		ex0TJBChgSuppLate	Suppr if Late Changed
		.... ..1.		ex0TJBChgPriority	Priority Changed
		1		ex0TJBChgDeadline	Deadline Changed
35	(23)	1...		ex0TJBChgAutoSub	Auto submit changed BSUA
		.1..		ex0TJBChgNop	Nop changed BSUA
36	(24)	CHARACTER	42	ex0TJBActions	Actions
36	(24)	CHARACTER	2	ex0TJBStatus	Operation Status
36	(24)	CHARACTER	1	ex0TJBOldStatus	new status
37	(25)	CHARACTER	1	ex0TJBNewStatus	old status
38	(26)	CHARACTER	10	ex0TJBIA	Input Arrival
38	(26)	CHARACTER	6	ex0TJBIADate	
44	(2C)	CHARACTER	4	ex0TJBIATime	
48	(30)	CHARACTER	12	ex0TJBUntilTm	Until Time CANC
48	(30)	CHARACTER	6	ex0TJBUntDate	CANC
54	(36)	CHARACTER	2	*	
56	(38)	SIGNED	4	ex0TJBUntTime	CANC
60	(3C)	CHARACTER	2	*	
		1...		ex0TJBHoldRel	Oldrelease Job
		.1..		ex0TJBTimeDep	Time Dependency
		..1.		ex0TJBSuppLate	Suppr if Late
		...1		ex0TJBJnFormat	TWS jobname format BOGa
		.... 1...		ex0TJBAutoSub	Auto submit BSUA
		.... .1..		ex0TJBNop	NoppedUn-Nopped BSUA
		.... ..1.		ex0TJBCScript	Centralized BSUA
62	(3E)	SIGNED	2	ex0TJBPriority	Operation Priority
64	(40)	SIGNED	2	ex0TJBOpNumberEx	Operation Number Ext
66	(42)	CHARACTER	2	ex0TJBExtStatus	Extended status for FT23a
66	(42)	BITSTRING	1	ex0TJBOldStFlg	Old flags
		1...		ex0TJBOFStart	
		.1..		ex0TJBOFSending	
		..1.		ex0TJBOFSuber	
		...1		ex0TJBOFErfail	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.... 1...		ex0TJBOFRecPrmp	
		.... .1..		ex0TJBOFRecJob	
		.... ..1.		ex0TJBOFRecRun	
		1		*	
67	(43)	BITSTRING	1	ex0TJBNewStFlg	New flags
		1...		ex0TJBNFStart	
		.1..		ex0TJBNFSending	
		..1.		ex0TJBNFSuber	
		...1		ex0TJBNFErfail	
		.... 1...		ex0TJBNFRecPrmp	
		.... .1..		ex0TJBNFRecJob	
		.... ..1.		ex0TJBNFRecRun	
		1		*	
68	(44)	CHARACTER	6	ex0TJBDeadlDate	job Deadline Date
74	(4A)	CHARACTER	4	ex0TJBDeadlTime	job Deadline Time2AaBOGc
78	(4E)	CHARACTER	32	ex0TJBJobInfo	Job Info BOGa

Add Operation Event mapping for ex0TwsBody Only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	112	ex0TwsJOBAdd	Add Job event record
0	(0)	CHARACTER	8	ex0TJAOccToken	Occurrence Token
8	(8)	SIGNED	2	ex0TJAOpNumber	Operation Number
10	(A)	SIGNED	2	ex0TJAOpNumberEx	Operation Number Ext
12	(C)	CHARACTER	16	ex0TJAAdName	Application Name
28	(1C)	CHARACTER	5	ex0TJAWS	WS Information
28	(1C)	CHARACTER	4	ex0TJAWsName	WS name
32	(20)	CHARACTER	1	ex0TJAWsType	WS Type
33	(21)	CHARACTER	1	ex0TJAStatus	Operation Status
34	(22)	CHARACTER	10	ex0TJAIA	Input Arrival
34	(22)	CHARACTER	6	ex0TJAIADate	
40	(28)	CHARACTER	4	ex0TJAIATime	
44	(2C)	CHARACTER	12	ex0TJAUntilTm	Until Time CANC
44	(2C)	CHARACTER	6	ex0TJAUntDate	CANC
50	(32)	CHARACTER	2	*	
52	(34)	SIGNED	4	ex0TJAUntTime	CANC
56	(38)	CHARACTER	2	*	
		1...		ex0TJAHoldRel	Oldrelease Job
		.1..		ex0TJATimeDep	Time Dependency
		..1.		ex0TJASuppLate	Suppr if Late
		...1		ex0TJACScript	Centralized Script
		.... 1...		ex0TJAJnFormat	TWS jobname format BOGa
		.... .1..		ex0TJAAutoSub	auto submit option BSUA
58	(3A)	SIGNED	2	ex0TJAPriority	Operation Priority
60	(3C)	CHARACTER	6	ex0TJADeadlDate	Job Deadline Date
66	(42)	CHARACTER	4	ex0TJADeadlTime	Job Deadline Time
70	(46)	CHARACTER	6	ex0TJAOccDLDate	Occ Deadline Date
76	(4C)	CHARACTER	4	ex0TJAOccDLTime	Occ Deadline Time2AaBOGc
80	(50)	CHARACTER	32	ex0TJAJobInfo	Job Info BOGa Job Name

Release Dependency Event mapping for ex0TwsBody only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	68	ex0JRD	release dep event 33a
0	(0)	CHARACTER	1	ex0JRDflags	flags 33a
		1...		ex0JRDJnFormat	TWS jobname format 33a
1	(1)	CHARACTER	3	*	free 33a
4	(4)	CHARACTER	16	ex0JRDOccName	Occurrence Name 33a
20	(14)	CHARACTER	8	ex0JRDOccToken	Occurrence Token 33a
28	(1C)	SIGNED	2	ex0JRDOpNumber	Operation Number 33a
30	(1E)	SIGNED	2	ex0JRDOpNumberEx	Oper Number Extd 33a
32	(20)	CHARACTER	32	ex0JRDJobInfo	Job Info 33a
64	(40)	CHARACTER	4	ex0JRDOpWs	Job Cpu 33a

Change Occurrence Event mapping for ex0TwsBody Only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	44	ex0OccChg	Chg Occ event record
0	(0)	CHARACTER	8	ex0OccChgToken	Occurrence Token
8	(8)	BITSTRING	2	ex0OccChgChgActionsFlags	Actions Flags
		1...		ex0OccChgChgHoldRel	HoldRelease Changed
		.1..		ex0OccChgChgDeadline	Dealine Changed
		..1.		ex0OccChgChgStatus	Change Status BZYa
10	(A)	CHARACTER	13	ex0OccChgChgActions	
10	(A)	CHARACTER	1	*	
		1...		ex0OccChgHoldRel	Oldrelease Occurrence
11	(B)	CHARACTER	6	ex0OccChgDeadLDate	Deadline date
17	(11)	CHARACTER	4	ex0OccChgDeadLTime	Deadline time
21	(15)	CHARACTER	2	ex0OccChgStatus	Change Status BZYa
21	(15)	CHARACTER	1	ex0OccChgOldStatus	Old Status BZYa
22	(16)	CHARACTER	1	ex0OccChgNewStatus	New Status 35cBZYa
23	(17)	CHARACTER	1	*	free
24	(18)	CHARACTER	16	ex0OccChgName	Occurrence Name
40	(28)	SIGNED	2	ex0OccChgIADate	Input Arrival Date
42	(2A)	SIGNED	2	ex0OccChgIATime	Input Arrival Time

Add Occurrence Event mapping for ex0TwsBody Only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	45	ex0OccAdd	Add Occ event record
0	(0)	CHARACTER	8	ex0OccAddToken	Occurrence Token
8	(8)	CHARACTER	16	ex0OccAddId	Occurrence Identifier
24	(18)	CHARACTER	6	ex0OccAddIaD	Occurrence IA Date
30	(1E)	CHARACTER	4	ex0OccAddIaT	Occurrence IA Time
34	(22)	CHARACTER	1	*	
		1...		ex0OccAddHoldRel	Oldrelease Occurrence
35	(23)	CHARACTER	6	ex0OccAddDeadLDate	Deadline date
41	(29)	CHARACTER	4	ex0OccAddDeadLTime	Deadline time

Define a user for a JOB mapping for ex0TwsBody Only

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	47	ex0JUsr	User event record
0	(0)	CHARACTER	47	ex0JUsrName	user name

Job Definition event mapping for ex0TwsBody Only

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	2	ex0JDef	Job Definition
0	(0)	BITSTRING	2	ex0JDefFlags	
		1...		ex0JDefIsCommand	Is a Command
		.1..		ex0JDefInteract	Is Interactive
		..1.		ex0JDefContinue	Recov Continue
		...1		ex0JDefRerun	Recov Rerun
0	(0)	BITSTRING	1	*	free
2	(2)	CHARACTER		ex0JDefBody	Job Definition Body

Prompt Definition event mapping for ex0TwsBody Only

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	2	ex0PDef	Job Definition
0	(0)	BITSTRING	2	ex0PDefFlags	
0	(0)	BITSTRING	2	*	free
2	(2)	CHARACTER		ex0PDefBody	Job Definition Body

Dependency Event mapping for ex0TwsBody Only

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	132	ex0Dep	tracker event record
0	(0)	CHARACTER	1	ex0DepAction	Dependency Action
1	(1)	CHARACTER	2	*	free
3	(3)	CHARACTER	1	ex0DepFlags	Actions Flags
		1...		ex0DepPendingPred	Pending predecessor
		.1..		ex0DepJnFormat	TWS jobname format BOGa
		..1.		ex0DepJnFormatPred	TWS jobname format BOGa of the predecessor BOGa
4	(4)	CHARACTER	16	ex0DepOccName	Occurrence Name
20	(14)	CHARACTER	8	ex0DepOccToken	Occurrence Token
28	(1C)	SIGNED	2	ex0DepIADate	Input Arrival Date
30	(1E)	SIGNED	2	ex0DepIATime	Input Arrival Time
32	(20)	SIGNED	2	ex0DepOpNumber	Operation Number
34	(22)	SIGNED	2	ex0DepOpNumberEx	Operation Number Ext
36	(24)	CHARACTER	16	ex0DepOccNamePred	Pred Occurrence Name
52	(34)	CHARACTER	8	ex0DepOccTokenPred	Pred Occurrence Token
60	(3C)	SIGNED	2	ex0DepIADatePred	Input Arriv. Date Pred
62	(3E)	SIGNED	2	ex0DepIATimePred	Input Arriv. Time Pred
64	(40)	SIGNED	2	ex0DepOpNumberPred	Pred Operation Number
66	(42)	SIGNED	2	ex0DepOpNumberExPred	Oper Number Ext 03aBOGc
68	(44)	CHARACTER	32	ex0DepJobInfo	Job Info BOGa
100	(64)	CHARACTER	32	ex0DepJobInfoPred	Pred Job Info BOGa

Special Resources Event mapping for ex0TwsBody only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	72	ex0SR	tracker event record
0	(0)	CHARACTER	1	ex0SRAction	spec res action
1	(1)	CHARACTER	1	ex0SRflags	flags BOGa
		1...		ex0SRJnFormat	TWS jobname format BOGa
2	(2)	CHARACTER	2	*	free 0CaBOGc
4	(4)	CHARACTER	16	ex0SROccName	Occurrence Name
20	(14)	CHARACTER	8	ex0SROccToken	Occurrence Token
28	(1C)	SIGNED	2	ex0SROpNumber	Operation Number
30	(1E)	SIGNED	2	ex0SROpNumberEx	Oper Number Extd 0CaBOGc
32	(20)	CHARACTER	32	ex0SRJobInfo	Job Info BOGa
64	(40)	CHARACTER	4	ex0SROpWs	Job Cpu 35aBZYa
68	(44)	SIGNED	2	ex0SRIADate	Input Arrival Date
70	(46)	SIGNED	2	ex0SRIATime	Input Arrival Time

Job Log Event mapping for ex0Body and ex0TwsBody

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	88	ex0Log	tracker event record
0	(0)	CHARACTER	8	ex0LogOccToken	Occurrence Token
8	(8)	SIGNED	2	ex0LogOpNumber	Operation Number
10	(A)	SIGNED	2	ex0LogOpNumberEx	Operation Number Ext
12	(C)	CHARACTER	16	ex0LogOccName	Occurrence Name
28	(1C)	CHARACTER	8	ex0LogUser	userid to be notified
36	(24)	CHARACTER	1	ex0LogAction	Retrieve result RCE
37	(25)	BITSTRING	1	*	
		1...		ex0LogRecJob	Recovery job log
		.1..		ex0LogJnFormat	TWS jobname format BOGa
		..11 1111		*	free 27c25aBOGc
38	(26)	CHARACTER	2	*	free
40	(28)	CHARACTER	4	ex0LogWsName	workstation name
44	(2C)	SIGNED	2	ex0LogStartTime	job start time
46	(2E)	SIGNED	2	ex0LogStartDate	job start date
48	(30)	SIGNED	4	ex0LogJobId	jobnumber numeric27aBOGc
52	(34)	CHARACTER	32	ex0LogJobInfo	Job Info 35cBOGa
84	(54)	SIGNED	2	ex0LogIADate	Input Arrival Date
86	(56)	SIGNED	2	ex0LogIATime	Input Arrival Time

JCL Event mapping for ex0Body and ex0TwsBody

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	92	ex0Jcl	tracker event record 0
0	(0)	CHARACTER	8	ex0JclOccToken	Occurrence Token
8	(8)	UNSIGNED	1	ex0JclOpNumber	Operation Number
9	(9)	UNSIGNED	1	ex0JclOpNumberEx	Operation Number Ext
10	(A)	CHARACTER	1	ex0JclAction	Retrieve result RCE
11	(B)	CHARACTER	1	ex0JCLflags	for alignement
		1...		ex0JCLJnFormat	TWS jobname format BOGa
12	(C)	CHARACTER	16	ex0JclOccName	Occurrence Name
28	(1C)	CHARACTER	8	ex0JclUser	user

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
36	(24)	CHARACTER	4	ex0JclWsName	WS Name
40	(28)	SIGNED	4	ex0JclLine	JCL lines number
44	(2C)	SIGNED	4	ex0JclSize	JCL size
48	(30)	CHARACTER	8	ex0JclFileName	JCL file name 22aBOGc
56	(38)	CHARACTER	32	ex0JclJobInfo	Job Info 35cBOGa
88	(58)	SIGNED	2	ex0JclIADate	Input Arrival Date
90	(5A)	SIGNED	2	ex0JclIATime	Input Arrival Time

Prompt Event mapping for ex0twsBody only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	5	ex0PRM	prompt event record 25a0
0	(0)	UNSIGNED	4	ex0PRMMsgNum	Prmpt message number
4	(4)	CHARACTER	1	ex0PRMstatus	reply YN

Syncronyzation Event mapping for ex0Body and ex0TwsBody

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	8	ex0SYN	Synchronization event
0	(0)	CHARACTER	3	*	
3	(3)	CHARACTER	1	ex0SyncType	Synchronization Type
4	(4)	SIGNED	4	ex0SymRunNum	Symphony run number

Set Time Event mapping for ex0TwsBody only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	8	ex0T0	Set time event 0
0	(0)	CHARACTER	4	ex0T0Sid	System Id
4	(4)	SIGNED	4	ex0T0gmtof	GMT offset

ActivateDeactivate job submission mapping for ex0twsBody only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	1	ex0JSub	2Ba0
0	(0)	CHARACTER	1	ex0JSubAction	ActivateDeactivate

Kill jobrecovery job Event mapping for ex0TwsBody Only

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	ex0Kill	tracker event record
0	(0)	CHARACTER	8	ex0KillOccToken	Occurrence Token
8	(8)	CHARACTER	16	ex0KillAdName	Application Name
24	(18)	CHARACTER	10	ex0KillIA	Input Arrival

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
24	(18)	CHARACTER	6	ex0KillIAdate	
30	(1E)	CHARACTER	4	ex0KillIAtime	
34	(22)	CHARACTER	1	ex0KillFlags	Flags
		1...		ex0KillRecjob	kill for recovery job
		.1..		ex0KillJnFormat	TWS jobname format
		..1.		ex0KillCScript	Centralized
35	(23)	CHARACTER	1	*	Free
36	(24)	CHARACTER	4	ex0KillWS	WS name
40	(28)	SIGNED	4	ex0KillJobid	jobid
44	(2C)	SIGNED	2	ex0KillOpNumber	Operation Number
46	(2E)	SIGNED	2	ex0KillOpNumberEx	Oper Number Extd
48	(30)	CHARACTER	32	ex0KillJobInfo	Job Info

Cross reference

Name	Hex Offset	Hex Value	Level
ex0	0		1
ex0BODY	C		2
ex0Cdat	0		2
ex0Ctim	4		2
ex0Dep	0		1
ex0DepAction	0		2
ex0DepFlags	3		2
ex0DepIADate	1C		2
ex0DepIADatePred	3C		2
ex0DepIATime	1E		2
ex0DepIATimePred	3E		2
ex0DepJnFormat	3	40	3
ex0DepJnFormatPred	3	20	3
ex0DepJobInfo	44		2
ex0DepJobInfoPred	64		2
ex0DepOccName	4		2
ex0DepOccNamePred	24		2
ex0DepOccToken	14		2
ex0DepOccTokenPred	34		2
ex0DepOpNumber	20		2
ex0DepOpNumberEx	22		2
ex0DepOpNumberExPred	42		2
ex0DepOpNumberPred	40		2
ex0DepPendingPred	3	80	3
ex0Event	0		1
ex0flags	A		2
ex0Gmtof	8		2
ex0InputEvent	18		2

Name	Hex Offset	Hex Value	Level
ex0Jcl	0		1
ex0JclAction	A		2
ex0JclFileName	30		2
ex0JCLflags	B		2
ex0JclIADate	58		2
ex0JclIATime	5A		2
ex0JCLJnFormat	B	80	3
ex0JclJobInfo	38		2
ex0JclLine	28		2
ex0JclOccName	C		2
ex0JclOccToken	0		2
ex0JclOpNumber	8		2
ex0JclOpNumberEx	9		2
ex0JclSize	2C		2
ex0JclUser	1C		2
ex0JclWsName	24		2
ex0JDef	0		1
ex0JDefBody	2		2
ex0JDefContinue	0	20	3
ex0JDefFlags	0		2
ex0JDefInteract	0	40	3
ex0JDefIsCommand	0	80	3
ex0JDefRerun	0	10	3
ex0JOB	0		1
ex0JOBbuff	10		2
ex0JOBerrc	0		2
ex0JobFail	E	10	3
ex0JOBflags	F		2
ex0JobId	4		2
ex0JobIdNum	C		2
ex0JOBisRecJob	F	01	3
ex0JOBmaps	0		1
ex0JOBmapt	0		1
ex0JobMsgLen	A		2
ex0JobMsgNo	8		2
ex0JOBnum	8		2
ex0JOBocctk	0		2
ex0JobPrmptAns	D		2
ex0JobRecCond	E		2
ex0JobRecJob	E	40	3
ex0JobRecMsg	E	20	3

Name	Hex Offset	Hex Value	Level
ex0JobRecovery	E	80	3
ex0JobRecType	C		2
ex0JobRecWs	4		2
ex0JOBsta	E		2
ex0JOBws	A		2
ex0JRD	0		1
ex0JRDflags	0		2
ex0JRDJnFormat	0	80	3
ex0JRDJobInfo	20		2
ex0JRDOccName	4		2
ex0JRDOccToken	14		2
ex0JRDOpNumber	1C		2
ex0JRDOpNumberEx	1E		2
ex0JRDOpWs	40		2
ex0JSub	0		1
ex0JSubAction	0		2
ex0JUsr	0		1
ex0JUsrName	0		2
ex0Kill	0		1
ex0KillAdName	8		2
ex0KillCScript	22	20	3
ex0KillFlags	22		2
ex0KillIA	18		2
ex0KillIAdate	18		3
ex0KillIAtime	1E		3
ex0KillJnFormat	22	40	3
ex0KillJobid	28		2
ex0KillJobInfo	30		2
ex0KillOccToken	0		2
ex0KillOpNumber	2C		2
ex0KillOpNumberEx	2E		2
ex0KillRecjob	22	80	3
ex0KillWS	24		2
ex0Log	0		1
ex0LogAction	24		2
ex0LogIADate	54		2
ex0LogIATime	56		2
ex0LogJnFormat	25	40	3
ex0LogJobId	30		2
ex0LogJobInfo	34		2
ex0LogOccName	C		2

Name	Hex Offset	Hex Value	Level
ex0LogOccToken	0		2
ex0LogOpNumber	8		2
ex0LogOpNumberEx	A		2
ex0LogRecJob	25	80	3
ex0LogStartDate	2E		2
ex0LogStartTime	2C		2
ex0LogUser	1C		2
ex0LogWsName	28		2
ex0OccAdd	0		1
ex0OccAddDeadLDate	23		2
ex0OccAddDeadLTime	29		2
ex0OccAddHoldRel	22	80	3
ex0OccAddIaD	18		2
ex0OccAddIaT	1E		2
ex0OccAddId	8		2
ex0OccAddToken	0		2
ex0OccChg	0		1
ex0OccChgChgActions	A		2
ex0OccChgChgActionsFlags	8		2
ex0OccChgChgDeadline	8	40	3
ex0OccChgChgHoldRel	8	80	3
ex0OccChgChgStatus	8	20	3
ex0OccChgDeadLDate	B		3
ex0OccChgDeadLTime	11		3
ex0OccChgHoldRel	A	80	4
ex0OccChgIADate	28		2
ex0OccChgIATime	2A		2
ex0OccChgName	18		2
ex0OccChgNewStatus	16		4
ex0OccChgOldStatus	15		4
ex0OccChgStatus	15		3
ex0OccChgToken	0		2
ex0Opcid	4C		2
ex0PDef	0		1
ex0PDefBody	2		2
ex0PDefFlags	0		2
ex0PRM	0		1
ex0PRMMsgNum	0		2
ex0PRMstatus	4		2
ex0SR	0		1
ex0SRAction	0		2

Name	Hex Offset	Hex Value	Level
ex0SRflags	1		2
ex0SRIADate	44		2
ex0SRIATime	46		2
ex0SRJnFormat	1	80	3
ex0SRJobInfo	20		2
ex0SROccName	4		2
ex0SROccToken	14		2
ex0SROpNumber	1C		2
ex0SROpNumberEx	1E		2
ex0SROpWs	40		2
ex0suspd	A	80	3
ex0SymRunNum	4		2
ex0SYN	0		1
ex0SyncType	3		2
ex0Syst	0		2
ex0TJAAdName	C		2
ex0TJAAutoSub	38	04	3
ex0TJACScript	38	10	3
ex0TJADeadlDate	3C		2
ex0TJADeadlTime	42		2
ex0TJAHoldRel	38	80	3
ex0TJAIA	22		2
ex0TJAIADate	22		3
ex0TJAIATime	28		3
ex0TJAJnFormat	38	08	3
ex0TJAJobInfo	50		2
ex0TJAOccDLDate	46		2
ex0TJAOccDLTime	4C		2
ex0TJAOccToken	0		2
ex0TJAOpNumber	8		2
ex0TJAOpNumberEx	A		2
ex0TJAPriority	3A		2
ex0TJAStatus	21		2
ex0TJASuppLate	38	20	3
ex0TJATimeDep	38	40	3
ex0TJAUntDate	2C		3
ex0TJAUntilTm	2C		2
ex0TJAUntTime	34		3
ex0TJAWS	1C		2
ex0TJAWsName	1C		3
ex0TJAWsType	20		3

Name	Hex Offset	Hex Value	Level
ex0TJBActions	24		2
ex0TJBActionsFlags	22		2
ex0TJBAdName	A		2
ex0TJBAutoSub	3C	08	4
ex0TJBChgAutoSub	23	80	3
ex0TJBChgDeadline	22	01	3
ex0TJBChgHoldRel	22	10	3
ex0TJBChgIA	22	40	3
ex0TJBChgNop	23	40	3
ex0TJBChgPriority	22	02	3
ex0TJBChgStatus	22	80	3
ex0TJBChgSuppLate	22	04	3
ex0TJBChgTimeDep	22	08	3
ex0TJBChgUntilTm	22	20	3
ex0TJBCommands	20		2
ex0TJBCScript	3C	02	4
ex0TJBDeadlDate	44		3
ex0TJBDeadlTime	4A		3
ex0TJBExecute	20	80	3
ex0TJBExtStatus	42		3
ex0TJBHoldRel	3C	80	4
ex0TJBIA	26		3
ex0TJBIADate	26		4
ex0TJBIATime	2C		4
ex0TJBJnFormat	3C	10	4
ex0TJBJobInfo	4E		2
ex0TJBNewStatus	25		4
ex0TJBNewStFlg	43		4
ex0TJBNFErfail	43	10	5
ex0TJBNFRecJob	43	04	5
ex0TJBNFRecPrmp	43	08	5
ex0TJBNFRecRun	43	02	5
ex0TJBNFSending	43	40	5
ex0TJBNFStart	43	80	5
ex0TJBNFSuber	43	20	5
ex0TJBNop	3C	04	4
ex0TJBOccToken	0		2
ex0TJBOFErfail	42	10	5
ex0TJBOFRecJob	42	04	5
ex0TJBOFRecPrmp	42	08	5
ex0TJBOFRecRun	42	02	5

Name	Hex Offset	Hex Value	Level
ex0TJBOFSending	42	40	5
ex0TJBOFStart	42	80	5
ex0TJBOFSuber	42	20	5
ex0TJBOldStatus	24		4
ex0TJBOldStFlg	42		4
ex0TJBOpNumber	8		2
ex0TJBOpNumberEx	40		3
ex0TJBPriority	3E		3
ex0TJBStatus	24		3
ex0TJBSuppLate	3C	20	4
ex0TJBTimeDep	3C	40	4
ex0TJBUntDate	30		4
ex0TJBUntilTm	30		3
ex0TJBUntTime	38		4
ex0TJBWS	1A		2
ex0TJBWsName	1A		3
ex0TJBWsType	1E		3
ex0Tws	0		1
ex0TwsBody	4		3
ex0TwsCommon	0		3
ex0TwsEvent	0		2
ex0TwsExType	2		4
ex0TwsJOB	0		1
ex0TwsJOBAdd	0		1
ex0TwsSyst	0		4
ex0TwsType	1		4
ex0Type	1		2
ex0T0	0		1
ex0T0gmtof	4		2
ex0T0Sid	0		2
ex0WS	0		1
ex0WSChgLimit	18	80	3
ex0WSDOM	4		2
ex0WSdomain	8		2
ex0WSflags	18		2
ex0WSflk	6		2
ex0WSID	0		2
ex0WSLimit	1A		2
ex0WSlnk	4		2
ex0WSMGR	0		1
ex0WSN	0		2

Name	Hex Offset	Hex Value	Level
ex0WSsta	5		2
ex0WStyp	7		2

EX1 - BULKDISC EVENT

Name : DCLEX1

Function:
This segment declares an event record for the BULKDISC command.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	EX1	BULKDISC EVENT
0	(0)	CHARACTER	1	EX1TYPE	1 FOR BULKDISC EVENT
1	(1)	CHARACTER	3	*	RESERVED
4	(4)	CHARACTER	8	EX1USER	NAME OF UPDATING USER
12	(C)	CHARACTER	4	*	RESERVED
16	(10)	SIGNED	4	EX1CDAT	CREATION DATE (00YYDDDF)
20	(14)	SIGNED	1	EX1CTIM	CREATION TIME (SEC* 100)
24	(18)	CHARACTER	52	*	RESERVED
76	(4C)	CHARACTER	4	EX1OPCID	EVENT ID EQQX

Cross reference

Name	Hex Offset	Hex Value	Level
EX1	0		1
EX1CDAT	10		2
EX1CTIM	14		2
EX1OPCID	4C		2
EX1TYPE	0		2
EX1USER	4		2

EX2 - z/OS Agent Event record

Name : DCLEX2

Function:
This record ....

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	EX2	zAgent event record
0	(0)	CHARACTER	1	EX2SYST	'2' event
1	(1)	CHARACTER	1	EX2TYPE	type event
2	(2)	UNSIGNED	1	EX2FLAGS	flags
		1...		EX2SUSPD	event suspended
		.1..		EX2NUMERR	on: err code is numeric

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
		..1.		EX2DUMMY	on: dummy start date
		...1 1111		*	reserved
3	(3)	CHARACTER	1	*	reserved
4	(4)	CHARACTER	16	EX2TOKENS	
4	(4)	CHARACTER	8	EX2OCCTOKEN	occurrence token
12	(C)	CHARACTER	8	EX2SUBTOKEN	submission token
20	(14)	CHARACTER	8	EX2JOBNUM	job number
28	(1C)	CHARACTER	1	EX2JCLI	parse error Y/N
29	(1D)	CHARACTER	1	EX2UNKNOWN	unknown result Y/N
30	(1E)	SIGNED	2	EX2OPNUM	oper num
32	(20)	CHARACTER	4	EX2WSNAME	WS name
36	(24)	CHARACTER	8	EX2JOB	job name
44	(2C)	SIGNED	4	EX2DAT2	date 2
48	(30)	SIGNED	4	EX2CDAT	creation date (00yydddf)
52	(34)	SIGNED	4	EX2CTIM	creation time (sec*100)
56	(38)	SIGNED	4	EX2TIM2	time 2
60	(3C)	SIGNED	4	EX2ERRN	error code (integer)
60	(3C)	CHARACTER	4	EX2ERRC	used by Z remote engine
64	(40)	CHARACTER	4	EX2SUBSYS	Controller subsystem name
68	(44)	SIGNED	4	EX2ADUR	actual duration
72	(48)	CHARACTER	4	*	free
76	(4C)	CHARACTER	4	EX2OPCID	event id 'EQQx'

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	80	EX2XD	XD actions layout ('X')
0	(0)	CHARACTER	1	EX2XDSYST	'2' event
1	(1)	CHARACTER	1	EX2XDTYPE	type event ('X')
2	(2)	UNSIGNED	1	EX2XDFLAGS	flags
		1...		EX2XDSUSPD	event suspended
		.1..		EX2XDPCP	ON: req for waitcp
		..1.		EX2XDPLT	ON: req for waitlt
		...1		EX2XDDPSET	ON: req for waitlt
		 1111		*	reserved
3	(3)	CHARACTER	1	EX2XDSUBT	D-notif. ended/remove XD F-failed to notify F-manage pending XD
4	(4)	CHARACTER	8	EX2XDTOKEN	xd99 key !_ keys used
12	(C)	CHARACTER	56	EX2XDALIAS	xd02 key ! alternatively
68	(44)	CHARACTER	8	*	free
76	(4C)	CHARACTER	4	EX2XDOPCID	event id 'EQQx'

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	80	EX2JOBINFO	2I and 2B layout
0	(0)	CHARACTER	1	EX2ISYST	'2' event
1	(1)	CHARACTER	1	EX2ITYPE	type event
2	(2)	UNSIGNED	1	EX2IFLAGS	flags
		1...		EX2ISUSPD	event suspended
		.111 1111		*	reserved
3	(3)	CHARACTER	1	*	reserved
4	(4)	CHARACTER	16	EX2ITOKENS	
4	(4)	CHARACTER	8	EX2IOCCTOKEN	occurrence token

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
12	(C)	CHARACTER	8	EX2ISUBTOKEN	submission token
20	(14)	CHARACTER	8	EX2IJOBNAME	matched job name
28	(1C)	SIGNED	2	EX2IOPNUM	oper num
30	(1E)	CHARACTER	2	*	free
32	(20)	CHARACTER	4	EX2IWSNAME	matched WS name
36	(24)	CHARACTER	4	EX2ISUBSYS	Controller subsystem name
40	(28)	SIGNED	4	EX2IIAD	matched ia date
44	(2C)	SIGNED	4	EX2IIAT	matched ia time
48	(30)	CHARACTER	28	*	free
76	(4C)	CHARACTER	4	EX2IOPCID	event id 'EQQx'

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EX2AGENTINFO	agent info event record
0	(0)	CHARACTER	1	EX2AISYST	'2' event
1	(1)	CHARACTER	1	EX2AITYPE	type event
2	(2)	UNSIGNED	1	EX2AIFLAGS	flags
		1...		EX2AISUSPD	event suspended
		.111 1111		*	reserved
3	(3)	CHARACTER	1	EX2AINEWSTAT	A | O (active or offline)
4	(4)	SIGNED	4	EX2AIFPLEVEL	fix pack level
8	(8)	CHARACTER	8	EX2AIDEST	destination name
16	(10)	CHARACTER	8	EX2AIVERSION	agent version
24	(18)	CHARACTER	16	EX2AIOSNAME	op. system
40	(28)	CHARACTER	16	EX2AIOSLEVEL	op. system level
56	(38)	SIGNED	2	EX2AIDUMMYCHG	first active
58	(3A)	SIGNED	2	*	reserved
60	(3C)	CHARACTER	16	*	reserved
76	(4C)	CHARACTER	4	EX2AIOPCID	event id 'EQQx'

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EX2JOBLOG	agent info event record
0	(0)	CHARACTER	1	EX2JLSYST	'2' event
1	(1)	CHARACTER	1	EX2JLTYPE	type event
2	(2)	UNSIGNED	1	EX2JLFLAGS	flags
		1...		EX2JLSUSPD	event suspended
		.111 1111		*	reserved
3	(3)	CHARACTER	1	EX2JLSTATUS	job log req status C|E
4	(4)	CHARACTER	8	EX2JLOCCTOKEN	occurrence token
12	(C)	CHARACTER	8	EX2JLUSER	job request user
20	(14)	ADDRESS	4	EX2JLBUFPTR	job log buffer
24	(18)	ADDRESS	4	EX2JLBUFLEN	job log buffer length
28	(1C)	SIGNED	2	EX2JLOPNUM	operation number
30	(1E)	CHARACTER	2	*	reserved
32	(20)	SIGNED	4	EX2JLLENGTH	joblog length
36	(24)	CHARACTER	40	*	reserved
76	(4C)	CHARACTER	4	EX2JLOPCID	event id 'EQQx'

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	100	EX2BIND	subscription (bind) req

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	CHARACTER	1	EX2BSYST	'2' event
1	(1)	CHARACTER	1	EX2BTYPE	'Y' type event
2	(2)	UNSIGNED	1	EX2BFLAGS	flags
		1...		EX2BSUSPD	event suspended
		.111 1111		*	reserved
3	(3)	CHARACTER	1	*	free
4	(4)	CHARACTER	18	EX2BTOMATCH	
4	(4)	CHARACTER	16	EX2BADID	application id
20	(14)	SIGNED	2	EX2BOPNUM	operation number
22	(16)	SIGNED	2	EX2BDESTNUM	number of backup dests
24	(18)	CHARACTER	8	EX2BIA	input arrival for match
24	(18)	SIGNED	4	EX2BIAD	date (00yydddf)
28	(1C)	SIGNED	4	EX2BIAT	time (sec*100)
32	(20)	CHARACTER	56	EX2BALIAS	alias
88	(58)	CHARACTER	8	EX2BDEST1	first destination name
96	(60)	ADDRESS	4	EX2BXPTR	ptr to extended part

C code allocates only one buffer: the extended part always follows the fixed part. Used ptr instead of char(0) for an easier addition of new fields in the event.

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	*	EX2BEXT	subscription ext part
0	(0)	CHARACTER	8	EX2BDESTS(*)	backup destinations
0	(0)	CHARACTER	8	EX2DESTNM	dest name

Events constants

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
1		CHARACTER	2	EX2EVENT	z-Centric Event Type
1		CHARACTER	P	EX2SPEVE	submit - post executed
1		CHARACTER	T	EX2STEVE	submit - post failed
1		CHARACTER	s	EX2SBEVE	submitted
1		CHARACTER	S	EX2SEEVE	start execution
1		CHARACTER	C	EX2JCEVE	job completed successfully
1		CHARACTER	K	EX2JKEVE	job killed
1		CHARACTER	E	EX2EEEVE	job ended in error (exec)
1		CHARACTER	e	EX2EREVE	recovery in progress
1		CHARACTER	F	EX2EFEVE	submission failed
1		CHARACTER	U	EX2UUEVE	job status unknown
1		CHARACTER	R	EX2RREVE	reset job status to ready
1		CHARACTER	B	EX2BFEVE	bind failed event
1		CHARACTER	I	EX2BIEVE	bind ok event
1		CHARACTER	Y	EX2BREVE	bind request
1		CHARACTER	O	EX2OOEVE	online/offline event
1		CHARACTER	L	EX2JLEVE	joblog retrieval
1		CHARACTER	X	EX2XDEVE	manage XD
1		CHARACTER	C	EX2JLCOM	joblog retrieval complete
1		CHARACTER	E	EX2JLERR	joblog retrieval error
1		CHARACTER	S	EX2JLSUS	joblog retrieval suspend
1		CHARACTER	P	EX2XDPND	check pending

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
1		CHARACTER	K	EX2XDKLL	kill bind request
1		CHARACTER	F	EX2XDERR	failed to notify mark XD failed
1		CHARACTER	D	EX2XDDEL	notification completed remove XD
1		CHARACTER	A	EX2_DESTACT	destination status active
1		CHARACTER	O	EX2_DESTOFF	destination status offline
1		CHARACTER	P	EX2BRPRE	matching criteria closest preceding
1		CHARACTER	I	EX2BRINT	matching criteria interval

Cross reference

Name	Hex Offset	Hex Value	Level
EX2	0		1
EX2ADUR	44		2
EX2AGENTINFO	0		1
EX2AIDEST	8		2
EX2AIDUMMYCHG	38		2
EX2AIFLAGS	2		2
EX2AIFPLEVEL	4		2
EX2AINEWSTAT	3		2
EX2AIOPCID	4C		2
EX2AIOSLEVEL	28		2
EX2AIOSNAME	18		2
EX2AISUSPD	2	80	3
EX2AISYST	0		2
EX2AITYPE	1		2
EX2AIVERSION	10		2
EX2BADID	4		3
EX2BALIAS	20		2
EX2BDESTNUM	16		2
EX2BDESTS	0		2
EX2BDEST1	58		2
EX2BEXT	0		1
EX2BFLAGS	2		2
EX2BIA	18		2
EX2BIAD	18		3
EX2BIAT	1C		3
EX2BIND	0		1
EX2BOPNUM	14		3
EX2BSUSPD	2	80	3
EX2BSYST	0		2
EX2BTOMATCH	4		2
EX2BTYPE	1		2

Name	Hex Offset	Hex Value	Level
EX2BXPTR	60		2
EX2CDAT	30		2
EX2CTIM	34		2
EX2DAT2	2C		2
EX2DESTNM	0		3
EX2DUMMY	2	20	3
EX2ERRC	3C		3
EX2ERRN	3C		2
EX2FLAGS	2		2
EX2IFLAGS	2		2
EX2IIAD	28		2
EX2IIAT	2C		2
EX2IJOBNAME	14		2
EX2IOCCTOKEN	4		3
EX2IOPCID	4C		2
EX2IOPNUM	1C		2
EX2ISUBSYS	24		2
EX2ISUBTOKEN	C		3
EX2ISUSPD	2	80	3
EX2ISYST	0		2
EX2ITOKENS	4		2
EX2ITYPE	1		2
EX2IWSNAME	20		2
EX2JCLI	1C		2
EX2JLBUFLEN	18		2
EX2JLBUFPTR	14		2
EX2JLFLAGS	2		2
EX2JLLENGTH	20		2
EX2JLOCCTOKEN	4		2
EX2JLOPCID	4C		2
EX2JLOPNUM	1C		2
EX2JLSTATUS	3		2
EX2JLSUSPD	2	80	3
EX2JLSYST	0		2
EX2JLTYPE	1		2
EX2JLUSER	C		2
EX2JOB	24		2
EX2JOBINFO	0		1
EX2JOBLOG	0		1
EX2JOBNUM	14		2
EX2NUMERR	2	40	3

Name	Hex Offset	Hex Value	Level
EX2OCCTOKEN	4		3
EX2OPCID	4C		2
EX2OPNUM	1E		2
EX2SUBSYS	40		2
EX2SUBTOKEN	C		3
EX2SUSPD	2	80	3
EX2SYST	0		2
EX2TIM2	38		2
EX2TOKENS	4		2
EX2TYPE	1		2
EX2UNKNOWN	1D		2
EX2WSNAME	20		2
EX2XD	0		1
EX2XDALIAS	C		2
EX2XDDPSET	2	10	3
EX2XDFLAGS	2		2
EX2XDOPCID	4C		2
EX2XDPCP	2	40	3
EX2XDPLT	2	20	3
EX2XDSUBT	3		2
EX2XDSUSPD	2	80	3
EX2XDSYST	0		2
EX2XDTOKEN	4		2
EX2XDTYPE	1		2

EXE - Backup event

Name : DCLEXE

Function:
This segment declares a resource data set backup event record. These event records are built via the EQQEVPGM and EQQEVCMD programs, and they are passed to the event writer via CSA buffers, and written to an event data set as part of an event record by the event writer.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	EXE	BACKUP EVENT
0	(0)	CHARACTER	1	EXETYPE	'E' FOR BACKUP EVENT
1	(1)	CHARACTER	2	EXERES	'CP' = CURRENT PLAN 'JS' = JCL REPOSITORY DSN
3	(3)	CHARACTER	1	*	RESERVED
4	(4)	CHARACTER	8	EXEUSER	NAME OF UPDATING USER
12	(C)	CHARACTER	4	*	RESERVED
16	(10)	SIGNED	4	EXECDAT	CREATION DATE (00YYDDDF)
20	(14)	SIGNED	4	EXECTIM	CREATION TIME (SEC-100)

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
24	(18)	CHARACTER	52	*	RESERVED
76	(4C)	CHARACTER	4	EXEOPCID	EVENT ID 'EQQX'

Cross reference

Name	Hex Offset	Hex Value	Level
EXE	0		1
EXECDAT	10		2
EXECTIM	14		2
EXEOPCID	4C		2
EXERES	1		2
EXETYPE	0		2
EXEUSER	4		2

EXF - Joblog retrieval event

Name : DCLEXF

Function:
This segment declares the joblog retrieval events for the Archiver.
The events describe the following stages of joblog retrieval:

FI	initiation
F0	started
F2	ended

Joblog Retrieval events are all created on the Controller side:
FI is created by GS task, when request to FL is sent
F0 is created by FL task, when request from GS is received
F2 is created by FL task, when retrieve result must be sent

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	80	EXF	JOBLOG-RETRIEVAL EVENT
0	(0)	CHARACTER	1	EXFEVTYP	EVENT TYPE = F
1	(1)	CHARACTER	1	EXFTYPE	TYPE OF INITIATED PROCESS I = JOBLOG RETRIEVAL INITIATED 0 = JOBLOG RETRIEVAL STARTED 2 = JOBLOG RETRIEVAL ENDED
2	(2)	CHARACTER	2	*	RESERVED
4	(4)	CHARACTER	16	EXFADID	APPLICATION ID
20	(14)	CHARACTER	10	EXFIA	INPUT ARRIVAL
20	(14)	CHARACTER	6	EXFIAD	.. IA DATE YYMMDD
26	(1A)	CHARACTER	4	EXFIAT	.. IA TIME HHMM
30	(1E)	UNSIGNED	2	EXFFLAGS	NOT USED
32	(20)	CHARACTER	30	EXFTYPES	DECLARATION OF TYPES
62	(3E)	SIGNED	2	EXFGMTOF	GMT OFFSET IN MINUTES
64	(40)	CHARACTER	8	EXFTIMES	REQUEST TIME STAMP AS PASSED FROM THE CONTROLLER WORD BNDRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
64	(40)	SIGNED	4	EXFTDATE	DATE FORMAT (00YYDDDF)
68	(44)	SIGNED	4	EXFTTIME	TIME FORMAT (SECS*100)
72	(48)	CHARACTER	4	EXFSSNM	NOT USED
76	(4C)	CHARACTER	4	EXFID	EVENT IDENTIFICATION 'EQQN'

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
32	(20)	STRUCTURE	30	EXFTYPEI	FI=JOBLOG-RETRIEVAL INITIATION
32	(20)	UNSIGNED	2	EXFIOPR#	OPERATION NUMBER
34	(22)	CHARACTER	8	EXFIUSER	TSO USERID
42	(2A)	CHARACTER	4	EXFIWS	WORK STATION NAME
46	(2E)	CHARACTER	8	EXFIJNM	JOBNAME
54	(36)	CHARACTER	8	EXFIJID	JOB ID

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
32	(20)	STRUCTURE	30	EXFTYPE0	F0=JOBLOG-RETRIEVAL STARTED
32	(20)	SIGNED	4	EXF0OCC#	OCCURRENCE NUMBER
36	(24)	SIGNED	4	EXF0OPR#	OPERATION NUMBER
40	(28)	CHARACTER	8	EXF0JNM	JOBNAME
48	(30)	CHARACTER	8	EXF0JID	JOB ID
56	(38)	CHARACTER	6	*	FREE

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
32	(20)	STRUCTURE	30	EXFTYPE2	F2=JOBLOG-RETRIEVAL ENDED
32	(20)	SIGNED	4	EXF2OPR#	OPERATION NUMBER
36	(24)	CHARACTER	1	EXF2STAT	FINAL RESULT OF JOBLOG RETRIEVAL C = COMPLETED NORMALLY E = PROCESSING ENDED IN ERROR
37	(25)	CHARACTER	1	EXF2XST	' ' = DEFAULT VALUE
38	(26)	CHARACTER	8	EXF2USER	USERID TO BE NOTIFIED
46	(2E)	CHARACTER	1	EXF2TYPE	REQUEST TYPE J = JOBLOG ONLY A = ALL IN MSGCLASS
47	(2F)	CHARACTER	4	EXF2WSN	WORK STATION NAME
51	(33)	CHARACTER	11	*	RESERVED

Cross reference

Name	Hex Offset	Hex Value	Level
EXF	0		1
EXFADID	4		2
EXFEVTYP	0		2
EXFFLAGS	1E		2
EXFGMTOF	3E		2
EXFIA	14		2
EXFIAD	14		3

Name	Hex Offset	Hex Value	Level
EXFIAT	1A		3
EXFID	4C		2
EXFIJID	36		2
EXFIJNM	2E		2
EXFIOPR#	20		2
EXFIUSER	22		2
EXFIWS	2A		2
EXFSSNM	48		2
EXFTDATE	40		3
EXFTIMES	40		2
EXFTTIME	44		3
EXFTYPE	1		2
EXFTYPEI	20		1
EXFTYPES	20		2
EXFTYPE0	20		1
EXFTYPE2	20		1
EXF0JID	30		2
EXF0JNM	28		2
EXF0OCC#	20		2
EXF0OPR#	24		2
EXF2OPR#	20		2
EXF2STAT	24		2
EXF2TYPE	2E		2
EXF2USER	26		2
EXF2WSN	2F		2
EXF2XST	25		2

EXH - History

Name : DCLEXH

Function:
This segment declares the creation of cp14 (joblog status) events when an "old" occurrence is selected from the DB2 History data base and is added to the current plan. The events are written to JT, but not to the event data set.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EXH	HISTORY ADD EVENT
0	(0)	CHARACTER	1	EXHEVTYP	EVENT TYPE = H
1	(1)	CHARACTER	1	EXHTYPE	SUBTYPE .. D DATASET RECORD ADD .. J JOBLOG STATUS ADD
2	(2)	CHARACTER	2	*	RESERVED
4	(4)	CHARACTER	16	EXHADID	APPLICATION ID

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
20	(14)	CHARACTER	10	EXHIA	INPUT ARRIVAL
20	(14)	CHARACTER	6	EXHIAD	.. IA DATE YYMMDD
26	(1A)	CHARACTER	4	EXHIAT	.. IA TIME HHMM
30	(1E)	UNSIGNED	2	EXHFLAGS	FLAG AREA
30	(1E)	BITSTRING	2	*	RESERVED
32	(20)	SIGNED	2	EXHIOPR#	OPERATION NUMBER
34	(22)	CHARACTER	8	EXHIUSER	TSO USERID
42	(2A)	CHARACTER	4	EXHIWS	WORK STATION NAME
46	(2E)	CHARACTER	8	EXHIJNM	JOBNAME
54	(36)	CHARACTER	8	EXHIJID	job id
62	(3E)	SIGNED	2	EXHGMTOF	GMT OFFSET IN MINUTES
64	(40)	CHARACTER	8	EXHTIMES	ADD OCCASION TIME STAMP WORD BNDRY
64	(40)	SIGNED	4	EXHTDATE	DATE FORMAT (00YYDDDF)
68	(44)	SIGNED	4	EXHTTIME	TIME FORMAT (SECS*100)
72	(48)	CHARACTER	4	EXHSSNM	ORIGINATING SS NAME
76	(4C)	CHARACTER	4	EXHID	EVENT IDENTIFICATION 'EQQN'

Cross reference

Name	Hex Offset	Hex Value	Level
EXH	0		1
EXHADID	4		2
EXHEVTYP	0		2
EXHFLAGS	1E		2
EXHGMTOF	3E		2
EXHIA	14		2
EXHIAD	14		3
EXHIAT	1A		3
EXHID	4C		2
EXHIJID	36		2
EXHIJNM	2E		2
EXHIOPR#	20		2
EXHIUSER	22		2
EXHIWS	2A		2
EXHSSNM	48		2
EXHTDATE	40		3
EXHTIMES	40		2
EXHTTIME	44		3
EXHTYPE	1		2

EXI - Initialization event

Name : DCLEXI

Function:
This segment declares an initialization event. These event records used to coordinate and checkpoint submit requests between the controller and the trackers.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EXI	INITIALIZATION EVENT
0	(0)	CHARACTER	1	EXIEVTYP	EVENT TYPE = I
1	(1)	CHARACTER	3	EXITYPE	TYPE OF INITIATED PROCESS 'J0 ' = SUBMIT SEQ# EVENT J1 = JCL SUBMIT J2 = JCL STARTED TASK WTO = WTO MESSAGE REL = JOB RELEASE OSI = OSI TYPE SUBMIT WLM = WLM JOB PROMOTION
4	(4)	CHARACTER	4	EXIWSID	WORK STATION ID
8	(8)	CHARACTER	16	EXIADID	APPLICATION NAME
8	(8)	CHARACTER	16	EXIJ4SCHENV	SCHEDULING ENVIRONMENT NAME (J4)
8	(8)	CHARACTER	8	EXIRCRE	REQUEST CREATE TIME (J0)
8	(8)	SIGNED	4	EXIRCRED	DATE (FORMAT: 00YYDDDF)
12	(C)	SIGNED	4	EXIRCRET	TIME (FORMAT: SECS*100)
16	(10)	CHARACTER	8	EXIOCCTOK	OCCURRENCE TOKEN
16	(10)	UNSIGNED	2	EXIWSEQ#	SUBMIT SEQ# FROM WS
18	(12)	CHARACTER	6	*	
24	(18)	CHARACTER	10	EXIOCIA	OCCURRENCE INPUT ARRIVAL
24	(18)	CHARACTER	8	EXIJ4OCCTOK	OCCURRENCE TOKEN (J4)
32	(20)	SIGNED	2	EXIJ4SPLEX	SYSPLEX ID (J4)
34	(22)	SIGNED	2	EXIOPNUM	OPERATION NUMBER
34	(24)	SIGNED	4	EXIWLMRC	WLM PROMOTION REQUEST RC
36	(24)	BITSTRING	4	EXIFLAGS	FLAG AREA
36	(24)	BITSTRING	1	EXIFLAG1	FLAG BYTE 1
		1...		EXIFAIL	ON=INIT OF PROCESS FAILED
		.1..		EXISFAIL	ON=SUBMIT FAIL
		..1.		EXINOEDP	ON=NO EDP UPDATES FOR THIS EV
		...1		EXIWLMAH	ON= JOB ALREADY IN HI PERFORM CLASS
		.... 1...		EXISEUND	ON= SCHEDULING ENVIRONMENT NOT DEFINED
		.... .1..		EXISENOAVA	ON= SCHEDULING ENVIRONMENT NOT AVAILABLE
		.... ..1.		EXISENOJPL	ON= SCHEDULING ENVIRONMENT NOT AVAILABLE AT JESPLEX
		1		*	RESERVED
37	(25)	BITSTRING	1	EXIFLAG2	FLAG BYTE 2
		1...		EXI2FAIL	INIT OF PROCESS FAILED
		.1..		EXI2PLTE	POSSIBLE LATE I EVENT
		..1.		EXI2BAD	STC FUNCTION NOT ACTIVE
		...1		EXI2NJSS	NOT JES MANAGED SUBSYSTEM
		.... 1...		EXI2FJES	STC FAILED JES INPUT SERVICE
		.... .1..		EXI2NCF	NCF DESTINATION NOT VALID
		11		*	RESERVED
38	(26)	BITSTRING	1	EXIFLAG3	FLAG BYTE 3
		1...		EXI3OOS	SUBMIT OUT-OF-SEQUENCE

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.1..		EXI3NCKP	CHECKPOINT NOT ACTIVE
		..1.		EXI31ST	1ST TIME SWITCH
		...1		EXI3MISR#	SEQ REQUEST MISMATCH
		.... 1...		EXI3ALLOK	VALID SUBMIT SEQUENCE OF WA VS. SU
		.... .1..		*	RESERVED
		.... ..1.		EXI3BADJ0	BAD J0 EVENT (OR CP)
		1		EXI3RDOP	ERROR READ CP OPER
39	(27)	BITSTRING	1	EXIFLAG4	FLAG BYTE 4
		1...		EXI4RDJN	ERROR READ CP JNT
		.1..		EXI4NSUB	SUB04 IS BLANK
		..1.		EXI4JOB#	JES04 IS BLANK
		...1		*	RESERVED
		.... 1...		EXI4NSOP	CPLSUBOP IS SET TO OFF
		.... .11.		*	RESERVED
		1		EXI4SUCC	SUCCESSFUL SUBMIT
40	(28)	CHARACTER	8	EXIJOBN	JOB/STC NAME
40	(28)	UNSIGNED	2	EXIWORK#	CURRENT CATCHUP VALUE(J0)
42	(2A)	UNSIGNED	2	EXIESEQW	CURRENT EVDS VALUE (J0)
44	(2C)	UNSIGNED	2	EXIEDOA#	CURRENT DOA VALUE (J0)
46	(2E)	UNSIGNED	2	EXIJASID	JOB/STC ASID
48	(30)	CHARACTER	8	EXIJOBID	JOB/STC NUMBER (J1, J2 ONLY)
48	(30)	CHARACTER	8	EXIWLMCLS	WLM CLASS (WLM ONLY)
48	(30)	CHARACTER	1	EXIEVLOG	RESERVED (WLM)
49	(31)	CHARACTER	7	*	RESERVED (WLM)
56	(38)	SIGNED	2	EXIGMTOF	GMT OFFSET, IN MINUTES
58	(3A)	SIGNED	2	EXIWLMRSN	WLM PROMOTION REQUEST, REASON CODE
60	(3C)	CHARACTER	8	EXICREAT	EVENT RECORD CREATION TIME WORD BNDRY
60	(3C)	SIGNED	4	EXIDATE	DATE FORMAT (00YYDDDF)
64	(40)	SIGNED	4	EXITIME	TIME FORMAT (SECS*100)
68	(44)	CHARACTER	4	EXISSNM	DQE ORIGINATING SUBSYS NAME
72	(48)	UNSIGNED	2	EXIESEQ#	CURRENT EVDS SEQ# (J0)
74	(4A)	UNSIGNED	2	EXISSEQ#	CURRENT SUBMIT SEQ# (J0)
74	(4A)	UNSIGNED	1	EXIRSEQ#	CURRENT REQUEST SEQ# (J0)
75	(4B)	CHARACTER	1	*	RESERVED
76	(4C)	CHARACTER	4	EXIID	EVENT ID 'EQQX'

Cross reference

Name	Hex Offset	Hex Value	Level
EXI	0		1
EXIADID	8		2
EXICREAT	3C		2
EXIDATE	3C		3
EXIEDOA#	2C		3
EXIESEQ#	48		2
EXIESEQW	2A		3

Name	Hex Offset	Hex Value	Level
EXIEVLOG	30		4
EXIEVTYP	0		2
EXIFAIL	24	80	5
EXIFLAGS	24		3
EXIFLAG1	24		4
EXIFLAG2	25		4
EXIFLAG3	26		4
EXIFLAG4	27		4
EXIGMTOF	38		2
EXIID	4C		2
EXIJASID	2E		3
EXIJOBID	30		2
EXIJOBN	28		2
EXIJ4OCCTOK	18		3
EXIJ4SCHENV	8		3
EXIJ4SPLEX	20		3
EXINOEDP	24	20	5
EXIOCCTOK	10		3
EXIOCIA	18		2
EXIOPNUM	22		2
EXIRCRE	8		3
EXIRCRED	8		4
EXIRCRET	C		4
EXIRSEQ#	4A		3
EXISENOAVA	24	04	5
EXISENOJPL	24	2	5
EXISENOVA	24	4	4
EXISEUND	24	8	5
EXISFAIL	24	40	5
EXISSEQ#	4A		2
EXISSNM	44		2
EXITIME	40		3
EXITYPE	1		2
EXIWLMAH	24	10	5
EXIWLMCLS	30		3
EXIWLMRC	24		2
EXIWLMRSN	3A		2
EXIWORK#	28		3
EXIWSEQ#	10		3
EXIWSID	4		2
EXI2BAD	25	20	5

Name	Hex Offset	Hex Value	Level
EXI2FAIL	25	80	5
EXI2FJES	25	08	5
EXI2NCF	25	04	5
EXI2NJSS	25	10	5
EXI2PLTE	25	40	5
EXI3ALLOK	26	08	5
EXI3BADJ0	26	02	4
EXI3MISR#	26	10	5
EXI3NCKP	26	40	5
EXI3OOS	26	80	5
EXI3RDOP	26	01	4
EXI31ST	26	20	5
EXI4JOB#	27	20	5
EXI4NSOP	27	08	5
EXI4NSUB	27	40	5
EXI4RDJN	27	80	5
EXI4SUCC	27	01	5

EXM - Job submit activate/deactivate event

Name : DCLEXM

Function:
This segment declares a job submit activate/deactivate event. These events are built by eqqevpgm and eqqevcmd programs, are passed to the event writer via csa buffers, and are written to an event data set as part of an event record by the event writer.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EXM	JOB SUB ACT/DEACT EVENT
0	(0)	CHARACTER	1	EXMTYPE	'M' FOR JOBSUB EVENT
1	(1)	CHARACTER	1	EXMACT	'Y' FOR ACTIVATE 'N' FOR DEACTIVATE
2	(2)	CHARACTER	1	EXMJSTYP	'H' = HOST ENVIRONMENT 'F' = DISTRIBUTED ENVIRONMENT 'B' = BOTH ENVIRONMENTS
3	(3)	CHARACTER	1	*	RESERVED
4	(4)	CHARACTER	8	EXMUSER	TSO USER ID
12	(C)	CHARACTER	4	*	RESERVED
16	(10)	SIGNED	4	EXMCDAT	EVENT DATE (00YYDDDF)
20	(14)	SIGNED	4	EXMCTIM	EVENT TIME (SEC-100)
24	(18)	CHARACTER	52	*	RESERVED
76	(4C)	CHARACTER	4	EXMOPCID	EVENT ID EQQX

Cross reference

Name	Hex Offset	Hex Value	Level
EXM	0		1
EXMACT	1		2
EXMCDAT	10		2
EXMCTIM	14		2
EXMJSTYP	2		2
EXMOPCID	4C		2
EXMTYPE	0		2
EXMUSER	4		2

EXO - ON/OFF line event

Name : DCLEXO

Function:
This segment declares a workstation availability event generated by WSSTAT, EQQUSINW, EQQUSIN, or from TCP/IP connected trackers.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EXO	ON/OFF LINE EVENT
0	(0)	CHARACTER	1	EXOEVTYP	EVENT TYPE = O
1	(1)	CHARACTER	1	*	RESERVED
2	(2)	CHARACTER	1	EXOSTYPE	EXO SUBTYPE 'F'=OFF,'N'=ON
3	(3)	CHARACTER	8	EXODEST	DESTINATION
11	(B)	CHARACTER	1	EXOTYPE	DESTINATION TYPE: 'X'=XCF 'D'=DASD 'H'=HOST 'S'=SNA
12	(C)	BITSTRING	4	EXOFLGS	EXO FLAGS
		`1... ....`		EXOSYSG	SYSTEM GONE FLAG
		`.1.. ....`		EXOEDP	REQUEST FOR EDP
		`..1. ....`		EXOUX009	EVENT ISSUED DUE TO UX009 RC
12	(C)	BITSTRING	3	*	RESERVED
16	(10)	SIGNED	2	EXOGMTOF	GMT OFFSET IN MINUTES
18	(12)	SIGNED	2	*	RESERVED
20	(14)	CHARACTER	8	EXOCREAT	EVENT RECORD CREATION TIME WORD BNDRY
20	(14)	SIGNED	4	EXODATE	DATE FORMAT (00YYDDDF)
24	(18)	SIGNED	4	EXOTIME	TIME FORMAT (SECS*100)
28	(1C)	CHARACTER	8	EXOEVET	EVENT CREATION TIME WORD BNDRY
28	(1C)	SIGNED	4	EXOEDATE	DATE FORMAT (00YYDDDF)
32	(20)	SIGNED	4	EXOETIME	TIME FORMAT (SECS*100)
36	(24)	CHARACTER	4	EXOWSNAME	WORKSTATION NAME
40	(28)	CHARACTER	1	EXOREROUTE	REROUTE INDICATOR
41	(29)	CHARACTER	1	EXOSTARTOPR	STARTED OPERATIONS ACTION

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
42	(2A)	CHARACTER	4	EXOALTWS	ALTERNATE WS NAME
46	(2E)	CHARACTER	1	EXOAACT	ACTIONS ON ACTIVATE
47	(2F)	CHARACTER	29	*	RESERVED
76	(4C)	CHARACTER	4	EXOID	EVENT ID 'EQQX'

Cross reference

Name	Hex Offset	Hex Value	Level
EXO	0		1
EXOAACT	2E		2
EXOALTWS	2A		2
EXOCREAT	14		2
EXODATE	14		3
EXODEST	3		2
EXOEDATE	1C		3
EXOEDP	C	40	3
EXOETIME	20		3
EXOEVET	1C		2
EXOEVTYP	0		2
EXOFLGS	C		2
EXOGMTOF	10		2
EXOID	4C		2
EXOREROUTE	28		2
EXOSTARTOPR	29		2
EXOSTYPE	2		2
EXOSYSG	C	80	3
EXOTIME	18		3
EXOTYPE	B		2
EXOUX009	C	20	3
EXOWSNAME	24		2

EXP - Restart and clean up event

Name : DCLEXP

Function:
This segment declares the restart and clean up event.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EXP	JOBLOG RETRIEVAL EVENT
0	(0)	CHARACTER	1	EXPEVTYP	EVENT TYPE = P

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
1	(1)	CHARACTER	1	EXPTYPE	TYPE OF INITIATED PROCESS: I=OPINFO RETRIEVAL INITIATED O=OPINFO RETRIEVAL STARTED 2=OPINFO RETRIEVAL ENDED C=CLEAN UP EVENT D=DELETE RESTART AND CLEANUP DSTORE OLD ENTRIES X=ASK OPINFO RETRY
2	(2)	CHARACTER	1	EXPSUBT	SUBTYPE: I=CLEAN UP INIT 0=CLEAN UP START 2=CLEAN UP END
3	(3)	CHARACTER	1	EXPCALLER	G=GS TASK W=WSA TASK E=EM TASK A=AR TASK F=FL TASK
4	(4)	CHARACTER	16	EXPADID	ADID
20	(14)	CHARACTER	8	EXPOCCTOK	OCCURRENCE TOKEN
28	(1C)	CHARACTER	8	EXPDEST	DESTINATION
36	(24)	CHARACTER	30	EXPTYPES	DECLARATION OF TYPES
66	(42)	UNSIGNED	2	EXPGMTOF	GMT OFFSET IN MINUTES
68	(44)	CHARACTER	8	EXPTIMES	REQUEST TIME STAMP AS PASSED FROM THE CONTROLLER
68	(44)	SIGNED	4	EXPTDATE	DATE FORMAT (00YYDDDF)
72	(48)	SIGNED	4	EXPTTIME	TIME FORMAT (SECS*100)
76	(4C)	CHARACTER	4	EXPID	EVENT IDENTIFICATION (EQQn)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
36	(24)	STRUCTURE	30	EXPTYPED	PD=DELETE RESTART AND CLEAN UP DSTORE OLD ENTRIES
36	(24)	CHARACTER	8	EXPDCP16	CP16 ID
44	(2C)	CHARACTER	22	*	FREE

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
36	(24)	STRUCTURE	30	EXPTYPEX	PX=ASK OPINFO RETRY
36	(24)	CHARACTER	8	EXPXJNM	
44	(2C)	CHARACTER	8	EXPXJID	
52	(34)	SIGNED	4	EXPXRDRD	
56	(38)	SIGNED	4	EXPXRDRT	
60	(3C)	UNSIGNED	3	EXPXOCC	
63	(3F)	UNSIGNED	1	EXPROPR#	
64	(40)	CHARACTER	1	EXPFLAGS	
65	(41)	CHARACTER	1	EXPCP15	SET TO Y IF FROM DCPC2

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
36	(24)	STRUCTURE	30	EXPTYPEI	PI=OPINFO RETRIEVAL INITIATION
36	(24)	UNSIGNED	2	EXPIOPR#	OPERATION NUMBER
38	(26)	CHARACTER	8	EXPIUSER	TSO USERID
46	(2E)	CHARACTER	4	EXPIWS	WORKSTATION NAME
50	(32)	CHARACTER	8	EXPIJNM	JOBNAME
58	(3A)	CHARACTER	8	EXPIJID	JOB ID

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
36	(24)	STRUCTURE	30	EXPTYPECI	PC=OPINFO RETRIEVAL INIT CP15
36	(24)	UNSIGNED	2	EXPPCIOPR#	OPERATION NUMBER
38	(26)	CHARACTER	8	EXPCIUSER	TSO USERID
46	(2E)	CHARACTER	4	EXPCIWS	WORKSTATION NAME
50	(32)	CHARACTER	8	EXPCIJNM	JOBNAME
58	(3A)	CHARACTER	8	EXPCIJID	JOB ID

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
36	(24)	STRUCTURE	24	EXPTYPE0	P0=OPINFO RETRIEVAL STARTED
36	(24)	UNSIGNED	2	EXP0OPR#	OPERATION NUMBER
38	(26)	CHARACTER	8	EXP0JNM	JOBNAME
46	(2E)	CHARACTER	8	EXP0JID	JOB ID
54	(36)	CHARACTER	6	*	FREE

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
36	(24)	STRUCTURE	30	EXPTYPE2	P2=OPINFO RETRIEVAL ENDED
36	(24)	UNSIGNED	2	EXP2OPR#	OPERATION NUMBER
38	(26)	CHARACTER	1	EXP2STAT	FINAL RESULT OF JOBLOG RETRIEVAL: C=COMPLETED NORMALLY E=PROCESSING ENDED IN ERROR
39	(27)	CHARACTER	1	EXP2XST	BLANK=DEFAULT VALUE
40	(28)	CHARACTER	8	EXP2USER	USERID TO BE NOTIFIED
48	(30)	CHARACTER	8	EXP2JNM	JOBNAME
56	(38)		8	EXP2JID	JOB ID
64	(40)	CHARACTER	1	EXP2FLAGS	FLAGS FROM ARC DQE
				EXP2FROMAR	OPI FROM AR TASK
				*	
65	(41)	CHARACTER	1	*	RESERVED

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
36	(24)	STRUCTURE	30	EXPTYPREQ	PLR/PCR=INIT APPLY LOGIC/CLEAN UP
36	(24)	UNSIGNED	2	EXPROPR#	OPERATION NUMBER
38	(26)	CHARACTER	8	EXPRJNM	JOBNAME
46	(2E)	CHARACTER	8	EXPRJID	JOB ID (ONLY NUMERIC PART)
54	(36)	CHARACTER	1	*	FREE
55	(37)	CHARACTER	1	EXPRDIA	CLEAN UP FROM DIALOG: Y=CLEAN UP IS PRESENT N=CLEAN UP NOT PRESENT BLANK=NOT FROM DIALOG
56	(38)	BITSTRING	1	EXPRFLAG	EXPANDED JCL REALLY USED
				EXPREXP JCLUSED	EXPANDED JCL USED
				*	FREE
57	(39)	CHARACTER	1	*	RESERVED
58	(3A)	CHARACTER	8	EXPRUSER	TSO USER

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
36	(24)	STRUCTURE	30	EXPTYPCNF	PLR/PCR=INIT APPLY LOGIC/CLEAN UP ENDED
36	(24)	UNSIGNED	2	EXPCOPR#	OPERATION NUMBER
38	(26)	CHARACTER	4	EXPCGSTOK	GS REQUEST TOKEN
42	(2A)	CHARACTER	8	EXPCJNM	JOBNAME
50	(32)	CHARACTER	8	EXPCJID	JOB ID
58	(3A)	CHARACTER	1	EXPCSTAT	C=COMPLETED E=ERROR
59	(3B)	CHARACTER	1	EXPCXST	EXTENDED STATUS
60	(3C)	ADDRESS	4	EXPCJCL	POINTER TO JCL FOR DOA
64	(40)	BITSTRING	1	EXPCFLAG	EXPANDED JCL REALLY USED
				EXPCEXPJCLUSED	
				*	FREE
65	(41)	CHARACTER	1	*	RESERVED

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
36	(24)	STRUCTURE	30	EXPTYPC2	PC2=MERGE FOR CLEAN UP OPERATION
36	(24)	UNSIGNED	2	EXPC2OPR#	OPERATION NUMBER
38	(26)	CHARACTER	4	EXPC2GSTOK	GS REQUEST TOKEN
42	(2A)	CHARACTER	8	EXPC2JNM	JOBNAME
50	(32)	CHARACTER	8	EXPC2JID	JOB ID
58	(3A)	CHARACTER	1	EXPC2OSTA	JOBLOG RETRIEVAL STATUS: C=COMPLETED E=ERROR
59	(3B)	CHARACTER	1	EXPC2STAT	CLEAN UP STATUS: C=COMPLETED E=ERROR
60	(3C)	CHARACTER	1	EXPC2XST	EXTENDED STATUS
61	(3D)	CHARACTER	1	EXPC2FLAGS	FLAGS FROM ARC DQE
62	(3E)	CHARACTER	4	*	RESERVED

Cross reference

Name	Hex Offset	Hex Value	Level
EXP	0		1
EXPADID	4		2
EXPCALLER	3		2
EXPCEXPJCLUSED	40	80	3
EXPCGSTOK	26		2
EXPCIJID	3A		2
EXPCIJNM	32		2
EXPCIOPR#	24		2
EXPCIUSER	26		2
EXPCIWS	2E		2
EXPCJCL	3C		2
EXPCJID	32		2
EXPCJNM	2A		2
EXPCOPR#	24		2

Name	Hex Offset	Hex Value	Level
EXPCSTAT	3A		2
EXPCXST	3B		2
EXPC2FLAGS	3D		2
EXPC2GSTOK	26		2
EXPC2JID	32		2
EXPC2JNM	2A		2
EXPC2OPR#	24		2
EXPC2OSTA	3A		2
EXPC2STAT	3B		2
EXPC2XST	3C		2
EXPDCP16	24		2
EXPDEST	1C		2
EXPEVTYP	0		2
EXPGMTOF	42		2
EXPID	4C		2
EXPIJID	3A		2
EXPIJNM	32		2
EXPIOPR#	24		2
EXPIUSER	26		2
EXPIWS	2E		2
EXPOCCTOK	14		2
EXPRDIA	37		2
EXPREXPJCLUSED	38	80	3
EXPRFLAG	38		2
EXPRJID	2E		2
EXPRJNM	26		2
EXPROPR#	24		2
EXPRUSER	3A		2
EXPSUBT	2		2
EXPTDATE	44		3
EXPTIMES	44		2
EXPTTIME	48		3
EXPTYPCNF	24		1
EXPTYPC2	24		1
EXPTYPE	1		2
EXPTYPECI	24		1
EXPTYPED	24		1
EXPTYPEI	24		1
EXPTYPES	24		2
EXPTYPEX	24		1
EXPTYPE0	24		1

Name	Hex Offset	Hex Value	Level
EXPTYPE2	24		1
EXPTYPREQ	24		1
EXPXCP15	41		2
EXPXFLAGS	40		2
EXPXJID	2C		2
EXPXJNM	24		2
EXPXOCC	3C		2
EXPXOPR#	3F		2
EXPXRDRD	34		2
EXPXRDRT	38		2
EXP0JID	2E		2
EXP0JNM	26		2
EXP0OPR#	24		2
EXP2FLAGS	40		2
EXP2FROMAR	40	80	3
EXP2JID	38		2
EXP2JNM	30		2
EXP2OPR#	24		2
EXP2STAT	26		2
EXP2USER	28		2
EXP2XST	27		2

EXR - exit record

Name : DCLEXR

Function:
This segment declares an exit record. Exit records are built by SMF and JES exits, passed to
The event writer via CSA buffers, and are written to an event data set as part of an event record
by the event writer.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	exr	exit record
0	(0)	CHARACTER	2	exrtype	record type
0	(0)	CHARACTER	1	exrsyst	system type AB
1	(1)	CHARACTER	1	exrevtyp	event type 123456
2	(2)	CHARACTER	1	exrstype	event subtype (only type3)
3	(3)	BITSTRING	1	exrflags	exit flags
		1...		exrretry	retry release command
		.1..		exropcho	this job is in hold
		..1.		exroheld	job was held by opc
		...1		exrjkill	job killed by jes in rdr
		.... 1...		exrjccec	error code from jcc
		.... .1..		exrjccch	checked by jcc
		.... ..1.		exrjccer	error in jcc

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		1		exrrlast	retcode(last) active
4	(4)	BITSTRING	1	exrtermf	termination flags
		1...		exrjcset	exrjcout is valid
		.1..		exrjcout	job has jcc sysout
		..1.		exrdiscr	discrepancy msg DBMC
		...1		exrmchld	1= msgclass is held
		.... 1...		exrcmreq	1= requeue msg class
		.... .1..	*		free D90C
		.... ..1.		exrjbtab	0=normal, 1=abend (job)
		.... ..1.		exrabend	0=normal, 1=abend (step)
		1		exrflush	0=normal, 1=flushed step
5	(5)	BITSTRING	1	exrerror	job error switch lcterror
		1...		exrfail	job failed
		.1..		exrJQA	ON= is a JQA
		..1.		exrZ2level	
		...1		exrR4level	
		.... 1...		*	alloc but not unalloc done
		.... .1..		exrcfal	job failed on cond codes
		.... ..1.		exrjcjob	JCJOB processed ok D54A
		1		exrlastfl	last step flushed BGIA
6	(6)	SIGNED	2	exrgmtof	gmt offset in minutes
8	(8)	CHARACTER	8	exrjobn	job name
16	(10)	CHARACTER	8	exrjobid	job number
24	(18)	CHARACTER	8	exrcreat	event creation time
24	(18)	SIGNED	4	exrdate	date format (00yydddf)
28	(1C)	SIGNED	4	exrtime	time format (secs*100)
32	(20)	CHARACTER	8	exrjsrdr	jes reader date & time
32	(20)	SIGNED	4	exrrdate	date format (00yydddf)
36	(24)	SIGNED	4	exrrtime	time format (secs*100)
40	(28)	SIGNED	4	exrsdate	operation start date
44	(2C)	SIGNED	4	exrstime	operation start time
48	(30)	SIGNED	4	exredate	operation end date
52	(34)	SIGNED	4	exretime	operation end time
52	(34)	SIGNED	4	exrorgid	nje origin job number
56	(38)	CHARACTER	8	exrstepn	job step name
56	(38)	CHARACTER	8	exronje	name of orig nje nod
56	(38)	CHARACTER	1	exrclass	printout class
57	(39)	CHARACTER	1	*	reserved
58	(3A)	SIGNED	2	exrasid	job asid
60	(3C)	SIGNED	4	exrexeid	NJE execution jobn
64	(40)	CHARACTER	8	exrpstep	procedure step name
64	(40)	CHARACTER	8	exrnnje	thisnext nje node
64	(40)	CHARACTER	8	exrform	form number
72	(48)	SIGNED	2	exrcode	completioncondition code
74	(4A)	CHARACTER	1	exrindic	status indicators
		1...		exrjesv4	jes sp4 or above13P
		.1..		exrspun	spun off ds rcd
		..1.		exrterm	oper terminated datagroup
		...1		exrinter	oper interrupted --
		.... 1...		exrrstrt	oper restarted --
		.... .1..		exrndest	not final fsysmsgs 3P
		.... ..1.		exrnods4	no sysmsgs found 3P
		1		exrsuspd	suspended

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
75	(4B)	UNSIGNED	1	exrstpnr	step number
75	(4B)	BITSTRING	1	exrpurge	job purge bits
		111.		*	not used
		...1		exrSDEP	SDEP filter used
		.... 1...		exrlastab	last step abended CAVA
		.... .1..		exrstall	stepevents(all)
		.... ..1.		exrstnz	stepevents(no) flag
		1		exropcan	cancelled by oper
76	(4C)	CHARACTER	4	exropcid	opc identifier

Cross reference

Name	Hex Offset	Hex Value	Level
exr	0		1
exrabend	4	02	4
exrasid	3A		4
exrcfal	5	04	3
exrclass	38		4
exrcmreq	4	08	3
exrcode	48		2
exrcreat	18		2
exrdate	18		3
exrdiscr	4	20	3
exredate	30		2
exrerror	5		2
exretime	34		2
exrevtyp	1		3
exrexeid	3C		4
exrfail	5	80	3
exrflags	3		2
exrflush	4	01	3
exrform	40		4
exrgmtof	6		2
exrindic	4A		2
exrinter	4A	10	3
exrjbtab	4	02	3
exrjccch	3	04	3
exrjccec	3	08	3
exrjccer	3	02	3
exrjcjob	5	02	3
exrjcout	4	40	3
exrjcset	4	80	3

Name	Hex Offset	Hex Value	Level
exrjesv4	4A	80	3
exrjkill	3	10	3
exrjobid	10		2
exrjobn	8		2
exrJQA	5	40	3
exrjsrdr	20		2
exrlastab	4B	08	4
exrlastfl	5	01	3
exrmchld	4	10	3
exrndest	4A	04	3
exrnnje	40		3
exrnods4	4A	02	3
exroheld	3	20	3
exronje	38		3
exropcan	4B	01	4
exropcho	3	40	3
exropcid	4C		2
exrorgid	34		3
exrpstep	40		2
exrpurge	4B		3
exrrdate	20		3
exrretry	3	80	3
exrrlast	3	01	3
exrrstrt	4A	08	3
exrrtime	24		3
exrR4level	5	10	3
exrsdate	28		2
exrSDEP	4B	10	4
exrspun	4A	40	3
exrstall	4B	04	4
exrstepn	38		2
exrstime	2C		2
exrstnz	4B	02	4
exrstpnr	4B		2
exrstype	2		2
exrsuspd	4A	01	3
exrsyst	0		3
exrterm	4A	20	3
exrtermf	4		2
exrtime	1C		3
exrtype	0		2

Name	Hex Offset	Hex Value	Level
exrZ2level	5	20	3

EXS - Resource event

Name : DCLEXS

Function:
This segment declares a special resource change event record. These event records are built by the EQQEVPGM and EQQEVCMD programs, are passed to the event writer via csa buffers, and are written to an event data set as part of an event record by the event writer. These events can also be generated from a call to EQQUSIN, ESSUSINS, or from the API.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EXS	SPECIAL RESOURCE EVENT
0	(0)	CHARACTER	1	EXSTYPE	'S' FOR SPECIAL RESOURCE EVENT
1	(1)	CHARACTER	1	EXSAVAIL	GLOBAL AVAILABILITY SETTING Y=AVAILABLE, N=NOT AVAILABLE R=RESET <BLANK>'=KEEP
2	(2)	CHARACTER	1	EXSCREATE	CREATE IF NEW, Y|N
3	(3)	CHARACTER	1	EXSSUBTYPE	NOT USED
4	(4)	UNSIGNED	1	EXSGDGL	LENGTH OF GDG DS ROOT
5	(5)	CHARACTER	1	EXSR3DEF	DEFAULT SET IN R3
6	(6)	SIGNED	2	EXSGMTOF	GMT OFFSET (MINUTES)
8	(8)	SIGNED	4	EXSCDAT	CREATION DATE (00YYDDDF)
12	(C)	SIGNED	4	EXSCTIM	CREATION TIME (SEC*100)
16	(10)	SIGNED	4	EXSEDAT	EVENT DATE (00YYDDDF)
20	(14)	SIGNED	4	EXSETIM	EVENT TIME (SEC*100)
24	(18)	CHARACTER	8	EXSUSER	NAME OF UPDATING USER
32	(20)	CHARACTER	44	EXSNAME	RESOURCE NAME
76	(4C)	CHARACTER	4	EXSOPCID	EVENT ID 'EQQX'

Cross reference

Name	Hex Offset	Hex Value	Level
EXS	0		1
EXSAVAIL	1		2
EXSCDAT	8		2
EXSCREATE	2		2
EXSCTIM	C		2
EXSEDAT	10		2
EXSETIM	14		2
EXSGDGL	4		2
EXSGMTOF	6		2
EXSNAME	20		2

Name	Hex Offset	Hex Value	Level
EXSOPCID	4C		2
EXSR3DEF	5		2
EXSSUBTYPE	3		2
EXSTYPE	0		2
EXSUSER	18		2

EXSA - Resource event extension

Name : DCLEXSA

Function:
Defines additional data for a resource event (s).

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	88	EXSA	RES EVENT ADDENDUM
0	(0)	CHARACTER	4	EXSA_ID	DESCRIPTOR - EXSA
4	(4)	CHARACTER	2	EXSA_VER	BLOCK MAPPING VERSION
6	(6)	CHARACTER	44	EXSANAME	RESOURCE NAME
50	(32)	CHARACTER	4	EXSAOPCID	OPCID
54	(36)	BITSTRING	1	EXSAFLG1	FLAG BYTE 1
		1...		EXSARODM	RODM EVENT
		.1..		EXSADEVS	DEVIATION SET
		..1.		EXSAQUAS	QUANTITY SET
		...1		EXSADEVR	DEVIATION RESET
		.... 1...		EXSAQUAR	QUANTITY RESET
		.... .1..		EXSARODMLOST	RODM LOST
		.... ..1.		EXSALIFESPANB	LIFESPAN SPECIFIED
		1		*	RESERVED
55	(37)	CHARACTER	1	EXSATYPE	UPDATE TYPE. A, Q OR D UPD TYPE REQD FOR RODM EV
56	(38)	SIGNED	4	EXSAQUANT	QUANTITY
60	(3C)	SIGNED	4	EXSADEV	DEVIATION
64	(40)	SIGNED	4	* (2)	RESERVED
72	(48)	CHARACTER	1	EXSASUBE	
73	(49)	CHARACTER	1	EXSALIFESPAN	Y=YES N=NO R=RESET
76	(4C)	SIGNED	4	EXSALIFETIME	MINUTES
80	(50)	CHARACTER	8	EXSAJOB	JOB NAME FROM DS TRIGGER
88	(58)	CHARACTER	2	*	RESERVED

Cross reference

Name	Hex Offset	Hex Value	Level
EXSA	0		1
EXSA_ID	0		2
EXSA_VER	4		2
EXSADEV	3C		2

Name	Hex Offset	Hex Value	Level
EXSADEVR	36	10	3
EXSADEVS	36	40	3
EXSAFLG1	36		2
EXSAJOB	50		2
EXSALIFESPAN	49		2
EXSALIFESPANB	36	02	3
EXSALIFETIME	4C		2
EXSANAME	6		2
EXSAOPCID	32		2
EXSAQUANT	38		2
EXSAQUAR	36	08	3
EXSAQUAS	36	20	3
EXSARODM	36	80	3
EXSARODMLOST	36	04	3
EXSASUBE	48		2
EXSATYPE	37		2

EXT - Ready to receive event

Name : DCLEXT

Function:
This segment declares a SET TIME (GMTOFFSET) event record. These event records
are built by the EQQEVPGM and EQQEVCMD programs, passed to the Event Writer through CSA buffers, and written to an event data set as part of an event record by the Event Writer.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	EXT	READY TO RECEIVE EVENT
0	(0)	CHARACTER	1	EXTETYPE	EVENT TYPE = T
1	(1)	CHARACTER	3	*	RESERVED
4	(4)	CHARACTER	4	EXTSID	SYSTEM IDENTIFICATION
8	(8)	SIGNED	4	EXTGMTOF	GMT OFFSET IN MINUTES
12	(C)	SIGNED	4	EXTCDAT	CREATION DATE FORMAT (00YYDDDF)
16	(10)	SIGNED	4	EXTCTIM	CREATION TIME FORMAT (SEC*100)
20	(14)	SIGNED	4	EXTEDAT	EVENT DATE FORMAT (00YYDDDF)
24	(18)	SIGNED	4	EXTETIM	EVENT TIME FORMAT (SEC*100)
28	(1C)	SIGNED	4	EXTGMTSEC	GMT OFFSET
32	(20)	CHARACTER	4	EXTENF	T TYPE ENF
36	(24)	UNSIGNED	1	EXTENFB1	ENF 53 BYTE 1
37	(25)	UNSIGNED	1	EXTENFB2	ENF 53 BYTE 2
38	(26)	UNSIGNED	1	EXTENFB3	ENF 53 BYTE 3
39	(27)	CHARACTER	1	EXTRCHA	REAL CHANGE Y/M
40	(28)	CHARACTER	48	*	RESERVED
76	(4C)	CHARACTER	4	EXTOPCID	EVENT ID 'EQQX'

Cross reference

Name	Hex Offset	Hex Value	Level
EXT	0		1
EXTCDAT	C		2
EXTCTIM	10		2
EXTEDAT	14		2
EXTETIM	18		2
EXTGMTOF	8		2
EXTGMTSEC	1C		2
EXTOPCID	4C		2
EXTSID	4		2
EXTTYPE	0		2

EXU - User event record

Name : DCLEXU

Function:
This segment declares a user exit event record. User event records are built by OPSTAT, EQQUSIN, EQQUSINT, or the API and passed to the event writer via CSA buffers, and are written to an event data set as part of an event record by the event writer.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	EXU	USER EVENT RECORD
0	(0)	CHARACTER	1	EXUSYST	'G' FOR GENERAL WS EVENT
1	(1)	CHARACTER	1	EXUTYPE	'S'=START,'C'=COMPLETE OR 'E'=ENDED IN ERROR
2	(2)	CHARACTER	1	EXUCLASS	SYSOUT CLASS
3	(3)	BITSTRING	1	EXUFLAGS	DHKC
		1...		EXUSORTIA	ON = SORT FOR IA DHKA
		.111 1111		*	DHKA
4	(4)	CHARACTER	8	EXUJOBN	JOB NAME
12	(C)	CHARACTER	16	EXUAPPL	APPLICATION NAME
28	(1C)	CHARACTER	4	EXUWSN	WORK STATION NAME
32	(20)	SIGNED	4	EXUCDAT	CREATION DATE (00YYDDDF)
36	(24)	SIGNED	4	EXUCTIM	CREATION TIME (SEC*100)
40	(28)	SIGNED	4	EXUEDAT	EVENT DATE (00YYDDDF)
44	(2C)	SIGNED	4	EXUETIM	EVENT TIME (SEC*100)
44	(2C)	SIGNED	2	EXUGMTOA	GMT OFFSET FOR APP
46	(2E)	SIGNED	2	*	RESERVED
48	(30)	SIGNED	2	EXUOPNO	OPERATION NUMBER
50	(32)	CHARACTER	10	EXUINP	APPL INP ARR (YYMMDDHHMM)
60	(3C)	SIGNED	2	EXUDUR	DURATION (MINUTES)
62	(3E)	SIGNED	2	EXUGMTOF	GMT OFFSET (MINUTES) ONLY USED IF EXUTYPE = 'C'
64	(40)	CHARACTER	4	EXUERR	ERROR CODE. ONLY USED IF EXUTYPE = 'E'
68	(44)	CHARACTER	8	EXUFORM	FORM NUMBER
76	(4C)	CHARACTER	4	EXUOPCID	event id 'EQQx'

Cross reference

Name	Hex Offset	Hex Value	Level
EXU	0		1
EXUAPPL	C		2
EXUCDAT	20		2
EXUCLASS	2		2
EXUCTIM	24		2
EXUDUR	3C		2
EXUEDAT	28		2
EXUERR	40		2
EXUETIM	2C		2
EXUFLAGS	3		2
EXUFORM	44		2
EXUGMTOA	2C		3
EXUGMTOF	3E		2
EXUINP	32		2
EXUJOBN	4		2
EXUOPCID	4C		2
EXUOPNO	30		2
EXUSORTIA	3	80	3
EXUSYST	0		2
EXUTYPE	1		2
EXUWSN	1C		2

EXUA - User event addendum

Name : DCLEXUA

Function:
This segment declares a user event addendum record.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	72	EXUA	USER EVENT ADDENDUM
0	(0)	CHARACTER	4	EXUA_ID	DESCRIPTOR - EXUA
4	(4)	CHARACTER	2	EXUA_VER	BLOCK MAPPING VERSION
6	(6)	CHARACTER	10	EXUAINP	IA (YYMMDDHHMM)
16	(10)	SIGNED	2	EXUADUR	DURATION (MINUTES)
18	(12)	SIGNED	2	EXUAGMTOF	GMT OFFSET (MINUTES)
20	(14)	CHARACTER	4	EXUAERR	ERROR CODE
24	(18)	CHARACTER	8	EXUAFORM	FORM NUMBER
32	(20)	CHARACTER	4	EXUAOPCID	SCHEDULER ID
36	(24)	CHARACTER	16	EXUUDATA	USERDATA
52	(34)	CHARACTER	1	EXUAOPIN	OPINFO COMMAND Y/N

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
53	(35)	CHARACTER	3	*	RESERVED
56	(38)	CHARACTER	8	EXUATOKEN_FULL	OPERATION FULL TOKEN
56	(38)	CHARACTER	4	EXUATOKEN_PRE	PREFIX
60	(3C)	UNSIGNED	4	EXUATOKEN	SCHEDULER TOKEN
64	(40)	CHARACTER	6	EXUAJOBNBR	JOB NUMBER
70	(46)	CHARACTER	2	*	RESERVED

Cross reference

Name	Hex Offset	Hex Value	Level
EXUA	0		1
EXUA_ID	0		2
EXUA_VER	4		2
EXUADUR	10		2
EXUAERR	14		2
EXUAFORM	18		2
EXUAGMTOF	12		2
EXUAINP	6		2
EXUAJOBNBR	40		2
EXUAOPCID	20		2
EXUAOPIN	34		2
EXUATOKEN	3C		3
EXUATOKEN_FULL	38		2
EXUATOKEN_PRE	38		3
EXUUDATA	24		2

EXV - Scheduling environment event

Name : DCLEXV

Function:
This segment declares a scheduling environment availability event.
It can be:
VS (single SE name available)
VM (multiple SE name available)
A buffer is associated with this event containing a list of SE names
VL (log event generated by the controller to track CP changes to VS/VM events)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	EXV	SCHEDULING ENVIRONMENT EVENT
0	(0)	CHARACTER	1	EXVTYP	EVENT TYPE=V

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
1	(1)	CHARACTER	1	EXVTTYPE	SUBTYPE: S=SINGLE SCHEDULING EVENT M=MULTIPLE SCHEDULING EVENT
2	(2)	CHARACTER	2	*	
4	(4)	BITSTRING	4	EXVQUAL	ENF EVENT QUALIFIER
8	(8)	CHARACTER	4	EXVTRA	ENFREQ MACRO TRACKER NAME PARAMETER
12	(C)	UNSIGNED	4	EXVEVCODE	EVENT CODE (57 or 41)
16	(10)	CHARACTER	14	*	
30	(1E)	UNSIGNED	2	EXVFLAGS	FLAGS
30	(1E)	BITSTRING	1	EXVFLAGS	
				EXVSUTOP	ON=EVENT GENERATED BY SUBMIT TASK
				*	
32	(20)	CHARACTER	30	*	
62	(3E)	SIGNED	2	EXVGMTOFF	GMT OFFSET IN MINUTES
64	(40)	CHARACTER	8	EXVTIMES	REQUEST TIME STAMP
64	(40)	SIGNED	4	EXVTDATE	DATE FORMAT (00YYDDDF)
68	(44)	SIGNED	4	EXVTTIME	TIME FORMAT (SECS*100)
72	(48)	CHARACTER	4	EXVSSNM	ECHO SUBSYSTEM NAME
76	(4C)	CHARACTER	4	EXVID	EVENT IDENTIFICATION 'EQQF'

Cross reference

Name	Hex Offset	Hex Value	Level
EXV	0		1
EXVDATA	20		2
EXVEVCODE	C		1
EXVEVTYP	0		2
EXVFLAGS	1E		2
EXVGMTOFF	3E		2
EXVID	4C		2
EXVLEVM	2C	20	3
EXVLEVS	2C	40	3
EXVLFLAG	2C		2
EXVLOCCTOK	24		2
EXVLOPNUM	22		2
EXVLPLXID	20		2
EXVLSCHE	2E		2
EXVLXSTAT	2C	80	3
EXVMNUM	24		2
EXVMPLXID	20		2
EXVQUAL	4		2
EXVSPLXID	20		3

Name	Hex Offset	Hex Value	Level
EXVSSCHE	24		3
EXVSSNM	48		2
EXVSSYSNM	34		2
EXVSUTOP	1E	80	3
EXVTDATE	40		2
EXVTIMES	40		1
EXVTRA	8		1
EXVTTIME	44		1
EXVTYPE	1		2
EXVTYPEL	20		2
EXVTYPEM	20		3
EXVTYPES	20		3

EXW - Workstation status event

Name : DCLEXW

Function:
This segment declares a workstation status event record. These event records are built via the WSSTAT, EQQUSIN, EQQUSINW, or the API. They are passed to the event writer via CSA buffers, and written to an event data set as part of an event record by the event writer.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EXW	WORKSTATION STATUS EVENT
0	(0)	CHARACTER	1	EXWTYPE	'W' FOR WSSTAT EVENT
1	(1)	CHARACTER	8	EXWUSER	NAME OF UPDATING USER
9	(9)	CHARACTER	8	EXWDEST	DESTINATION
17	(11)	CHARACTER	4	EXWWSN	WORKSTATION NAME
21	(15)	BITSTRING	1	EXWFLAGS	FLAGS
		1...		EXWTRK	TRK GENERATED EVENT
		.111 1111		*	RESERVED
22	(16)	CHARACTER	1	EXWSTAT	NEW STATUS
23	(17)	CHARACTER	1	EXWREROUT	REROUTE INDICATOR
24	(18)	CHARACTER	1	EXWSTOP	STARTED OPERATION INDICATOR
25	(19)	CHARACTER	4	EXWALTWS	ALTERNATE WORKSTATION NAME
29	(1D)	CHARACTER	3	*	RESERVED
32	(20)	SIGNED	4	EXWCDAT	CREATION DATE (00YYDDDF)
36	(24)	SIGNED	4	EXWCTIM	CREATION TIME (SEC-100)
40	(28)		1	EXWCMD	CMD L/U/S/P
41	(29)	CHARACTER	16	EXWMGR	DOMAIN MANAGER
57	(39)	CHARACTER	19	*	RESERVED
76	(4C)	CHARACTER	4	EXWOPCID	EVENT ID 'EQQX'

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	EXX	TRACKER EVENT RECORD

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	1	EXXSYST	'X' TYPE EVENT
1	(1)	CHARACTER	1	EXXTYPE	INDICATOR
2	(2)	CHARACTER	1	EXXSYSID	INDICATOR
3	(3)	BITSTRING	1	EXXFLAG	FLAGS
		1...		EXXERRCC	ERRC IN CHAR
		.111 1111		*	RESERVED
4	(4)	CHARACTER	4	*	RESERVED
8	(8)	CHARACTER	8	EXXJOBN	JOB NAME
16	(10)	CHARACTER	5	EXXJNUM	JOB NUMBER
21	(15)	CHARACTER	3	*	RESERVED
24	(18)	SIGNED	4	EXXCDAT	CREATION DATE (00YYDDDF)
28	(1C)	SIGNED	4	EXXCTIM	CREATION TIME (SEC*100)
32	(20)	SIGNED	2	EXXGMTOF	GMT OFFSET (MINUTES)
34	(22)	CHARACTER	2	*	RESERVED
36	(24)	SIGNED	4	EXXERR	ERROR CODE BIN FORM
40	(28)	CHARACTER	4	EXXERRC	ERROR CODE CHAR FORM
44	(2C)	CHARACTER	8	EXXTOKEN_FULL	OPERATION FULL TOKEN
44	(2C)	CHARACTER	4	EXXTOKEN_PRE	PREFIX
48	(30)	UNSIGNED	4	EXXTOKEN	OPCTOKEN
52	(34)	CHARACTER	7	EXXAS4ERRC	FULL AS400 ERROR CODE
59	(3B)	CHARACTER	17	*	RESERVED
76	(4C)	CHARACTER	4	EXXOPCID	EVENT ID 'EQQX'

Cross reference

Name	Hex Offset	Hex Value	Level
EXW	0		1
EXWALTWS	19		2
EXWCDAT	20		2
EXWCMD	20		2
EXWCTIM	24		2
EXWDEST	9		2
EXWFLAGS	15		2
EXWMGR	29		2
EXWOPCID	4C		2
EXWREROUT	17		2
EXWSTAT	16		2
EXWSTOP	18		2
EXWTRK	15	80	3
EXWTYPE	0		2
EXWUSER	1		2
EXWWSN	11		2

Name	Hex Offset	Hex Value	Level
EXX	0		1

Name	Hex Offset	Hex Value	Level
EXXAS4ERRC	34		2
EXXCDAT	18		2
EXXCTIM	1C		2
EXXERR	24		2
EXXERRC	28		2
EXXERRCC	3	80	3
EXXFLAG	3		2
EXXGMTOF	20		2
EXXJNUM	10		2
EXXJOBN	8		2
EXXOPCID	4C		2
EXXSYSID	2		2
EXXSYST	0		2
EXXTOKEN	30		3
EXXTOKEN_FULL	2C		2
EXXTOKEN_PRE	2C		3
EXXTYPE	1		2

EXX - Tracker agent event record

Name : DCLEXX

Function:
This segment declares an event record from a tracker agent.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	EXX	TRACKER EVENT RECORD
0	(0)	CHARACTER	1	EXXSYST	'X' TYPE EVENT
1	(1)	CHARACTER	1	EXXTYPE	INDICATOR
2	(2)	CHARACTER	1	EXXSYSID	INDICATOR
3	(3)	BITSTRING	1	EXXFLAG	FLAGS
		1...		EXXERRCC	ERRC IN CHAR
		.111 1111		*	RESERVED
4	(4)	CHARACTER	4	*	RESERVED
8	(8)	CHARACTER	8	EXXJOBN	JOB NAME
16	(10)	CHARACTER	5	EXXJNUM	JOB NUMBER
21	(15)	CHARACTER	3	*	RESERVED
24	(18)	SIGNED	4	EXXCDAT	CREATION DATE (00YYDDDF)
28	(1C)	SIGNED	4	EXXCTIM	CREATION TIME (SEC*100)
32	(20)	SIGNED	2	EXXGMTOF	GMT OFFSET (MINUTES)
34	(22)	CHARACTER	2	*	RESERVED
36	(24)	SIGNED	4	EXXERR	ERROR CODE BIN FORM
40	(28)	CHARACTER	4	EXXERRC	ERROR CODE CHAR FORM
44	(2C)	CHARACTER	8	EXXTOKEN_FULL	OPERATION FULL TOKEN
44	(2C)	CHARACTER	4	EXXTOKEN_PRE	PREFIX
48	(30)	UNSIGNED	4	EXXTOKEN	OPCTOKEN

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	CHARACTER	7	EXXAS4ERRC	FULL AS400 ERROR CODE
59	(3B)	CHARACTER	17	*	RESERVED
76	(4C)	CHARACTER	4	EXXOPCID	EVENT ID 'EQQX'

Cross reference

Name	Hex Offset	Hex Value	Level
EXX	0		1
EXXAS4ERRC	34		2
EXXCDAT	18		2
EXXCTIM	1C		2
EXXERR	24		2
EXXERRC	28		2
EXXERRCC	3	80	3
EXXFLAG	3		2
EXXGMTOF	20		2
EXXJNUM	10		2
EXXJOBN	8		2
EXXOPCID	4C		2
EXXSYSID	2		2
EXXSYST	0		2
EXXTOKEN	30		3
EXXTOKEN_FULL	2C		2
EXXTOKEN_PRE	2C		3
EXXTYPE	1		2

FOPC - OPERINFO structure

Name : DCLFOPC

Function:
This segment declares the operinfo structure, stored in the controller local data store.
 The OPERINFO record is structured as follows:

```
     1 OPERINFO
       2 common data
       2 run info ( )
         3 step info ( )
           4 DDINFO ( )
             5 DSNINFO ( )
             5 CATINFO ( )
             5 VOLINFO ( )
```

Common data segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	OPER_INFO	
0	(0)	CHARACTER	12	OPI_KEY	
0	(0)	CHARACTER	10	OPI_OPRTOK	OPERATION TOKEN
0	(0)	CHARACTER	8	OPI_OCCTOK	OCCURRENCY TOKEN
8	(8)	SIGNED	2	OPI_OPRNUM	OPERATION NUMBER
10	(A)	CHARACTER	2	*	RESERVED
12	(C)	CHARACTER	56	OPI_COMMINFO	
12	(C)	CHARACTER	16	OPI_APPLID	APPLICATION ID
28	(1C)	CHARACTER	10	OPI_IATIME	IA TIME
38	(26)	CHARACTER	8	OPI_JOBNAME	JOB NAME
46	(2E)	CHARACTER	8	OPI_JOBID	JOB IDENTIFIER
54	(36)	CHARACTER	2	*	RESERVED
56	(38)	SIGNED	4	OPI_JCLRBA	JCL RBA
60	(3C)	SIGNED	4	OPI_TOTOPSIZE	TOTAL LENGTH OF OPERINFO
64	(40)	SIGNED	2	OPI_OPRUNCTR	NUMBER OF RUNS OF THE OPERATION
66	(42)	SIGNED	2	OPI_TOTSTPCTR	STEP COUNTER NUMBER OF STEPS
68	(44)	CHARACTER	*	OPI_VARDATA	
68	(44)	CHARACTER	1	OPI_VARBYTE(*)	

Run info segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	32	OPI_RUNINFO	
0	(0)	SIGNED	4	OPI_NXTRUNRBA	NEXT RUN
4	(4)	SIGNED	2	OPI_RUNID	NUMBER OF RUN
6	(6)	SIGNED	2	OPI_STEPCTR	STEP COUNTER NUMBER OF STEPS
8	(8)	CHARACTER	8	OPI_JOBDT	
8	(8)	CHARACTER	4	OPI_JYEAR	YEAR OF JOBDATE
12	(C)	CHARACTER	2	OPI_JMONTH	MONTH OF JOBDATE
14	(E)	CHARACTER	2	OPI_JDAY	DAY OF JOBDATE
16	(10)	CHARACTER	8	OPI_JOBTM	JOBTIME IN X'HHMMSSHH' FORM
24	(18)	CHARACTER	8	OPI_RUNJOBID	JES JOB IDENTIFIER

Step info sub-segment

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	44	OPI_STPINFO	
0	(0)	SIGNED	4	OPI_NXTSTPRBA	NEXT STEP POSITION IN OPERINFO AREA
4	(4)	CHARACTER	8	OPI_STEPNAME	STEP NAME
12	(C)	CHARACTER	8	OPI_PRSTNAME	PROCSTEP NAME
20	(14)	SIGNED	2	OPI_STEPNUMB	STEP NUMBER
22	(16)	CHARACTER	5	OPI_STEPCODE	COMPLETION CODE (RC, USER, SYSTEM, FLUSH)
27	(1B)	CHARACTER	8	OPI_PGMNAME	PROGRAM NAME

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
35	(23)	CHARACTER	1	OPI_STEPSTAT	STEP STATUS
36	(24)	CHARACTER	2	OPI_RESCHAR	RESERVED
38	(26)	SIGNED	2	OPI_STPRUNID	NUMBER OF RUN
40	(28)	SIGNED	4	OPI_DDCTR	NUMBER OF DDS IN STEP

DDINFO Sub-segment:

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	24	OPI_DDINFO	
0	(0)	SIGNED	4	OPI_NXTDDRBA	NEXT DD POSITION WITHIN OPERINFO
4	(4)	CHARACTER	8	OPI_DDNAME	DDNAME - FILENAME OR STEPCAT OR JOBCA
12	(C)	SIGNED	4	OPI_DDNUMB	DD PROGRESSIVE NUMBER IN THE STEP
16	(10)	SIGNED	2	OPI_DDRUNID	NUMBER OF RUN WHERE THE DD WAS ALLOCATED
18	(12)	CHARACTER	1	OPI_DDTYPE	DD TYPE: U=USER S=STEPCAT
19	(13)	CHARACTER	1	OPI_DDINFOX	EXTRA INFO ABOUT DD (BIT BATTERY)
				OPI_DDPROT	DD PROTECTED
				OPI_DDNEVER	DD: STEP NEVER RE-EXECUTABLE
				OPI_DDNOREST	DD: STEP NEVER RESTARTABLE
				OPI_DDALWAYS *	DD: STEP RESTARTABLE
20	(14)	SIGNED	4	OPI_DSNCTR	NUMBER OF DSN WITHIN THE DD

DSNINFO Sub-segment:

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	82	OPI_DSNINFO	
0	(0)	SIGNED	4	OPI_NXTDSNRBA	NEXT DSN POSITION IN OPERINFO
4	(4)	CHARACTER	44	OPI_DSNAME	DATASET NAME
48	(30)	CHARACTER	1	OPI_DSNINST	DATASET STATUS: N=NEW O=OLD
49	(31)	CHARACTER	4	OPI_DSNOKST	DISP IF STEP IS NORMALLY EXECUTED
53	(35)	CHARACTER	4	OPI_DSNKOST	DISP IF STEP IS ABNORMALLY ENDED
57	(39)	CHARACTER	4	OPI_DSNDNORM	DISP NORMAL FROM JCL
61	(3D)	CHARACTER	4	OPI_DSNDABNM	DISP ABNORMAL FROM JCL
65	(41)	CHARACTER	1	OPI_DSNSTAT	STATUS OF DATASET AT END OF LAST JOBEXE
66	(42)	CHARACTER	1	OPI_DSNTYPE	ID DATASET AS GDG, SMS MANAGED
				OPI_GDG	.. DATASET IS A GDG
				OPI_SMS	.. DATASET IS SMS MANAGED
				OPI_VSAM	.. DATASET IS A VSAM

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
				OPI_MIGR	.. DATASET IS MIGRAT
				OPI_REFBK	.. DATASET IS MIGRAT
				OPI_JBCAT	.. IS A JOBCAT
				OPI_STCAT	.. IS A STEPCAT
				*	*
67	(43)	CHARACTER	1	OPI_DEVTYPE	DEVICE TYPE: D=DASD T=TAPE
68	(44)	CHARACTER	1	OPI_DSNCLEAN	SUBJECT TO CLEAN UP ACTIONS? (Y/N)
69	(45)	CHARACTER	1	OPI_DSNPROT	FLAG INDICATOR: DATASET IS PROTECTED
70	(46)	SIGNED	2	OPI_GDGRELNR	RELATIVE NUMBER FOR GDG
72	(48)	SIGNED	2	OPI_DSNNUMB	PROGRESSIVE DSN NUMBER WITHIN THE DD
74	(4A)	SIGNED	2	OPI_DSNCLNRC	CLEAN UP INTERNAòL REASON CODE
76	(4C)	SIGNED	2	OPI_DSNRUNID	NUMBER OF RUN WHERE THE DSN WAS ALLOCATED
78	(4E)	SIGNED	2	OPI_VOLCTR	NUMBER OF VOL FOR THE DATASET
80	(50)	CHARACTER	1	OPI_CATTYPE	CATALOG TYPE: BLANK=MASTERCAT
81	(51)	CHARACTER	1	OPI_DSNFLAGS	OTHER FLAGS
				OPI_VALIDREL	VALID GDG RELEASE NUMBER INDICATOR
				OPI_JCLFLVOL	VOLUME FROM JCL
				*	EVENT ID 'EQQX'

CATINFO Sub-segment:

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	44	OPI_CATLINFO	EXISTS ONLY IF OPI_CATTYPE IS NOT BLANK
0	(0)	CHARACTER	44	OPI_CATNAME	CATALOG NAME

VOLINFO Sub-segment:

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	12	OPI_VOLINFO	
0	(0)	CHARACTER	6	OPI_VOLSER	VOLUME SERIAL NUMBER
6	(6)	SIGNED	2	OPI_VOLNUMB	PROGRESSIVE VOL NUMBER
8	(8)	SIGNED	2	OPI_DSNSQNUM	SEQUENCE NUMBER OF THE DATASET
10	(A)	CHARACTER	2	OPI_RESCHAR	RESERVED

Cross reference

Name	Hex Offset	Hex Value	Level
OPER_INFO	0		1
OPI_APPLID	C		3

Name	Hex Offset	Hex Value	Level
OPI_CATLINFO	0		1
OPI_CATNAME	0		2
OPI_CATTYPE	50		2
OPI_COMMINFO	C		2
OPI_DDALWAYS	13	10	3
OPI_DDCTR	28		2
OPI_DDINFO	0		1
OPI_DDINFOX	13		2
OPI_DDNAME	4		2
OPI_DDNEVER	13	40	3
OPI_DDNOREST	13	20	3
OPI_DDNUMB	C		2
OPI_DDPROT	13	80	3
OPI_DDRUNID	10		2
OPI_DDTYPE	12		2
OPI_DEVTYPE	43		2
OPI_DSNAME	4		2
OPI_DSNCLEAN	44		2
OPI_DSNCLNRC	4A		2
OPI_DSNCTR	14		2
OPI_DSNDABNM	3D		2
OPI_DSNDNORM	39		2
OPI_DSNFLAGS	51		2
OPI_DSNINFO	0		1
OPI_DSNINST	30		2
OPI_DSNKOST	35		2
OPI_DSNNUMB	48		2
OPI_DSNOKST	31		2
OPI_DSNPROT	45		2
OPI_DSNRUNID	4C		2
OPI_DSNSQNUM	8		2
OPI_DSNSTAT	41		2
OPI_DSNTYPE	42		2
OPI_GDG	42	80	3
OPI_GDGRELNR	46		2
OPI_IATIME	1C		3
OPI_JBCAT	42	04	3
OPI_JCLFLVOL	51	40	3
OPI_JCLRBA	38		3
OPI_JDAY	E		3
OPI_JMONTH	C		3

Name	Hex Offset	Hex Value	Level
OPI_JOBDT	8		2
OPI_JOBID	2E		3
OPI_JOBNAME	26		3
OPI_JOBTM	10		2
OPI_JYEAR	8		3
OPI_KEY	0		2
OPI_MIGR	42	10	3
OPI_NXTDDRBA	0		2
OPI_NXTDSNRBA	0		2
OPI_NXTRUNRBA	0		2
OPI_NXTSTPRBA	0		2
OPI_OCCTOK	0		4
OPI_OPRNUM	8		4
OPI_OPRTOK	0		3
OPI_OPRUNCTR	40		3
OPI_PGMNAME	1B		2
OPI_PRSTNAME	C		2
OPI_REFBK	42	08	3
OPI_RESCHAR	24		2
OPI_RUNID	4		2
OPI_RUNINFO	0		1
OPI_RUNJOBID	18		2
OPI_SMS	42	40	3
OPI_STCAT	42	02	3
OPI_STEPCODE	16		2
OPI_STEPCTR	6		2
OPI_STEPNAME	4		2
OPI_STEPNUMB	14		2
OPI_STEPSTAT	23		2
OPI_STPINFO	0		1
OPI_STPRUNID	26		2
OPI_TOTOPSIZE	3C		3
OPI_TOTSTPCTR	42		3
OPI_VALIDREL	51	80	3
OPI_VARBYTE	44		3
OPI_VARDATA	44		2
OPI_VOLCTR	4E		2
OPI_VOLINFO	0		1
OPI_VOLNUMB	6		2
OPI_VOLSER	0		2
OPI_VSAM	42	20	3

FSAB - Secondary key task control block

Name : DCLFSAB

Function:
This segment declares the secondary key task control block structure.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	37	FSAB	SECONDARY KEYS CLASS
0	(0)	CHARACTER	10	SABOPRTK	OPERATION TOKEN
0	(0)	CHARACTER	8	SABOCCTK	OCCURRENCE TOKEN
8	(8)	SIGNED	2	SABOPRNR	OPERATION NUMBER
10	(A)	CHARACTER	27	SABRESRV	RESERVED CHARS

FSCB - Current primary index record control block

Name : DCLFSCB

Function:
This module contains the declarative of the FSCB structure which maps a generic record of the VSAM file constituting the primary index inside the database of the data store. It is used each time that an internal module of the data store requires a physical direct access to the primary index.
Inside the module, you can distinguish the primary key of the SYSOUT, that is, the key of the KSDS:
  Jobname
  Execution date
  Execution time
 SYSOUT identifier
Other data comprise:
  Structured and unstructured datafile number
  Structured and unstructured datapage RBN (relative block number)
  Structured and unstructured datapage total number
A set of these three elements (separately for structured and unstructured data) constitutes the coordinates used to correctly locate the data portion of the SYSOUT, inside both a structured and unstructured datafile.
Finally, to complete this structure, there is the secondary SYSOUT key. This comprises:
  Occurrence token
  Operation number
These data form the operation token representing, inside the cleanup and restart local data store, the alternative key to get a job log. A filler for future use closes the structure.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	77	FSCB	CPRIMARYIDXREC CONTROL BLOCK
0	(0)	CHARACTER	77	SKPIDXRC	CURRENT PRIMARY KEY
0	(0)	CHARACTER	34	FSXB	PRIMARY KEY CLASS CONTROL BLOCK
0	(0)	CHARACTER	8	SXBJOBNM	JOBNAME
8	(8)	CHARACTER	8	SXBJOBDT	
8	(8)	CHARACTER	4	SXBYEAR	YEAR OF JOBDATE
12	(C)	CHARACTER	2	SXBMONTH	MONTH OF JOBDATE
14	(E)	CHARACTER	2	SXBDAY	DAY OF JOBDATE

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
16	(10)	CHARACTER	8	SXBJOBTM	JOB TIME IN X'HHMMSSHH' FORMAT
24	(18)	CHARACTER	8	SXBJOBID	JES JOB IDENTIFIER
32	(20)	SIGNED	2	SXBDSID	SYSOUT DD IDENT. IT IS EQUAL TO JES DSID
34	(22)	UNSIGNED	1	SFSTFILN	STRUCTURED DATAFILE FILE NUMBER
35	(23)	UNSIGNED	1	SFUNFILN	UNSTRUC. DATAFILE FILE NUMBER
36	(24)	SIGNED	4	SFSTRBN	SYSOUT RBN IN STRUCTURED DATAFILE
40	(28)	SIGNED	4	SFUNRBN	SYSOUT RBN IN UNSTRUCTURED DATAFILE
44	(2C)	SIGNED	4	SKBUTOTP	TOTAL NUMB.OF PAGES FOR UNSTRUCTURED SYSOUT
48	(30)	SIGNED	4	SKBSTOTP	TOTAL NUMB.OF PAGES FOR STRUCTURED SYSOUT
52	(34)	CHARACTER	10	SABELEM	ELEM.OF SECOND. KEY
52	(34)	CHARACTER	10	SABOPRTK	OPERATION TOKEN
52	(34)	CHARACTER	8	SABOCCTK	OCCURRENCE TOKEN
60	(3C)	SIGNED	2	SABOPRNR	OPERATION NUMBER
62	(3E)	CHARACTER	15	SKPFILL	FILER TO COMPLETE PIDX RECORD

Cross reference

Name	Hex Offset	Hex Value	Level
FSCB	0		1
FSXB	0		3
SABELEM	34		3
SABOCCTK	34		5
SABOPRNR	3C		5
SABOPRTK	34		4
SFSTFILN	22		3
SFSTRBN	24		3
SFUNFILN	23		3
SFUNRBN	28		3
SKBSTOTP	30		3
SKBUTOTP	2C		3
SKPFILL	3E		3
SKPIDXRC	0		2
SXBDAY	E		5
SXBDSID	20		4
SXBJOBDT	8		4
SXBJOBID	18		4
SXBJOBNM	0		4
SXBJOBTM	10		4

Name	Hex Offset	Hex Value	Level
SXBMONTH	C		5
SXBYEAR	8		5

FSFB - Datafile control block

Name : DCLFSFB

Function:
This module contains the control block that represents the Datafile Sub Task.
There is a Datafile Sub Task for each VSAM file (SDF or UDF) defined to Data Store.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	100	FSFB	CDATAFILE CTLBLK
0	(0)	CHARACTER	4	SFBDESC	DATAFILE CTL BLOCK
4	(4)	ADDRESS	4	SDBP	MAIN SYSOUTDB CTLBLK ADDRESS
8	(8)	ADDRESS	4	SFBRLSTP	CDATAFILE REQUEST QUEUE
12	(C)	ADDRESS	4	SFBECBAP	CDATAFILE SUBTASK ATTACH
16	(10)	ADDRESS	4	SFBECBOP	CDATAFILE OPEN PROCESS
20	(14)	UNSIGNED	4	SFBECBAT	CDATAFILE SUBTASK ECB
24	(18)	UNSIGNED	4	SFBECBOC	DATAFILE OPEN COMPLETED ECB
28	(1C)	UNSIGNED	4	SFBECBRQ	CDATAFILE SUBTASK REQUEST
32	(20)	ADDRESS	4	SFBTCBP	CDATAFILE SUBTASK TCB
36	(24)	ADDRESS	4	SMBP	SPACE MAP CTLBLK ADDRESS
40	(28)	CHARACTER	8	SFBRDID	DATAFILE READER DIV
48	(30)	SIGNED	4	SFBRDSIZ	DATAFILE SIZE IN NUMBER OF
52	(34)	CHARACTER	8	SFBUTID	DATAFILE UTILITY DIV
60	(3C)	SIGNED	4	SFBUTSIZ	DATAFILE SIZE IN NUMBER OF
64	(40)	CHARACTER	8	SFBUPID	DATAFILE UPDATE DIV
72	(48)	SIGNED	4	SFBSIZE	DATAFILE SIZE IN NUMBER OF
76	(4C)	SIGNED	4	SFBSTRC	DATAFILE SUBTASK RETURN CODE
80	(50)	SIGNED	4	SFBSTRSC	DATAFILE SUBTASK RETURN CODE
84	(54)	CHARACTER	6	SFBHDPAG	HEADER PAGE CONTENTS
84	(54)	CHARACTER	4	SFBHPDES	HEADER PAGE DESCRIPTOR;
88	(58)	CHARACTER	1	SFFILTYP	DATAFILE TYPE:
89	(59)	UNSIGNED	1	SFFILNUM	DATAFILE FILE NUMBER
90	(5A)	CHARACTER	1	SFBFLAGS	DATAFILE INDICATORS
		1...		SFBIMEXF	IMPOSSIBLE EXTENSION FLAG;
		.1..		SFBOPINC	DATAFILE OPEN INCOMPLETE
		..1.		SFBRDYF	DATAFILE READY INDICATOR
		...1		SFBCLOSF	DATAFILE CLOSED STATUS
		.... 1...		SFBROOPF	READY (OPEN) FOR READER
		.... .1..		SFBFMTRQ	OPEN WITH FORMAT REQUEST:
		.... ..1.		SFBUTOPF	READY (OPEN IN READ MODE)
91	(5B)	CHARACTER	9	SFBDD	DATAFILE DDNAME STRUCTURE
91	(5B)	UNSIGNED	1	SFBDDLTH	DATAFILE DDNAME LENGTH
92	(5C)	CHARACTER	8	SFBDDN	DATAFILE DDNAME

Cross reference

Name	Hex Offset	Hex Value	Level
FSFB	0		1
SDBP	4		2
SFBCLOSF	5A	10	3
SFBDD	5B		2
SFBDDLTH	5B		3
SFBDDN	5C		3
SFBDESC	0		2
SFBECBAP	C		2
SFBECBAT	14		2
SFBECBOC	18		2
SFBECBOP	10		2
SFBECBRQ	1C		2
SFBFLAGS	5A		2
SFBFMTRQ	5A	04	3
SFBHDPAG	54		2
SFBHPDES	54		3
SFBIMEXF	5A	80	3
SFBOPINC	5A	40	3
SFBRDID	28		2
SFBRDSIZ	30		2
SFBRDYF	5A	20	3
SFBRLSTP	8		2
SFBROOPF	5A	08	3
SFBSIZE	48		2
SFBSTRC	4C		2
SFBSTRSC	50		2
SFBTCBP	20		2
SFBUPID	40		2
SFBUTID	34		2
SFBUTOPF	5A	02	3
SFBUTSIZ	3C		2
SFFILNUM	59		3
SFFILTYP	58		3
SMBP	24		2

FSIR - Secondary index record structure

Name : DCLFSIR

Function:
This module contains the declarative of the secondary index record structure.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	FSIR	SECONDARY INDEX RECORD STRUCTURE
0	(0)	CHARACTER	40	FSYB	GENERIC SECONDARY ACCESS
0	(0)	CHARACTER	38	SYBVSKEY	SECONDARY ACCESS KEY
0	(0)	UNSIGNED	1	SYBTIPSK	SECONDARY ACCESS KEY TYPE
1	(1)	CHARACTER	37	SYBGSVAL	SECONDARY ACCESS KEY VALUE
1	(1)	CHARACTER	1	SYBGSARR (37)	
1	(1)	CHARACTER	1	SYBGSBYT	SINGLE BYTE OF GENERIC
38	(26)	UNSIGNED	1	SYBGSLTH	TOTAL LENGTH OF SECONDARY
39	(27)	CHARACTER	1	RESCHAR	RESERVED CHARACTER
40	(28)	SIGNED	2	SIRPKCTR	COUNTER OF STORED PRIMARY KEYS
42	(2A)	CHARACTER	34	SKPIDXRC (*)	CURRENT PRIMARY KEY RECORD
42	(2A)	CHARACTER	34	FSXB	PRIMARY KEY CLASS CONTROL
42	(2A)	CHARACTER	8	SXBJOBNM	JOBNAME
50	(32)	CHARACTER	8	SXBJOBDT	JOBDATE
50	(32)	CHARACTER	4	SXBYEAR	YEAR OF JOBDATE
54	(36)	CHARACTER	2	SXBMONTH	MONTH OF JOBDATE
56	(38)	CHARACTER	2	SXBDAY	DAY OF JOBDATE
58	(3A)	CHARACTER	8	SXBJOBTM	JOB TIME IN X'HHMMSSHH' FORMAT
66	(42)	CHARACTER	8	SXBJOBID	JES JOB IDENTIFIER
74	(4A)	SIGNED	2	SXBDSID	SYSOUT DD IDENTIFIER

Cross reference

Name	Hex Offset	Hex Value	Level
FSIR	0		1
FSXB	2A		3
FSYB	0		2
RESCHAR	27		3
SIRPKCTR	28		2
SKPIDXRC	2A		2
SXBDAY	38		5
SXBDSID	4A		4
SXBJOBDT	32		4
SXBJOBID	42		4
SXBJOBNM	2A		4
SXBJOBTM	3A		4
SXBMONTH	36		5
SXBYEAR	32		5
SYBGSARR	1		5

Name	Hex Offset	Hex Value	Level
SYBGSBYT	1		6
SYBGSLTH	26		3
SYBGSVAL	1		4
SYBTIPSK	0		4
SYBVSKEY	0		3

FSPB - Datapage instance structure

Name : DCLFSPB

Function:
This module contains the declarative of the FSPB structure, which maps a generic datapage of a datafile, that is, a generic 4096 bytes page inside one of the linear VSAM data sets reserved to contain the SYSOUTs represented in the form of either structured or unstructured data. This structure is used by several modules of data store to read / write the SYSOUT data, to interpret, clean, import, and export it. In the datapage structure you can find:

- A prefix, reserved to identify the sysout inside a specific datafile. Within the more significant prefix data we can list, in the real order:
  - The data page status indicator, which characterizes the data as logically deleted (leftmost bit in on)
  - The primary key of the sysout
  - The RBN (relative block number) of the first page of the SYSOUT
  - The total number of pages that constitute the SYSOUT
  - The secondary key of the SYSOUT
- The information about the real space used to store into the datapage the whole SYSOUT (if a single page can contain it completely) or a portion of it. Clearly, this length excludes the prefix and itself.

The generic area reserved to contain a SYSOUT data portion is expressed as a generic array of single characters. The ability to interpret the contents of the data portion of the datapage is reserved to specific modules of the data store. A SYSOUT data portion inside an *unstructured* datafile is represented as an ordered of variable-length records. However, inside a *structured* datafile, a SYSOUT has a more complex layout, to interpret which further mapping structures are needed.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	4096	FSPB	DATAPAGE INST.STRUC
0	(0)	CHARACTER	86	SPBPRFX	DATAPAGE PREFIX STR
				DPGDELFL	DATAPAGE STATUS INDICATOR
2	(2)	SIGNED	2	RESHALFW	RESERVED HALFWORD
4	(4)	CHARACTER	34	FSXB	PRIMARY KEY CLASS CONTROL
4	(4)	CHARACTER	8	SXBJOBNM	JOBNAME
12	(C)	CHARACTER	8	SXBJOBDT	
12	(C)	CHARACTER	4	SXBYEAR	YEAR OF JOBDATE
16	(10)	CHARACTER	2	SXBMONTH	MONTH OF JOBDATE
18	(12)	CHARACTER	2	SXBDAY	DAY OF JOBDATE
20	(14)	CHARACTER	8	SXBJOBTM	JOB TIME IN X'HHMMSSHH'
28	(1C)	CHARACTER	8	SXBJOBID	JES JOB IDENTIFIER
36	(24)	SIGNED	2	SXBDSID	SYSOUT DD IDENTIFIER
40	(28)	SIGNED	4	SPBRBN	RBN OF THE FIRST SYSOUT
44	(2C)	SIGNED	4	SPBTOTPG	TOTAL NUMBER OF PAGES FOR SYSOUT
48	(30)	CHARACTER	37	FSAB	SECONDARY KEYS CLASS
48	(30)	CHARACTER	10	SABOPRTK	OPERATION TOKEN
48	(30)	CHARACTER	8	SABOCCTK	OCCURRENCE TOKEN

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
56	(38)	SIGNED	2	SABOPRNR	OPERATION NUMBER
58	(3A)	CHARACTER	27	SABRESRV	RESERVED CHARACTERS
85	(55)	CHARACTER	1	RESCHAR	RESERVED CHARACTER
86	(56)	SIGNED	2	SPBDATLT	DATA LENGTH IN DATA DATAPAGE
88	(58)	CHARACTER	4008	SPBDATA	
88	(58)	CHARACTER	1	SPBCHARY(4008)	DATA PAGE DATA
88	(58)	CHARACTER	1	DPGDCHAR	DATAPAGE SINGLEDATA

Cross reference

Name	Hex Offset	Hex Value	Level
DPGDCHAR	58		4
DPGDELFL	0	80	3
FSAB	30		3
FSPB	0		1
FSXB	4		3
RESCHAR	55		3
RESHALFW	2		3
SABOCCTK	30		5
SABOPRNR	38		5
SABOPRTK	30		4
SABRESRV	3A		4
SPBCHARY	58		3
SPBDATA	58		2
SPBDATLT	56		2
SPBPRFX	0		2
SPBRBN	28		3
SPBTOTPG	2C		3
SXBDAY	12		5
SXBDSID	24		4
SXBJOBDT	C		4
SXBJOBID	1C		4
SXBJOBNM	4		4
SXBJOBTM	14		4
SXBMONTH	10		5
SXBYEAR	C		5

FSSB - Structured datapage layout

Name : DCLFSSB

Function:
This module contains the declarative of the structured datapage structure.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	4096	FSSB	STRUCTURED DATAPAGE LAYOUT
0	(0)	CHARACTER	86	SPBPRFX	DATAPAGE PREFIX STRUCTURE
				DPGDELFL	DATAPAGE STATUS INDICATOR
2	(2)	SIGNED	2	RESHALFW	RESERVED HALFWORD
4	(4)	CHARACTER	34	FSXB	PRIMARY KEY CLASS CONTROL BLOCK
4	(4)	CHARACTER	8	SXBJOBNM	JOBNAME
12	(C)	CHARACTER	8	SXBJOBDT	
12	(C)	CHARACTER	4	SXBYEAR	YEAR OF JOBDATE
16	(10)	CHARACTER	2	SXBMONTH	MONTH OF JOBDATE
18	(12)	CHARACTER	2	SXBDAY	DAY OF JOBDATE
20	(14)	CHARACTER	8	SXBJOBTM	JOB TIME IN X'HHMMSSHH' FORMAT
28	(1C)	CHARACTER	8	SXBJOBID	JES JOB IDENTIFIER
36	(24)	SIGNED	2	SXBDSID	SYSOUT DD IDENTIFIER. IT IS EQUAL TO JES JSID
40	(28)	SIGNED	4	SPBRBN	RBN OF THE FIRST SYSOUT DATAPAGE
44	(2C)	SIGNED	4	SPBTOTPG	TOTAL NUMBER OF PAGES FOR SYSOUT
48	(30)	CHARACTER	37	FSAB	SECONDARY KEYS CLASS CONTROL BLOCK
48	(30)	CHARACTER	10	SABOPRTK	OPERATION TOKEN
48	(30)	CHARACTER	8	SABOCCTK	OCCURRENCE TOKEN
56	(38)	SIGNED	2	SABOPRNR	OPERATION NUMBER
58	(3A)	CHARACTER	27	SABRESRV	RESERVED CHARACTERS
85	(55)	CHARACTER	1	RESCHAR	RESERVED CHARACTER
86	(56)	SIGNED	2	SPBDATLT	DATA LENGTH IN DATA PAGE
88	(58)	CHARACTER	4008	SSBPAGDT	DATA PORTION OF A STRUCTURED DATA PAGE
88	(58)	CHARACTER	8	SSBPRFX	
88	(58)	CHARACTER	1	SSBSTRTY	TYPE OF STRUCTURED DATA SECTION: J=EXPANDED JCL C=RESTART AND CLEAN UP
90	(5A)	SIGNED	2	SSBEXENR	EXECUTION NUMBER. IN LOCAL DATA STORE AND ONLY FOR TYPE ′C′. IT IDENTIFIES THE REPEATED EXECUTIONS OF THE SAME OPERATION.
92	(5C)	SIGNED	2	SSBPARPG	NUMBER OF PAGES FILLED OUT FOR A SINGLE STRUCTURED DATA PORTION (SINGLE EXECUTION). IT IS A SUBTOTAL OF THE TOTAL WITHIN THE DATA PAGE PREFIX
94	(5E)	SIGNED	2	SSBPAGNR	PROGRESSIVE NUMBER OF DATA PAGE WITHIN A SINGLE STRUCTURED DATA PORTION (EXPANDED JCL OR SPECIFIC EXECUTION)

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
96	(60)	CHARACTER	4000	SSBDATA	
96	(60)	CHARACTER	1	SSBCHARY(4000)	STRUCTURED DATA PAGE CHAR ARRAY CONSTITUTING THE DATA PORTION OF STRUCTURED DATA PAGE. IT IS THE GENERIC REPRESENTATION OF STRUCTURED DATA WITHIN A DATA PAGE
96	(60)	CHARACTER	1	DPGDCHAR	DATA PAGE SINGLE DATA CHARACTER

Cross reference

Name	Hex Offset	Hex Value	Level
DPGDCHAR	60		5
DPGDELFL	0	80	3
FSAB	30		3
FSSB	0		1
FSXB	4		3
RESCHAR	55		3
RESHALFW	2		3
SABOCCTK	30		5
SABOPRNR	38		5
SABOPRTK	30		4
SABRESRV	3A		4
SPBDATLT	56		2
SPBPRFX	0		2
SPBRBN	28		3
SPBTOTPG	2C		3
SSBCHARY	60		4
SSBDATA	60		3
SSBEXENR	5A		4
SSBPAGDT	58		2
SSBPAGNR	5E		4
SSBPARPG	5C		4
SSBPRFX	58		3
SSBSTRTY	58		4
SXBDAY	12		5
SXBDSID	24		4
SXBJOBDT	C		4
SXBJOBID	1C		4
SXBJOBNM	4		4
SXBJOBTM	14		4

Name	Hex Offset	Hex Value	Level
SXBMONTH	10		5
SXBYEAR	C		5

ID - Tracker identification record

Name : DCLID

Function:
This segment maps the identification data sent to the controller from a tracker system. The id event serves 4 purposes:
- acts as 'logon' data for tracker systems.
- propagates execution critical data (configuration) of the trk to con.
- reports on loss or activation of function in the tracker system.
- can perform pulse processing.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	50	ID	ID EVENT - MAX 50 BYTES
0	(0)	CHARACTER	2	IDEYE	EYE CATCHER FOR ID
2	(2)	CHARACTER	2	IDVER	VERSION NUMBER OF ID EVENT
4	(4)	CHARACTER	8	IDFMID	FMID OR ACRONYM
12	(C)	CHARACTER	2	IDLEVEL	SERVICE LEVEL
14	(E)	CHARACTER	2	IDTYPE	TYPE OF ID EVENT WHEN SENT
16	(10)	CHARACTER	16	IDSYS	OPERATING SYSTEM
16	(10)	CHARACTER	8	IDSYSNAME	NAME OF OPSYS FROM ..ID EVENT CONSTANTS
24	(18)	CHARACTER	8	IDSYSLEV	THE LEVEL OF OP SYS | BLANK ..FROM CVTPRODN FOR MVS
32	(20)	UNSIGNED	4	IDLOG	UNDOCUMENTED AREA
36	(24)	BITSTRING	2	IDBITS	SOME FLAGS
		1...		IDASCII	ON: THIS IS AN ASCII SYS
		.1..		IDINIT	ON: SENT BY TRK START
		..1.		IDCONCON	ON: ID CONFIRMED BY CON
		...1		IDTRKCON	ON: ID CONFIRMED BY TRK
		.... 1...		IDTRKDEAD	ON: TRACKER PULSE LOST
		.... .1..		IDDEADMSG	ON: CARDIAC ARREST MSG DONE
		.... ..1.		IDTRKPULSE	ON: ID PULSED BY TRACKER OFF: RETURNED BY CONTROLLER
		1		IDCONPULSE	ON: ID PULSED BY CONTROLLER OFF: RETURNED BY TRACKER
37	(25)	BITSTRING	1	UNKNOWNV1	BITS NOT IN V1 MAP
		1...		IDGSS	2 3RD PART AUTH FLAG
		.1..		IDUNICODE	2 DATA IN UNICODE
		..11 1111		UNKNOWNV2	2 NOT IN V2 MAP
38	(26)	CHARACTER	8	IDDEST	DESTINATION NAME (ROUTEOPTS)
46	(2E)	SIGNED	2	IDCASTNUM	DTB NUMBER FROM CONTROLLER
48	(30)	SIGNED	2	IDPULSIVL	PULSE RATE (IN MINUTES)
50	(32)	CHARACTER		IDEND_LABEL	FIXED AREA END LABEL

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	2	IDALIGN	ALIGN AREA
0	(0)	CHARACTER	2	*	ALIGN 2 BYTES
2	(2)	CHARACTER		IDALIGN_END	END LABEL

ID - MVS TRACKER DATA AREA BASE ON IDEND_LABEL WHEN USED FROM MCA.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	100	IDMVS	MVS TRACKER ID DATA
0	(0)	CHARACTER	16	IDMESYS	JOB ENTRY SUBSYSTEM
0	(0)	CHARACTER	8	IDMEFMID	FMID OF THE JESX
8	(8)	CHARACTER	1	IDMEID	..FROM <SSXJES>, A/B
9	(9)	CHARACTER	1	IDMETYPEJ3	..JES3: G OR L
10	(A)	CHARACTER	6	*	RESERVED
16	(10)	CHARACTER	8	IDMNODE	TRACKER'S NJE NODE NAME
24	(18)	CHARACTER	4	IDMSYSID	SMF ID OF THIS SYSTEM

BELOW FLAGS TO BE SET BY EACH FUNCTION/SUBTASK WHEN INITIALIZED AND RUNNING. THE ID EVENT SHOULD THEN BE QUEUED TO EW.

28	(1C)	BITSTRING	4	IDMFUNC	FUNCTIONS IN TRK
28	(1C)	BITSTRING	1	IDMTASK	- STARTED TASKS (|)
		1...		IDMJCC	ON: JCC ACTIVE
		..1.		IDMRODM	ON: RODM TASK ACTIVE
		...1		IDMSUBM	ON: SUBMITTOR ACTIVE
		.... .111		*	RESERVED
29	(1D)	BITSTRING	1	*	CATM FUNCTION BITS
30	(1E)	BITSTRING	1	IDMTRACK	SOME JOB TRACKING
		1...		IDMSTEPALL	ON: STEPEVENTS(ALL)
		.1..		IDMSTEPNZERO	ON: STEPEVENTS(NZERO)
		..1.		IDMJOBHOLD	ON: HOLD ALL JOBS
		...1		IDMJOBUHOLD	ON: HOLDJOB(USER)
		 1111		*	RESERVED
31	(1F)	BITSTRING	1	IDMF4	RESERVED
31	(1F)	BITSTRING	1	*	RESERVED
32	(20)	BITSTRING	4	IDMFUNC2	RESERVED: MORE FUNCTION BITS
36	(24)	CHARACTER	32	IDMDATA	FUNCTION DATA
36	(24)	CHARACTER	4	IDMSUBS	NAME OF TRACKER SUBSYSTEM
40	(28)	CHARACTER	1	IDMEWRETC	RETCODE(HIGH | LAST)
41	(29)	CHARACTER	1	IDMPTREV	PRINTEVENTS(NO | ALL | END)
68	(44)	CHARACTER	32	IDMDATA2	SECOND DATA PORTION
68	(44)	CHARACTER	8	IDCONSTAMP	TIME IN GMT BIN..
68	(44)	CHARACTER	4	IDCONDATE	DATE OF CONTROLLER CONFIRM
72	(48)	CHARACTER	4	IDCONTIME	TIME OF ...
100	(64)	CHARACTER		IDMVS_END01	LABEL FOR V01 END ADDRESS

ID - AIX TRACKER DATA AREA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	IDAIX	AIX TRACKER ID DATA
0	(0)	CHARACTER	32	IDAIP	IP ADDRESS OF JOBSUB SERVER

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
32	(20)	UNSIGNED	4	IDAPORT	..AND PORT NUMBER OF IT
36	(24)	CHARACTER	8	IDACODEPAGE	CONTROLLER CODE PAGE
		FUNCTION FLAGS AND OPTIONS			
44	(2C)	BITSTRING	4	IDAFUNC	FUNCTIONS ACTIVE IN TRK
44	(2C)	BITSTRING	1	IDAF1	
		1...		IDAAPILL	ON: LOADLEV API ACTIVE
		.1..		IDANOJOBLOG	ON: NO JOBLOG SUPPORT
		..1.		IDAJOBLOGIMD	ON: JOBLOG IMMED
		...1		IDAEVM	2 ON: EVM ACTIVE
		 1111		*	RESERVED
45	(2D)	BITSTRING	1	IDAF2	RESERVED
45	(2D)	BITSTRING	1	*	
46	(2E)	BITSTRING	1	IDAF3	RESERVED
46	(2E)	BITSTRING	1	*	
47	(2F)	BITSTRING	1	IDAF4	RESERVED
47	(2F)	BITSTRING	1	*	
48	(30)	SIGNED	4	IDANUMSUB	NUMBER OF SUBMITTOR PROCS
52	(34)	CHARACTER	8	IDALLLEVEL	2 LOADLEVELER LEVEL | BLNK
60	(3C)	CHARACTER	8	IDAGSSLEVEL	2 GSS-API LEVEL | BLNK
68	(44)	CHARACTER	12	*	RESERVED
80	(50)	CHARACTER		IDAIX_END01	LABEL FOR V02 END ADDRESS

ID - OS/400® TRACKER DATA AREA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	68	ID400	OS/400 TRACKER ID DATA
0	(0)	BITSTRING	4	ID4FUNC	FUNCTIONS ACTIVE IN TRK
4	(4)	CHARACTER	64	ID4	...<TOBEDETERMINED>
68	(44)	CHARACTER		ID400_END01	LABEL FOR V01 END ADDRESS

| |XWA- ID - OS/2 AND WIN NT TRACKER DATA AREA (WORKGROUPS)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	76	IDAWG	AIX TRACKER ID DATA
0	(0)	CHARACTER	32	IDWIP	IP ADDRESS OF TRACKER
32	(20)	UNSIGNED	4	IDWPORT	..AND PORT NUMBER
36	(24)	CHARACTER	8	IDWCODEPAGE	CONTROLLER CODE PAGE
44	(2C)	CHARACTER	32	IDWGF	RESERVED

ID - FUNCTION LOSS DATA AREA NOTE: ALL FLOSSTYPE BITS SHOULD SET 'OFF' FOR MVS SUBTASK FAILURE/STOP.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	48	IDFROST	FUNCTION ARE LOST
0	(0)	CHARACTER	16	IDFLOSS	NAME OF (FROM STBNAME)
16	(10)	BITSTRING	4	IDFLOSSTYPE	2 FUNCTION LOSS TYPE
		1...		IDCODEPAGE	2 ON: CODEPAGE IN IDFLOSS 2 USED INSTEAD ON CON 2 REQUESTED CODEPAGE
		.1..		IDNOPULSE	2 ON: CANNOT DO PULSE PROC 2 IDFLOSS = 'PULSE'
20	(14)	CHARACTER	28	*	RESERVED

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
48	(30)	CHARACTER		IDFL_END01	LABEL FOR V01 END ADDRESS

Cross reference

Name	Hex Offset	Hex Value	Level
ID	0		1
IDAAPILL	2C	80	4
IDACODEPAGE	24		2
IDAEVM	2C	10	4
IDAFUNC	2C		2
IDAF1	2C		3
IDAF2	2D		3
IDAF3	2E		3
IDAF4	2F		3
IDAGSSLEVEL	3C		2
IDAIP	0		2
IDAIX	0		1
IDAIX_END01	50		2
IDAJOBLOGIMD	2C	20	4
IDALIGN	0		1
IDALIGN_END	2		2
IDALLLEVEL	34		2
IDANOJOBLOG	2C	40	4
IDANUMSUB	30		2
IDAPORT	20		2
IDASCII	24	80	3
IDAWG	0		1
IDBITS	24		2
IDCASTNUM	2E		2
IDCODEPAGE	10	80	3
IDCONCON	24	20	3
IDCONDATE	44		4
IDCONPULSE	24	01	3
IDCONSTAMP	44		3
IDCONTIME	48		4
IDDEADMSG	24	04	3
IDDEST	26		2
IDEND_LABEL	32		2
IDEYE	0		2
IDFL_END01	30		2

Name	Hex Offset	Hex Value	Level
IDFLOSS	0		2
IDFLOSSTYPE	10		2
IDFMID	4		2
IDFROST	0		1
IDGSS	25	80	4
IDINIT	24	40	3
IDLEVEL	C		2
IDLOG	20		2
IDMDATA	24		2
IDMDATA2	44		2
IDMEFMID	0		3
IDMEID	8		3
IDMESYS	0		2
IDMETYPEJ3	9		3
IDMEWRETC	28		3
IDMFUNC	1C		2
IDMFUNC2	20		2
IDMF4	1F		3
IDMJCC	1C	80	4
IDMJOBHOLD	1E	20	4
IDMJOBUHOLD	1E	10	4
IDMNODE	10		2
IDMPTREV	29		3
IDMRODM	1C	20	4
IDMSTEPALL	1E	80	4
IDMSTEPNZERO	1E	40	4
IDMSUBM	1C	10	4
IDMSUBS	24		3
IDMSYSID	18		2
IDMTASK	1C		3
IDMTRACK	1E		3
IDMVS	0		1
IDMVS_END01	64		2
IDNOPULSE	10	40	3
IDPULSIVL	30		2
IDSYS	10		2
IDSYSLEV	18		3
IDSYSNAME	10		3
IDTRKCON	24	10	3
IDTRKDEAD	24	08	3
IDTRKPULSE	24	02	3

Name	Hex Offset	Hex Value	Level
IDTYPE	E		2
IDUNICODE	25	40	4
IDVER	2		2
IDWCODEPAGE	24		2
IDWGF	2C		2
IDWIP	0		2
IDWPORT	20		2
ID4	4		2
ID4FUNC	0		2
ID400	0		1
ID400_END01	44		2
UNKNOWNV1	25		3
UNKNOWNV2	25	3F	4

JCLREC - Common data of JCL record

Name : DCLJCL

Function:
This segment declares a JCL record in the JCL repository, physically located in EQQJS1DS and EQQJS2DS.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	JCLREC	
0	(0)	CHARACTER	80	JCLBDY	FIXED PART OF JCL RECORD
0	(0)	CHARACTER	28	JCLKEY	
0	(0)	CHARACTER	16	JCLADID	APPLICATION NAME
16	(10)	CHARACTER	10	JCLIA	OCC INPUT ARRIVAL DATE+TIME
16	(10)	CHARACTER	6	JCLIAT1	DATE
22	(16)	CHARACTER	4	JCLIAT2	TIME
26	(1A)	SIGNED	2	JCLOPNUM	OPERATION NUMBER
28	(1C)	CHARACTER	4	JCLEYE	EYE CATCHER
32	(20)	UNSIGNED	1	JCLVERS	VERSION NUMBER
33	(21)	CHARACTER	1	*	FREE
34	(22)	CHARACTER	8	JCLJOBNM	JOBNAME
42	(2A)	CHARACTER	4	JCLWSN	WORK STATION NAME
46	(2E)	CHARACTER	10	JCLUPDAT	LAST UPDATE, DATE+TIME
46	(2E)	CHARACTER	6	JCLUPDT1	DATE
52	(34)	CHARACTER	4	JCLUPDT2	TIME
56	(38)	CHARACTER	8	JCLUSER	LAST UPDATE, USERID
64	(40)	CHARACTER	1	JCLUPTYP	UPDATING FUNCTION: L: LTP W: WSD R: READY LIST M: MCP
65	(41)	CHARACTER	1	JCLSTAT	OP. STATUS: S:SUB:ED V:SAVED T:TSAVED
66	(42)	UNSIGNED	2	JCLLINES	NO OF LINES IN RECORD
68	(44)	CHARACTER	1	JCLFLAGS	FLAGBYTE
		1...		JCLJSFND	ON = JCL READ FROM JS
		.1..		JCLEDITD	ONCE BEEN EDITED

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		..11 1111		*	NOT USED
69	(45)	CHARACTER	1	*	FREE
70	(46)	SIGNED	2	JCLVLINE	NUMBER OF VARIABLES
72	(48)	SIGNED	4	JCLSUBP	SUBPOOL FOR FREEMAIN
76	(4C)	CHARACTER	4	*	FREE
80	(50)	CHARACTER	80	JCLTAB (*)	JCL RECORDS
80	(50)	CHARACTER	72	JCLTEXT	COL 1-72 OF THE JCL RECORD
152	(98)	CHARACTER	8	JCLLNNO	COL 73-80 OF THE JCL RECORD

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	JCLVTAB (*)	
0	(0)	CHARACTER	80	JCLV	
0	(0)	CHARACTER	8	JCLVNAME	VARIABLE NAME
8	(8)	CHARACTER	16	JCLVTNAM	VARIABLE TABLE NAME
24	(18)	CHARACTER	1	JCLVTYPE	P = PROMPT, Y= SETUP,N=SUB
25	(19)	CHARACTER	1	JCLVSET	E(XIT),D(EFAULT),P(REP),VAR
26	(1A)	CHARACTER	8	JCLVUSER	USER ID
26	(1A)	CHARACTER	8	JCLVEXIT	EXIT NAME
26	(1A)	CHARACTER	8	JCLVSNAM	SETTING VARIABLE NAME
34	(22)	CHARACTER	44	JCLVVAL	VARIABLE VALUE SET
78	(4E)	SIGNED	2	JCLVLGT	LENGTH OF VALUE

Cross reference

Name	Hex Offset	Hex Value	Level
JCLADID	0		4
JCLBDY	0		2
JCLEDITD	44	40	4
JCLEYE	1C		3
JCLFLAGS	44		3
JCLIA	10		4
JCLIAT1	10		5
JCLIAT2	16		5
JCLJOBNM	22		3
JCLJSFND	44	80	4
JCLKEY	0		3
JCLLINES	42		3
JCLLNNO	98		3
JCLOPNUM	1A		4
JCLREC	0		1
JCLSTAT	41		3
JCLSUBP	48		3
JCLTAB	50		2
JCLTEXT	50		3

Name	Hex Offset	Hex Value	Level
JCLUPDAT	2E		3
JCLUPDT1	2E		4
JCLUPDT2	34		4
JCLUPTYP	40		3
JCLUSER	38		3
JCLV	0		2
JCLVERS	20		3
JCLVEXIT	1A		4
JCLVLGT	4E		3
JCLVLINE	46		3
JCLVNAME	0		3
JCLVSET	19		3
JCLVSNAM	1A		5
JCLVTAB	0		1
JCLVTNAM	8		3
JCLVTYPE	18		3
JCLVUSER	1A		3
JCLVVAL	22		3
JCLWSN	2A		3

JSLOG - Job-log record in EQQJSXDS

Name : DCLJSLOG

Function:
This segment declares the sysout passed to a controller from jcc subtask, as it is stored in the JS data set. If more then one record is required to store the information, a subsequent record is created, and a flag is set in the forgoing record.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	147	JSLOG	
0	(0)	CHARACTER	28	JSLOGKEY	KEY OF RECORD IN JS
0	(0)	CHARACTER	2	JSTYPE	TYPE (=14)
2	(2)	CHARACTER	16	JSLOGAID	APPLICATION ID
18	(12)	CHARACTER	8	JSLOGIA	OCCURRENCE INPUT ARRIVAL
18	(12)	SIGNED	4	JSLOGIADF	TREATED AS BINARY
18	(12)	CHARACTER	4	JSLOGIAD	IA DATE IN BINARY
22	(16)	SIGNED	4	JSLOGIATF	TREATED AS BINARY
22	(16)	CHARACTER	4	JSLOGIAT	IA TIME IN BINARY
26	(1A)	UNSIGNED	1	JSLOGOPN	OPERATION NUMBER
27	(1B)	UNSIGNED	1	JSLOGSEQ	SEQUENCE #, 1 REC = 01
28	(1C)	CHARACTER	1	JSLOGMORE	IF MORE RECORD FOR OPER = Y
29	(1D)	CHARACTER	2	JSLOGVERS	RECORD VERSION
31	(1F)	CHARACTER	1	*	NOT USED
32	(20)	SIGNED	4	JSLOGSZ	SIZE OF RECORD

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
36	(24)	CHARACTER	111	JSLOG14REC	COPY OF CP14 HEADER INFO
36	(24)	CHARACTER	8	JSLOG14KEY	COPY OF CP14 KEY AREA
44	(2C)	CHARACTER	85	JSLOG14BDY	COPY OF CP14 BODY +4
129	(81)	CHARACTER	2	*	STATUS
131	(83)	CHARACTER	16	JSLOGST	STATUS
147	(93)	CHARACTER		JSLOGDATA	DATA PORTION (COMPRESSED)

Cross reference

Name	Hex Offset	Hex Value	Level
JSLOG	0		1
JSLOGAID	2		3
JSLOGDATA	93		2
JSLOGIA	12		3
JSLOGIAD	12		5
JSLOGIADF	12		4
JSLOGIAT	16		5
JSLOGIATF	16		4
JSLOGKEY	0		2
JSLOGMORE	1C		2
JSLOGOPN	1A		3
JSLOGSEQ	1B		3
JSLOGST	83		3
JSLOGSZ	20		2
JSLOGVERS	1D		2
JSLOG14BDY	2C		3
JSLOG14KEY	24		3
JSLOG14REC	24		2
JSTYPE	0		3

JV - JCL variable table

Name : DCLJV

Function:
This segment defines the layout of a JCL variable table, in the JV logical file, physically located in EQQADDS

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	*	JV	JCL VARIABLE TABLE
0	(0)	CHARACTER	98	JVCOMMON	IDENTIFIER
0	(0)	CHARACTER	2	*	RESERVED FOR VSAM MODS/02
2	(2)	CHARACTER	23	JVKEY	KEY OF RECORD TABLE

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
2	(2)	CHARACTER	16	JVTABLE	JCL VARIABLE TABLE ID
18	(12)	CHARACTER	7	*	ALWAYS BLANK
25	(19)	CHARACTER	1	*	NOT USED
26	(1A)	CHARACTER	8	JVLU	LAST UPDATING USER
34	(22)	CHARACTER	4	JVLT	LAST UPDATE TIME HHMM
38	(26)	CHARACTER	6	JVLD	LAST UPDATE DATE YYMMDD
44	(2C)	CHARACTER	24	JVDES	TABLE DESCRIPTION
68	(44)	SIGNED	2	JV#VAR	NUMBER OF VARS IN TABLE
70	(46)	CHARACTER	16	JVOWN	OWNER ID
86	(56)	CHARACTER	2	*	NOT USED
88	(58)	CHARACTER	8	JVLUTS	LAST UPDATE TIMESTAMP
96	(60)	CHARACTER	2	*	NOT USED
98	(62)	CHARACTER	*	JVARSECT	VARIABLE PART OF TABLE

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	476	JVTAB	JCL VAR TABLE, VAR PART
0	(0)	CHARACTER	8	JVVAR	JCL VARIABLE NAME
8	(8)	CHARACTER	44	JVDFL	JCL VARIABLE DEF VALUE
52	(34)	CHARACTER	1	JVUC	Y = UPPERCASE, N=MIXED
53	(35)	CHARACTER	1	JVSTP	PROMPT / SETUP / SUBMIT
54	(36)	SIGNED	2	JVLG	VALUE LENGTH
56	(38)	CHARACTER	7	JVTYP	VERIFICATION TYPE
63	(3F)	CHARACTER	8	JVEX	SUBSTITUTION EXIT NAME
71	(47)	CHARACTER	1	JVINP	INPUT REQUIRED
72	(48)	SIGNED	2	JVPOS	REPLACE POSITION JCL DATA
74	(4A)	CHARACTER	1	JVNUM	NUMERIC
75	(4B)	CHARACTER	2	JVCMP	COMPARISON OPERATOR
77	(4D)	CHARACTER	44	JVPAT	VALIDATION PATTERN
121	(79)	CHARACTER	102	JVVLD	VALID VALUES
121	(79)	CHARACTER	51	JVVLD1	FIRST LINE
172	(AC)	CHARACTER	51	JVVLD2	SECOND LINE
223	(DF)	CHARACTER	204	JVTXT	DIALOG TEXT
427	(1AB)	CHARACTER	20	JVTDES	DESCRIPTION
447	(1BF)	CHARACTER	1	*	RESERVED
448	(1C0)	SIGNED	2	JVNRP	NUMBER OF DEP VALUES
450	(1C2)	CHARACTER	8	JVIND	INDEPENDENT VARIABLE NAME
458	(1CA)	CHARACTER	2	JVVERS	VERSION NUMBER
460	(1CC)	CHARACTER	2	JVSUBS	SUBSTRING START POSITION
462	(1CE)	CHARACTER	2	JVSUBL	SUBSTRING LENGTH
464	(1D0)	CHARACTER	12	*	RESERVED

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	88	JVD	DEPENDENCIES
0	(0)	CHARACTER	44	JVDIV	VALUE OF SETTING VARIABLE
44	(2C)	CHARACTER	44	JVDDV	DEPENDENT VARIABLE VALUE

Cross reference

Name	Hex Offset	Hex Value	Level
JV	0		1
JV#VAR	44		3
JVARSECT	62		2
JVCMP	4B		2
JVCOMMON	0		2
JVD	0		1
JVDDV	2C		2
JVDES	2C		3
JVDFL	8		2
JVDIV	0		2
JVEX	3F		2
JVIND	1C2		2
JVINP	47		2
JVKEY	2		3
JVLD	26		3
JVLG	36		2
JVLT	22		3
JVLU	1A		3
JVLUTS	58		3
JVNRP	1C0		2
JVNUM	4A		2
JVOWN	46		3
JVPAT	4D		2
JVPOS	48		2
JVSTP	35		2
JVSUBL	1CE		2
JVSUBS	1CC		2
JVTAB	0		1
JVTABLE	2		4
JVTDES	1AB		2
JVTXT	DF		2
JVTYP	38		2
JVUC	34		2
JVVAR	0		2
JVVERS	1CA		2
JVVLD	79		2
JVVLD1	79		3
JVVLD2	AC		3

LTP - Long-term plan

Name : DCLLTP

Function:
This is the long-term plan record layout. The main type is declared as LTP.
The second type of record is the header record.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	ltp	
0	(0)	CHARACTER	176	ltpdp	
0	(0)	CHARACTER	2	*	reserved by io-routines
2	(2)	CHARACTER	26	ltpkey	
2	(2)	CHARACTER	6	ltprday	run date ===key
8	(8)	CHARACTER	16	ltpaid	application name ===key
24	(18)	CHARACTER	4	ltpstime	input arrival time===key
28	(1C)	CHARACTER	4	ltpeye	ltp eye catcher
32	(20)	UNSIGNED	1	ltpvers	ltp version number
33	(21)	CHARACTER	8	ltpauth	authority group
41	(29)	CHARACTER	16	ltpowner	application owner id
57	(39)	CHARACTER	10	ltporia	original input arrival
57	(39)	CHARACTER	6	ltporday	original run date
63	(3F)	CHARACTER	4	ltportim	original ia time
67	(43)	CHARACTER	10	ltpdeadl	occurrence deadline
67	(43)	CHARACTER	6	ltpdday	deadline date
73	(49)	CHARACTER	4	ltpdtime	deadline time
77	(4D)	CHARACTER	1	ltpprt	priority
78	(4E)	CHARACTER	4	ltpercod	error code
82	(52)	CHARACTER	1	ltpadm	flags set by online ltp
		1...		ltpdel	marked 'DELETED' by online
		.1..		ltpadd	added by online function
		..1.		ltpmod	occ. modified online
		...1		ltpmov	rundatetime modified=move
		.... 1...		ltpdepch	dependency changed manual
		.... .1..		ltpcompl	occurrence completed in jt
		11		*	
83	(53)	CHARACTER	1	ltpflgs	
		1...		ltpoptru	moved due to optional rule
		.1..		ltpexpre	external predecessor
		..11 1111		*	free to use
84	(54)	SIGNED	4	ltppred	number of external preds
88	(58)	SIGNED	4	ltpsucc	number of external succs
92	(5C)	SIGNED	4	ltpchop	number of changed ops
96	(60)	CHARACTER	1	ltpdstat	runday status
97	(61)	CHARACTER	16	ltpjvtab	variable table name
113	(71)	CHARACTER	1	*	free
114	(72)	CHARACTER	16	ltpgroupid	group application id adga
130	(82)	CHARACTER	16	ltpcal	calendar name
146	(92)	CHARACTER	8	ltpluts	last update timestamp
154	(9A)	SIGNED	2	ltprunid	run cycle id
156	(9C)	SIGNED	4	ltpcondpred	num of ext cond preds
160	(A0)	SIGNED	4	ltpcondsucc	num of ext cond succs
164	(A4)	SIGNED	4	ltpPNDpred	num mand. pending pred
168	(A8)	CHARACTER	8	ltpfiller	free

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
176	(B0)	CHARACTER	*	ltpvars	variable section

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
176	(B0)	STRUCTURE	*	ltppreds	the predecessor sections
176	(B0)	CHARACTER	35	ltppred (*)	
176	(B0)	CHARACTER	27	ltppredi	predecessor id part
176	(B0)	CHARACTER	26	ltpprid	predecessor key part
176	(B0)	CHARACTER	6	ltpprday	run date
182	(B6)	CHARACTER	16	ltppaid	application name
198	(C6)	CHARACTER	4	ltppiat	input arrival time
202	(CA)	CHARACTER	1	ltppflgs	pred dependency flags
		1...		ltppdel	dependency deleted
		.1..		ltppadd	dependency manually added
		..1.		ltppdone	predecessor has completed
		...1		ltpMandPend	is mandatory pending
		 1111		*	free to use
203	(CB)	CHARACTER	1	ltppeye	eye catcher 'P'
204	(CC)	CHARACTER	1	ltppprMand	PCN
205	(CD)	CHARACTER	6	*	free to use

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	ltpsuccs	
0	(0)	CHARACTER	35	ltpsucc (*)	a successor section
0	(0)	CHARACTER	27	ltpsucci	successor id part
0	(0)	CHARACTER	26	ltpsuid	successor key part
0	(0)	CHARACTER	6	ltpsrday	run date
6	(6)	CHARACTER	16	ltpsaid	application name
22	(16)	CHARACTER	4	ltpsiat	input arrival time
26	(1A)	CHARACTER	1	ltpsflgs	succ dependency flags
		1...		ltpsdel	dependency deleted
		.1..		ltpsadd	dependency manually added
		..11 1111		*	free to use
27	(1B)	CHARACTER	1	ltpseye	eye catcher 'S'
28	(1C)	CHARACTER	7	*	free to use

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	ltpcondpreds	
0	(0)	CHARACTER	35	ltpcondpred (*)	cond pred section
0	(0)	CHARACTER	27	ltpcondpredi	predecessor id part
0	(0)	CHARACTER	26	ltpcondprid	predecessor key part
0	(0)	CHARACTER	6	ltpcondprday	run date
6	(6)	CHARACTER	16	ltpcondpaid	application name
22	(16)	CHARACTER	4	ltpcondpiat	input arrival time
26	(1A)	CHARACTER	1	ltpcondpflgs	pred dependency flags
		1...		ltpcondpdel	dependency deleted
		.1..		ltpcondpadd	dependency added
		..1.		ltpcondpdone	pred has completed
		...1		ltpCMandPend	is mandatory pending
		 1111		*	free to use

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
27	(1B)	CHARACTER	2	ltpcondpeye	eye catcher 'PC'
29	(1D)	CHARACTER	1	ltpcondpmand	PCN
30	(1E)	CHARACTER	5	*	free to use

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	ltpcondsuccs	
0	(0)	CHARACTER	35	ltpcondsucc (*)	cond succ section
0	(0)	CHARACTER	27	ltpcondsucci	successor id part
0	(0)	CHARACTER	26	ltpcondsuid	successor key part
0	(0)	CHARACTER	6	ltpcondsrday	run date
6	(6)	CHARACTER	16	ltpcondsaid	application name
22	(16)	CHARACTER	4	ltpcondsiat	input arrival time
26	(1A)	CHARACTER	1	ltpcondsflgs	succ dependency flags
		1...		ltpcondsdel	dependency deleted
		.1..		ltpcondsadd	dependency added
		..11 1111		*	free to use
27	(1B)	CHARACTER	2	ltpcondseye	eye catcher 'SC'
29	(1D)	CHARACTER	6	*	free to use

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	ltpchops	
0	(0)	CHARACTER	58	ltpchop (*)	changed operation data
0	(0)	CHARACTER	6	ltpchoid	operation id
0	(0)	CHARACTER	4	ltpchws	workstation name
4	(4)	SIGNED	2	ltpchoin	operation number
6	(6)	CHARACTER	10	ltpchia	op input arrival
6	(6)	CHARACTER	6	ltpchiad	op input arrival date
12	(C)	CHARACTER	4	ltpchiat	op input arrival time
16	(10)	CHARACTER	10	ltpchd	op deadline
16	(10)	CHARACTER	6	ltpchdd	op deadline date
22	(16)	CHARACTER	4	ltpchdt	op deadline time
26	(1A)	CHARACTER	24	ltpchtxt	op text description
50	(32)	CHARACTER	1	ltpscr	centralized script
51	(33)	CHARACTER	7	*	free to use

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	92	ltphead	header record
0	(0)	CHARACTER	2	*	reserved by io-routines
2	(2)	CHARACTER	26	ltphdkey	ltp key === key equal ' HEADER '
28	(1C)	CHARACTER	4	ltpheye	ltp eye catcher
32	(20)	UNSIGNED	1	ltphvers	ltp version number
33	(21)	UNSIGNED	1	ltpremSHIFT	
34	(22)	CHARACTER	2	*	free to use 71C
36	(24)	CHARACTER	6	ltpsday	ltp start day
42	(2A)	CHARACTER	6	ltpeday	ltp end day
48	(30)	CHARACTER	6	ltplup	latest update
54	(36)	CHARACTER	6	ltpfnonc	date with first non complete occurrence
60	(3C)	CHARACTER	10	ltppnen	
60	(3C)	CHARACTER	6	ltppnend	plan next period end date

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
66	(42)	CHARACTER	4	ltppnent	and time
70	(46)	SIGNED	2	ltprepl	number of replans since last plan next
72	(48)	BITSTRING	1	ltphflgs	header flags
		1...		ltpbuild	LTP build is complete
		.111 11..		*	free to use
		.... ..1.		ltpoia	all days in op ia
		1		ltpoda	all days in op dl
73	(49)	CHARACTER	3	*	free to use
76	(4C)	SIGNED	4	ltpmaxsz	(dialog) max LT size
80	(50)	CHARACTER	10	ltplevel	level of lt plan ds
80	(50)	CHARACTER	2	ltplev_ver	version of lt plan ds
82	(52)	CHARACTER	8	ltplev_fmid	fmid of lt plan ds
90	(5A)	CHARACTER	2	*	free to use

Cross reference

Name	Hex Offset	Hex Value	Level
ltp	0		1
ltpchop	5C		3
ltpcondpred	9C		3
ltpcondsucc	A0		3
ltpPNDpred	A4		3
ltppred	54		3
ltprepl	46		2
ltpsucc	58		3
ltpadd	52	40	4
ltpadm	52		3
ltpaid	8		4
ltpauth	21		3
ltpbuild	48	80	3
ltpcal	82		3
ltpchd	10		3
ltpchdd	10		4
ltpchdt	16		4
ltpchia	6		3
ltpchiad	6		4
ltpchiat	C		4
ltpchoid	0		3
ltpchoin	4		4
ltpchop	0		2
ltpchops	0		1
ltpchtxt	1A		3
ltpchws	0		4

Name	Hex Offset	Hex Value	Level
ltpCMandPend	1A	10	5
ltpcompl	52	04	4
ltpcondpadd	1A	40	5
ltpcondpaid	6		5
ltpcondpdel	1A	80	5
ltpcondpdone	1A	20	5
ltpcondpeye	1B		3
ltpcondpflgs	1A		4
ltpcondpiat	16		5
ltpcondpmand	1D		3
ltpcondprday	0		5
ltpcondpred	0		2
ltpcondpredi	0		3
ltpcondpreds	0		1
ltpcondprid	0		4
ltpcondsadd	1A	40	5
ltpcondsaid	6		5
ltpcondsdel	1A	80	5
ltpcondseye	1B		3
ltpcondsflgs	1A		4
ltpcondsiat	16		5
ltpcondsrday	0		5
ltpcondsucc	0		2
ltpcondsucci	0		3
ltpcondsuccs	0		1
ltpcondsuid	0		4
ltpdday	43		4
ltpdeadl	43		3
ltpdel	52	80	4
ltpdepch	52	08	4
ltpdp	0		2
ltpdstat	60		3
ltpdtime	49		4
ltpeday	2A		2
ltpercod	4E		3
ltpexpre	53	40	4
ltpeye	1C		3
ltpfiller	A8		3
ltpflgs	53		3
ltpfnonc	36		2
ltpgroupid	72		3

Name	Hex Offset	Hex Value	Level
ltphdkey	2		2
ltphead	0		1
ltpheye	1C		2
ltphflgs	48		2
ltphvers	20		2
ltpjvtab	61		3
ltpkey	2		3
ltplev_fmid	52		3
ltplev_ver	50		3
ltplevel	50		2
ltplup	30		2
ltpluts	92		3
ltpMandPend	CA	10	5
ltpmaxsz	4C		2
ltpmod	52	20	4
ltpmov	52	10	4
ltpoda	48	01	3
ltpoia	48	02	3
ltpoptru	53	80	4
ltporday	39		4
ltporia	39		3
ltportim	3F		4
ltpowner	29		3
ltppadd	CA	40	5
ltppaid	B6		5
ltppdel	CA	80	5
ltppdone	CA	20	5
ltppeye	CB		3
ltppflgs	CA		4
ltppiat	C6		5
ltppnen	3C		2
ltppnend	3C		3
ltppnent	42		3
ltppprMand	CC		3
ltpprday	B0		5
ltppred	B0		2
ltppredi	B0		3
ltppreds	B0		1
ltpprid	B0		4
ltpprt	4D		3
ltprday	2		4

Name	Hex Offset	Hex Value	Level
ltpremSHIFT	21		2
ltprunid	9A		3
ltpsadd	1A	40	5
ltpsaid	6		5
ltpscr	32		3
ltpsday	24		2
ltpsdel	1A	80	5
ltpseye	1B		3
ltpsflgs	1A		4
ltpsiat	16		5
ltpsrday	0		5
ltpstime	18		4
ltpsucc	0		2
ltpsucci	0		3
ltpsuccs	0		1
ltpsuid	0		4
ltpvars	B0		2
ltpvers	20		3

MCA - Main communication area

Name : DCLMCA

Function:
This segment declares the scheduler common area. Most control blocks can be reached via the MCA.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	392	mca	TWSz communication area
0	(0)	CHARACTER	4	mcadesc	block descriptor = 'MCA '
4	(4)	CHARACTER	10	mcalevel	MCA block code level
4	(4)	CHARACTER	2	mcaver	block mapping version
6	(6)	CHARACTER	8	mcafmid	TWSz tracker fmid
14	(E)	BITSTRING	2	mcaflags	mca options
		1...		mcatso	1 tso user built mca
		.1..		mcaasxb	1 asxb was modified
		..1.		mcaacee	1 acee built
		...1		mcaracf	1 subres profiles
		.... 1...		mcaesa	1 active on mvsesa
		.... .1..		mcasp4	1 mvssp 4.1 or later
		.... ..1.		mcamlog	1 msg log is reserved
		1		mca313	1 mvssp 3.1.3 or later
15	(F)	1...		mcaqfcan	1 ss cancld due to qfull
		.1..		uncompress	1 do not open parmlib
		..1.		mcaedpw	1 waiting for edp
		...1		mcasp422	1 mvssp 4.2.2 or later
		.... 1...		mcapace	1 event inflow paced

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.... .1..		mcaupace	1 emgr ask for resume
		.... ..1.		mcartrq	1 queue to DRTQ
		1		mcatcpr	1 ta task restarting

TWSz control block addresses follow

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
16	(10)	ADDRESS	4	mcaascb	eqqmajor ascb address
20	(14)	ADDRESS	4	mcamtcb	eqqmajor tcb address
24	(18)	ADDRESS	4	mcaopts	addr of options block
28	(1C)	ADDRESS	4	mcaftab	file table address
32	(20)	ADDRESS	4	mcastemj	addr 1st major subtask
36	(24)	ADDRESS	4	mcastenm	addr 1st nmm subtask
40	(28)	ADDRESS	4	mcastegs	addr 1st genserv subtask
44	(2C)	ADDRESS	4	mcaopecb	addr of oper cmd ecb
48	(30)	ADDRESS	4	mcassvt	addr of opc ssvt
52	(34)	ADDRESS	4	mcassct	addr of opc ssct
56	(38)	ADDRESS	4	mcatsob	addr of tso table block
60	(3C)	ADDRESS	4	mcanmmb	addr of nmm parameters
64	(40)	ADDRESS	4	mcaedpb	addr of edp block
68	(44)	ADDRESS	4	mcaprip	addr of pri block
72	(48)	ADDRESS	4	mcasrap	addr of serv routn array
76	(4C)	ADDRESS	4	mcasure	addr of sure table
80	(50)	ADDRESS	4	mcanabb	addr of vtam io params
84	(54)	ADDRESS	4	mcadiap	addr of diagnose options
88	(58)	ADDRESS	4	mcacpnq	addr of cp enq statistics
92	(5C)	ADDRESS	4	mcatmlog	tcb addr of mlog reserver
96	(60)	ADDRESS	4	mcajancp	js interface anchor block
100	(64)	ADDRESS	4	mcaxsip	XCF system info cb
104	(68)	ADDRESS	4	mcaqfecb	addr to q full ecb
108	(6C)	ADDRESS	4	mcaasip	APPC system info cb APPA
112	(70)	ADDRESS	4	mcaarep	APPC restart ecb ptr APPA
116	(74)	ADDRESS	4	mcaTSRAp	addr of Topology parsing
120	(78)	ADDRESS	4	mcaFSRAp	addr of Data Store D90C service routines D90A
124	(7C)	ADDRESS	4	mcasub	addr of sub parm area
128	(80)	ADDRESS	4	mcassxp	SSX block address
132	(84)	ADDRESS	4	mcarsip	RODM system info cb
136	(88)	ADDRESS	4	mcarrep	RODM restart ecb ptr
140	(8C)	ADDRESS	4	mcarodmopt	RODM options table
144	(90)	ADDRESS	4	mcahcm	HCMain address
148	(94)	ADDRESS	4	mcaID	Local Id event area
152	(98)	ADDRESS	4	mcaJopts	address of JOBOPTS parms
156	(9C)	ADDRESS	4	mcaanmmp	always nmmpointer
160	(A0)	ADDRESS	4	mcaqueptr	current queue elem ptr

TWSz service routine addresses follow

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
164	(A4)	ADDRESS	4	mcamsgx	message routine address
168	(A8)	ADDRESS	4	mcaseqx	seq io service routine
172	(AC)	ADDRESS	4	mcapdsx	pds io service routine
176	(B0)	ADDRESS	4	mcaprmx	param member parse rtn
180	(B4)	ADDRESS	4	mcaquex	queue server routine

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
184	(B8)	ADDRESS	4	mcanowx	current time routine
188	(BC)	ADDRESS	4	mcavsam	addr vsam file handler
192	(C0)	ADDRESS	4	mcavsamb	addr bex vsam file handler
196	(C4)	ADDRESS	4	mcasubx	addr job submit routine
200	(C8)	ADDRESS	4	mcarelx	addr job release routine
204	(CC)	ADDRESS	4	mcaevhx	addr of event handler
208	(D0)	ADDRESS	4	mcamcpx	addr of modify curr plan
212	(D4)	ADDRESS	4	mcabexp	bex services address
216	(D8)	ADDRESS	4	mcaaidx	ZNOWX format clone address
220	(DC)	ADDRESS	4	mcalvck	level check routine adres
224	(E0)	ADDRESS	4	mcaznqd	ZNQDX lock dsname
228	(E4)	ADDRESS	4	mcaettp	ETT info for CP04 BVBC
232	(E8)	ADDRESS	4	mcaIDecb	EW ID event ECB

miscellaneous opc constants

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
236	(EC)	SIGNED	2	mcansubs	number of subsys subtasks
238	(EE)	SIGNED	2	mcagmtof	gmt offset, minutes
240	(F0)	SIGNED	2	mcaracrtrc	racroute trace level
242	(F2)	BITSTRING	2	mcaDSTORE	Data Store task status
		1...		DBAReady	Data base init OK AYYA
		.1..		DBAFail	Data base ended AYYA
		..1.		JQUReady	Jes queue init OK AYYA
		...1		JQUFail	Jes queue ended AYYA
		.... 1...		mcaRefrCP	54C56A56A
		.... .1..		mcaMajd	diag major DGCA
		.... ..1.		mcashft	FNONC shift active
242	(F2)	BITSTRING		*	54C55C56CDGCC72C
243	(F3)	.... .1..		Fprocin	BFOA
		.... ..1.		Fmethod	Data Store method trace
		1		FParser	Data Store parser trace
244	(F4)	SIGNED	4	mcagmtSEC	gmt offset, seconds
248	(F8)	CHARACTER	1	mcajes	primary jes, A=js2, B=js3
249	(F9)	CHARACTER	1	mcacjes	jes command 1st character
250	(FA)	CHARACTER	4	mcassnm	TWSz subsystem name
254	(FE)	CHARACTER	8	mcamajnm	TWSz major enq name
262	(106)	CHARACTER	8	mcaclass	racf resource class name
270	(10E)	CHARACTER	8	mcanjenm	nje node name
278	(116)	CHARACTER	4	mcaqfqn	name of full queue
282	(11A)	CHARACTER	8	mcanvid	Netview Receiver ID
290	(122)	CHARACTER	1	mcadsclas	JES class for Datastore
291	(123)	CHARACTER	1	mcaSPIN	Y = SPIN available N = SPIN not available
292	(124)	ADDRESS	4	mcaphbp	to server block
296	(128)	CHARACTER	5	mcaclnjob	clean up job name D90A
301	(12D)	CHARACTER	1	mcaddrspc	address space type O = controllertracker S = server D = data store B = batch T = Trial EQQDTTOP CBLA L = Batch Loader CUXA U = Output Collector
302	(12E)	CHARACTER	8	MCAJesfmid	JES fmid
310	(136)	CHARACTER	1	MCAtraces	
		1...		MCAzzSPIN	SPIN traces
		.111 1111	*		
311	(137)	CHARACTER	1	mcallopt	LISTLOG option AFN
312	(138)	BITSTRING	6	mcasubrs	protected subresources

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		1...		mcaadnm	ada.adname is a resource
		.1..		mcaadow	ado.owner is a resource
		..1.		mcaadgr	adg.group is a resource
		...1		mcaadjb	adj.jobname is a resource
		.... 1...		mcacpad	cpa.adname is a resource
		.... .1..		mcacpow	cpo.owner is a resource
		.... ..1.		mcacpgr	cpg.group is a resource
		1		mcacpjb	cpj.jobname is a resource
313	(139)	1...		mcacpws	cpw.wsname is a resource
		.1..		mcajcad	jsa.adname is a resource
		..1.		mcajcjb	jsj.jobname is a resource
		...1		mcajcws	jsw.wsname is a resource
		.... 1...		mcajcow	jso.owner is a resource
		.... .1..		mcajcgr	jsg.group is a resource
		.... ..1.		mcaltad	lta.adname is a resource
		1		mcaltow	lto.owner is a resource
314	(13A)	1...		mcaoiad	oia.adname is a resource
		.1..		mcawsws	wsw.wsname is a resource
		..1.		mcarlad	rla.adname is a resource
		...1		mcarlow	rlo.owner is a resource
		.... 1...		mcarlgr	rlg.group is a resource
		.... .1..		mcarlws	rlw.wsname is a resource
		.... ..1.		mcaclcn	clc.calname is a resource
		1		mcaprpn	prp.pername is a resource
315	(13B)	1...		mcaetnm	ete.name is a resource
		.1..		mcaetad	eta.name is a resource
		..1.		mcasrnm	srs.name is a resource
		...1		mcavjvo	jv.owner is a resource
		.... 1...		mcavjvn	jv.tabname is a resource
		.... .1..		mcacpwo	cpz.wsname is a res
		.... ..1.		mcacpgd	cpd.OCCgrp is a res ADGA
		1		mcaltgd	ltd.OCCgrp is a res ADGA
316	(13C)	1...		mcaadgd	add.adgrp is a res ADGA
		.1..		mcarlwst	rl.wsstat is a res
		..1.		mcardrn	rdr.name is a res
		...1		mcaadex	ade.extname is a res
		.... 1...		mcacpex	cpe.extname is a res
		.... .1..		mcaadse	ad.secelem is a res CHWA
		.... ..1.		mcacpse	cp.secelem is a res CHWA
		1		mcadbrp	rp.reptype is a res CMIA
317	(13D)	1...		mcajld	JL.DSNAME library access
		.1..		mcajlm	JL.MEMBER member access
		..1.		mcaaduf	AD.UFVAL access control
		...1		mcacpuf	CP.UFVAL access control
		.... 1...		mcargrn	RG.RGNAME
		.... .1..		mcargow	RG.OWNER
		.... ..1.		mcacprs	cp.resname DJJA
		1		mcaadrs	ad.resname DJJC
318	(13E)	BITSTRING	2	mcaflags2	flags
		1...		mcasp52	1 mvssp 5.2 or later
		.1..		mcasymb	1 perform symbol subst
		..1.		mcaux002	exit2 invoked AESA
		...1		mcawaenq	deq after abend AESA
		.... 1...		mcadbg	for debug purpose
		.... .1..		mcajtblock	ON JTB is locked 31cCRFC
		.... ..1.		mcaTWSCntlStart	On at controller StartUp BRDa
		1		mcaBulkDiscoveryRunning	bulk disc is already runni33a

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
319	(13F)	1...		mcalock2b	CRVA
		.1..		mcastopc	On = stop command issue CZNA
		..1.		mcaRESTART	
		...1		mcaLOGRCopen	On = LOGRC open
		.... 1...		mcaLOGRCempty	On = LOGRC empty done
		.... .1..		mcaLOGRCUpend	On = unload pending
		.... ..1.		mcaLOGRCLdone	On = load done
		1		mcaMLOCKskip	DLHC
320	(140)	ADDRESS	4	mcatplgyp	Topology CB address
324	(144)	ADDRESS	4	mcaSCLIBdcb	sclib dcb ptr CBLA
328	(148)	ADDRESS	4	mcaCtoken	token for C environment CBLA
332	(14C)	SIGNED	4	mcauserf	reserved for opc exits
336	(150)	SIGNED	2	mcaquelen	QUEUELEN changed value
338	(152)	BITSTRING	2	mcaperf	Performance flags
		1...		mcaexiDB	EXIT debug
		.1..		mcajclDB	JCL debug
		..1.		mcaE105	E105 msg flag D55A
		...1		mcaZ308	Z308 msg flag p904A
		.... 1...		mcan069	N069 msg flag CTXA
		.... .1..		mcaTDChk	checkTimeDep CTXC DCRA
		.... ..1.		mcaADinuse	AD used by batch
		1		mcaADVERrun	AD VER done
339	(153)	1...		mcapif	
		.1..		mcaMLOG2	EQQMLOG2 DD specified
		..1.		mcaMLOG2inuse	EQQMLOG2 is to be used
		...1		mcaLOOP	EQQLOOP open OK
		 1111	*	free DADD	
340	(154)	SIGNED	2	mcaTimeSta	stats msg interval time
342	(156)	SIGNED	2	mcablrc	BL rc with EQQY221E CUXC
344	(158)	ADDRESS	4	MCADBGp	
348	(15C)	ADDRESS	4	mcamlogd	mlog dsname address AUNA
352	(160)	ADDRESS	4	mcafarb	farb ptr
356	(164)	SIGNED	4	lockrc	05CBVUA
360	(168)	CHARACTER	4	mcaFINDmem	CBLA
360	(168)	BITSTRING	3	mcaTTR	CBLA
363	(16B)	BITSTRING	1	mcaconc	CBLA
364	(16C)	SIGNED	4	mcaENFTOK57	ENFREQ 57 dtoken CCUA
368	(170)	SIGNED	4	mcaWLMQsz	WLM query size CCUA
372	(174)	SIGNED	4	mcaENFTOK41	ENFREQ 41 dtoken CCUA
376	(178)	ADDRESS	4	mcamsgi	bufmsg routine address CHWA
380	(17C)	ADDRESS	4	mcaEXTp	address of extended MCA CMDA
384	(180)	SIGNED	4	mcaENFTOK53	ENFREQ 53 dtoken CHMA
388	(184)	ADDRESS	4	mcaux014	address of eqqux014 CRPA
392	(188)	CHARACTER		mcaend	end of mca

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	324	mcaEXT	extended MCA CMDA
0	(0)	ADDRESS	4	mcaANCQp	adress of mcaFLque CMDA
4	(4)	CHARACTER	8	mcaSYSNAME	&SYSNAME COBA
12	(C)	ADDRESS	4	mcajtab	address of EQQZJTAB CRFA
16	(10)	ADDRESS	4	mcadsiox	address of EQQDSIOX CRFA
20	(14)	ADDRESS	4	mcadsini	address of EQQDSINI CRFA
24	(18)	ADDRESS	4	mcajtbp	ptr to JTB CRFA
28	(1C)	ADDRESS	4	mcaX14tabp	exit14 tabptr CRPA
32	(20)	SIGNED	4	mcaX14numr	exit14 numrow CRPA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
36	(24)	SIGNED	4	mcaX14rsiz	exit14 recsize CRPA
40	(28)	ADDRESS	4	mcaDSViewp	ptr to DSV command area
44	(2C)	CHARACTER	8	mcaoptmem	options member
52	(34)	ADDRESS	4	mcadtbaux	pointer to refresh dest
56	(38)	SIGNED	4	mcaavildst	destination slots available for refresh
60	(3C)	ADDRESS	4	mcahtca	pointer to htca
64	(40)	SIGNED	4	mcahtcauxn	len of aux htc
68	(44)	ADDRESS	4	mcahtcaux	pointer to auxiliary htc
72	(48)	ADDRESS	4	mcaPerfJTBp	perf measurement JTB DCAA
76	(4C)	ADDRESS	4	mcaareas	storage for JEKJNodes DCAA
80	(50)	ADDRESS	4	mcaJNOfr	first free JNODE DCAA
84	(54)	ADDRESS	4	mcaJEKfr	first free JEK entry DCAA
88	(58)	ADDRESS	4	mcaJEKfp	first job entry elem DCAA
92	(5C)	ADDRESS	4	mcaJEKlp	last job entry elem DCAA
96	(60)	ADDRESS	4	mcaJNOfp	Root JNODE DCAA
100	(64)	ADDRESS	4	mcaJNOflp	First leaf JNODE DCAA
104	(68)	ADDRESS	4	mcaJNOllp	Last Leaf JNODE DCAA
108	(6C)	ADDRESS	4	mcaJEKlastip	Last Leaf JNODE DCAA
112	(70)	SIGNED	2	mcaJEKlastipo	Last relative JE positionDCAA
114	(72)	SIGNED	2	mcaJNodeno	Number of JNODES DCAA
116	(74)	SIGNED	4	mcahtsauxn	len of aux hts
120	(78)	ADDRESS	4	mcahtsaux	pointer to auxiliary hts
124	(7C)	ADDRESS	4	mcahtsa	pointer to htsa
128	(80)	ADDRESS	4	mcaZRTAB	address of EQQZRTAB
132	(84)	ADDRESS	4	mcaZRXCM	address of EQQZRXCM
136	(88)	ADDRESS	4	mcaGCAMtoken	GCAM token
140	(8C)	CHARACTER	1	mcaCONTch	JCL continuation char
141	(8D)	CHARACTER	2	mcaVSTRch	variable prefix
143	(8F)	CHARACTER	1	mcaVENDch	free
144	(90)	ADDRESS	4	mcaSSXtrP	SSX trace DGCA
148	(94)	ADDRESS	4	mcamvdeadlp	moved deadline ptr DCRA
152	(98)	ADDRESS	4	mcaoucm	pointer to oucm
156	(9C)	ADDRESS	4	mcaplglstp	ptr to plugin list
160	(A0)	SIGNED	4	mcaG1x	
164	(A4)	SIGNED	4	mcaG1sum	
168	(A8)	SIGNED	4	mcaG2x	
172	(AC)	SIGNED	4	mcaG2sum	
176	(B0)	SIGNED	4	mcaG3x	
180	(B4)	SIGNED	4	mcaG3sum	
184	(B8)	SIGNED	4	mcaG4x	
188	(BC)	SIGNED	4	mcaG4sum	
192	(C0)	SIGNED	4	mcaG5x	
196	(C4)	SIGNED	4	mcaG5sum	
200	(C8)	ADDRESS	4	mcaxt50z	
204	(CC)	ADDRESS	4	mcamctabp	tab match crit (groups)
208	(D0)	SIGNED	4	mcaCPHFLOW	CPH flow calling dsiox DKTA
212	(D4)	ADDRESS	4	* (22)	
300	(12C)	CHARACTER	8	mcaMLOGproc	switch MLOG proc name default = EQQMLOG
308	(134)	SIGNED	4	mcaMLOG2lim	MLOG2 wrap around limit (SWITCHMLOGLIM parm)
312	(138)	SIGNED	4	mcaMLOG2cnt	MLOG2 counter
316	(13C)	ADDRESS	4	mcaCKPTp	
320	(140)	ADDRESS	4	mcaHTdbfP	free

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	12	mcaFLque	CMDA
0	(0)	ADDRESS	4	mcaANCQp1	floptmsgqu1 CMDA
4	(4)	ADDRESS	4	mcaANCQp2	floptmsgqu2 CMDA
8	(8)	ADDRESS	4	mcaANCQpT	floptmsgqut CMDA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	28	mcaDSView	
0	(0)	CHARACTER	16	mcaDSV_AD	adid
16	(10)	CHARACTER	10	mcaDSV_IA	ia
26	(1A)	SIGNED	2	mcaDSV_OP	opnum

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	48	mcaPerfJTB	DCAA
0	(0)	CHARACTER	12	mcaOperDesc	DCAA
12	(C)	SIGNED	4	mcaOperNum	DCAA
16	(10)	CHARACTER	12	mcaTimeDesc	DCAA
28	(1C)	SIGNED	4	mcaTimeNum	DCAA
32	(20)	CHARACTER	12	mcaJTBTDesc	DCAA
44	(2C)	SIGNED	4	mcaJTBTNum	DCAA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	mcaHTpage	
0	(0)	CHARACTER	12	mcaHTfix	
0	(0)	ADDRESS	4	mcaHTnext	
4	(4)	ADDRESS	4	mcaHTtot	
8	(8)	SIGNED	4	mcaHToff	
12	(C)	CHARACTER	*	mcaHTend	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	10	mcamvDeadl	moved deadline blck DCRA
0	(0)	CHARACTER	6	mcamvDlDate	Date DCRA
6	(6)	CHARACTER	4	mcamvDlTime	Time DCRA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	40	mcaSSXtr	DGCA
0	(0)	ADDRESS	4	mcaZSSXFsuse	DGCA
4	(4)	ADDRESS	4	mcaZSSXFsus2	DGCA
8	(8)	ADDRESS	4	mcaZSSXFsuse_2	DGCA
12	(C)	ADDRESS	4	mcaZSSXFsus2_2	DGCA
16	(10)	ADDRESS	4	mcaZSVTXsuse	DGCA
20	(14)	ADDRESS	4	mcaZSVTXsus2	DGCA
24	(18)	ADDRESS	4	mcaZCLNXsuse	DGCA
28	(1C)	ADDRESS	4	mcaZCLNXsus2	DGCA
32	(20)	ADDRESS	4	mcaSSXfr1	DGCA
36	(24)	CHARACTER	1	mcaSSXRest	DGCA
37	(25)	CHARACTER	3	*	DGCA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	mcamctab	
0	(0)	CHARACTER	16	mcamchdr	
0	(0)	CHARACTER	4	mcamceye	
4	(4)	SIGNED	4	mcamctotent	
8	(8)	SIGNED	4	mcamclastent	
12	(C)	SIGNED	4	mcamctabsize	
16	(10)	CHARACTER	60	mcamcdata (*)	
16	(10)	CHARACTER	16	mcamcsuccadid	
32	(20)	SIGNED	2	mcamcsuccopno	
34	(22)	CHARACTER	16	mcamcpreadid	
50	(32)	SIGNED	2	mcamcpreopno	
52	(34)	SIGNED	2	mcamccondID	
54	(36)	CHARACTER	1	mcamctype	
55	(37)	CHARACTER	1	mcamcmand	
56	(38)	CHARACTER	2	*	
58	(3A)	CHARACTER	7	mcamcIVLFrom	
58	(3A)	CHARACTER	1	mcamcIVLFr_when	
59	(3B)	CHARACTER	1	mcamcIVLFr_day	
60	(3C)	CHARACTER	3	mcamcIVLFr_HHH	
60	(3C)	CHARACTER	1	*	
61	(3D)	CHARACTER	2	mcamcIVLFr_HH	
63	(3F)	CHARACTER	2	mcamcIVLFr_MM	
65	(41)	CHARACTER	7	mcamcIVLTo	
65	(41)	CHARACTER	1	mcamcIVLTo_when	
66	(42)	CHARACTER	1	mcamcIVLTo_day	
67	(43)	CHARACTER	3	mcamcIVLTo_HHH	
67	(43)	CHARACTER	1	*	
68	(44)	CHARACTER	2	mcamcIVLTo_HH	
70	(46)	CHARACTER	2	mcamcIVLTo_MM	
72	(48)	CHARACTER	4	mcamcwde	

Cross reference

Name	Hex Offset	Hex Value	Level
DBAFail	F2	40	3
DBAReady	F2	80	3
Fmethod	F3	02	3
FParser	F3	01	3
Fprocin	F3	04	3
JQUFail	F2	10	3
JQUReady	F2	20	3
lockrc	164		2
mca	0		1
mcaacee	E	20	3
mcaadex	13C	10	3
mcaadgd	13C	80	3
mcaadgr	138	20	3
mcaADinuse	152	02	3
mcaadjb	138	10	3

Name	Hex Offset	Hex Value	Level
mcaadnm	138	80	3
mcaadow	138	40	3
mcaadrs	13D	01	3
mcaadse	13C	04	3
mcaaduf	13D	20	3
mcaADVERrun	152	01	3
mcaaidx	D8		2
mcaANCQp	0		2
mcaANCQpT	8		2
mcaANCQp1	0		2
mcaANCQp2	4		2
mcaanmmp	9C		2
mcaareas	4C		2
mcaarep	70		2
mcaascb	10		2
mcaasip	6C		2
mcaasxb	E	40	3
mcaavildst	38		2
mcabexp	D4		2
mcablrc	156		2
mcaBulkDiscoveryRunning	13E	01	3
mcacjes	F9		2
mcaCKPTp	13C		2
mcaclass	106		2
mcaclcn	13A	02	3
mcaclnjob	128		2
mcaconc	16B		3
mcaCONTch	8C		2
mcacpad	138	08	3
mcacpex	13C	08	3
mcacpgd	13B	02	3
mcacpgr	138	02	3
mcaCPHFLOW	D0		2
mcacpjb	138	01	3
mcacpnq	58		2
mcacpow	138	04	3
mcacprs	13D	02	3
mcacpse	13C	02	3
mcacpuf	13D	10	3
mcacpwo	13B	04	3
mcacpws	139	80	3

Name	Hex Offset	Hex Value	Level
mcaCtoken	148		2
mcadbg	13E	08	3
MCADBGp	158		2
mcadbrp	13C	01	3
mcaddrspc	12D		2
mcadesc	0		2
mcadiap	54		2
mcadsclas	122		2
mcadsini	14		2
mcadsiox	10		2
mcaDSTORE	F2		2
mcaDSV_AD	0		2
mcaDSV_IA	10		2
mcaDSV_OP	1A		2
mcaDSView	0		1
mcaDSViewp	28		2
mcadtbaux	34		2
mcaedpb	40		2
mcaedpw	F	20	3
mcaend	188		2
mcaENFTOK41	174		2
mcaENFTOK53	180		2
mcaENFTOK57	16C		2
mcaesa	E	08	3
mcaetad	13B	40	3
mcaetnm	13B	80	3
mcaettp	E4		2
mcaevhx	CC		2
mcaexiDB	152	80	3
mcaEXT	0		1
mcaEXTp	17C		2
mcaE105	152	20	3
mcafarb	160		2
mcaFINDmem	168		2
mcaflags	E		2
mcaflags2	13E		2
mcaFLque	0		1
mcafmid	6		3
mcaFSRAp	78		2
mcaftab	1C		2
mcaGCAMtoken	88		2

Name	Hex Offset	Hex Value	Level
mcagmtof	EE		2
mcagmtSEC	F4		2
mcaG1sum	A4		2
mcaG1x	A0		2
mcaG2sum	AC		2
mcaG2x	A8		2
mcaG3sum	B4		2
mcaG3x	B0		2
mcaG4sum	BC		2
mcaG4x	B8		2
mcaG5sum	C4		2
mcaG5x	C0		2
mcahcm	90		2
mcahtca	3C		2
mcahtcaux	44		2
mcahtcauxn	40		2
mcaHTdbfP	140		2
mcaHTend	C		2
mcaHTfix	0		2
mcaHTnext	0		3
mcaHToff	8		3
mcaHTpage	0		1
mcahtsa	7C		2
mcahtsaux	78		2
mcahtsauxn	74		2
mcaHTtot	4		3
mcaID	94		2
mcaIDecb	E8		2
mcajancp	60		2
mcajcad	139	40	3
mcajcgr	139	04	3
mcajcjb	139	20	3
mcajclDB	152	40	3
mcajcow	139	08	3
mcajcws	139	10	3
mcaJEKfp	58		2
mcaJEKfr	54		2
mcaJEKlastip	6C		2
mcaJEKlastipo	70		2
mcaJEKlp	5C		2
mcajes	F8		2

Name	Hex Offset	Hex Value	Level
MCAJesfmid	12E		2
mcajld	13D	80	3
mcajlm	13D	40	3
mcaJNodeno	72		2
mcaJNOflp	64		2
mcaJNOfp	60		2
mcaJNOfr	50		2
mcaJNOllp	68		2
mcaJopts	98		2
mcajtab	C		2
mcajtblock	13E	04	3
mcajtbp	18		2
mcaJTBTDesc	20		2
mcaJTBTNum	2C		2
mcalevel	4		2
mcallopt	137		2
mcalock2b	13F	80	3
mcaLOGRCempty	13F	08	3
mcaLOGRCLdone	13F	02	3
mcaLOGRCopen	13F	10	3
mcaLOGRCUpend	13F	04	3
mcaLOOP	153	10	3
mcaltad	139	02	3
mcaltgd	13B	01	3
mcaltow	139	01	3
mcalvck	DC		2
mcaMajd	F2	04	3
mcamajnm	FE		2
mcamccondID	34		3
mcamcdata	10		2
mcamceye	0		3
mcamchdr	0		2
mcamcIVLFr_day	3B		4
mcamcIVLFr_HH	3D		5
mcamcIVLFr_HHH	3C		4
mcamcIVLFr_MM	3F		4
mcamcIVLFr_when	3A		4
mcamcIVLFrom	3A		3
mcamcIVLTo	41		3
mcamcIVLTo_day	42		4
mcamcIVLTo_HH	44		5

Name	Hex Offset	Hex Value	Level
mcamcIVLTo_HHH	43		4
mcamcIVLTo_MM	46		4
mcamcIVLTo_when	41		4
mcamclastent	8		3
mcamcmand	37		3
mcamcpreadid	22		3
mcamcpreopno	32		3
mcamcpx	D0		2
mcamcsuccadid	10		3
mcamcsuccopno	20		3
mcamctab	0		1
mcamctabp	CC		2
mcamctabsize	C		3
mcamctotent	4		3
mcamctype	36		3
mcamcwde	48		3
mcaMLOCKskip	13F	01	3
mcamlog	E	02	3
mcamlogd	15C		2
mcaMLOGproc	12C		2
mcaMLOG2	153	40	3
mcaMLOG2cnt	138		2
mcaMLOG2inuse	153	20	3
mcaMLOG2lim	134		2
mcamsgi	178		2
mcamsgx	A4		2
mcamtcb	14		2
mcamvDeadl	0		1
mcamvdeadlp	94		2
mcamvDlDate	0		2
mcamvDlTime	6		2
mcanabb	50		2
mcanjenm	10E		2
mcanmmb	3C		2
mcanoprm	F	40	3
mcanowx	B8		2
mcansubs	EC		2
mcanvid	11A		2
mcan069	152	08	3
mcaoiad	13A	80	3
mcaopecb	2C		2

Name	Hex Offset	Hex Value	Level
mcaOperDesc	0		2
mcaOperNum	C		2
mcaoptmem	2C		2
mcaopts	18		2
mcaoucm	98		2
mcapace	F	08	3
mcapdsx	AC		2
mcaperf	152		2
mcaPerfJTB	0		1
mcaPerfJTBp	48		2
mcaphbp	124		2
mcapif	153	80	3
mcaplglstp	9C		2
mcaprip	44		2
mcaprmx	B0		2
mcaprpn	13A	01	3
mcaqfcan	F	80	3
mcaqfecb	68		2
mcaqfqn	116		2
mcaquelen	150		2
mcaqueptr	A0		2
mcaquex	B4		2
mcaracf	E	10	3
mcaracrtrc	F0		2
mcardrn	13C	20	3
mcaRefrCP	F2	08	3
mcarelx	C8		2
mcaRESTART	13F	20	3
mcargow	13D	04	3
mcargrn	13D	08	3
mcarlad	13A	20	3
mcarlgr	13A	08	3
mcarlow	13A	10	3
mcarlws	13A	04	3
mcarlwst	13C	40	3
mcarodmopt	8C		2
mcarrep	88		2
mcarsip	84		2
mcartrq	F	02	3
mcaSCLIBdcb	144		2
mcaseqx	A8		2

Name	Hex Offset	Hex Value	Level
mcashft	F2	02	3
mcaSPIN	123		2
mcasp4	E	04	3
mcasp422	F	10	3
mcasp52	13E	80	3
mcasrap	48		2
mcasrnm	13B	20	3
mcassct	34		2
mcassnm	FA		2
mcassvt	30		2
mcaSSXfr1	20		2
mcassxp	80		2
mcaSSXRest	24		2
mcaSSXtr	0		1
mcaSSXtrP	90		2
mcastegs	28		2
mcastemj	20		2
mcastenm	24		2
mcastopc	13F	40	3
mcasub	7C		2
mcasubrs	138		2
mcasubx	C4		2
mcasure	4C		2
mcasymb	13E	40	3
mcaSYSNAME	4		2
mcatcpr	F	01	3
mcaTDChk	152	04	3
mcaTimeDesc	10		2
mcaTimeNum	1C		2
mcaTimeSta	154		2
mcatmlog	5C		2
mcatplgyp	140		2
MCAtraces	136		2
mcatso	E	80	3
mcatsob	38		2
mcaTSRAp	74		2
mcaTTR	168		3
mcaTWSCntlStart	13E	02	3
mcaupace	F	04	3
mcauserf	14C		2
mcaux002	13E	20	3

Name	Hex Offset	Hex Value	Level
mcaux014	184		2
mcaVENDch	8F		2
mcaver	4		3
mcavjvn	13B	08	3
mcavjvo	13B	10	3
mcavsam	BC		2
mcavsamb	C0		2
mcaVSTRch	8D		2
mcawaenq	13E	10	3
mcaWLMQsz	170		2
mcawsws	13A	40	3
mcaxsip	64		2
mcaxt50z	C8		2
mcaX14numr	20		2
mcaX14rsiz	24		2
mcaX14tabp	1C		2
mcaZCLNXsuse	18		2
mcaZCLNXsus2	1C		2
mcaznqd	E0		2
mcaZRTAB	80		2
mcaZRXCM	84		2
mcaZSSXFsuse	0		2
mcaZSSXFsuse_2	8		2
mcaZSSXFsus2	4		2
mcaZSSXFsus2_2	C		2
mcaZSVTXsuse	10		2
mcaZSVTXsus2	14		2
MCAzzSPIN	136	80	3
mcaZ308	152	10	3
mca313	E	01	3

MFB - Missed feedback report record in DP

Name : DCLMFB

Function:
Describes layout of feedback records representing report data on operations (and occurrences) in the new current plan. The records have the standard recsize and keysize of dpin records.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	114	MFBAREA	
0	(0)	CHARACTER	114	MFBKEY	KEY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	SIGNED	2	MFBTYPE	TYPE IS ALWAYS 9
2	(2)	CHARACTER	36	*	ADJUST TO ADIDOFFS
38	(26)	CHARACTER	26	MFBOCCID	IDENTIFICATION
38	(26)	CHARACTER	16	MFBADID	ADID OF OCC OF OPERATION
54	(36)	CHARACTER	6	MFBIADT	APPL INPUT DATE
54	(36)	CHARACTER	4	*	
58	(3A)	CHARACTER	2	MFBIADD	DAY
60	(3C)	CHARACTER	4	MFBIAHM	APPL INPUT TIME
60	(3C)	CHARACTER	2	MFBIAHH	HOURS
62	(3E)	CHARACTER	2	MFBIAMM	MINUTES
64	(40)	CHARACTER	6	MFBOPID	OPERATION ID
64	(40)	CHARACTER	4	MFBOP	WORKSTATION ID AND
68	(44)	SIGNED	2	MFBOPNO	OPERATION NUMBER
70	(46)	CHARACTER	4	MFBPLDUR	PLANNED DURATION
70	(46)	CHARACTER	2	MFBPLHH	HOURS
72	(48)	CHARACTER	2	MFBPLMM	MINUTES
74	(4A)	CHARACTER	6	MFBACDUR	ACTUAL DURATION
74	(4A)	CHARACTER	4	MFBACHH	HOURS
78	(4E)	CHARACTER	2	MFBACMM	MINUTES
80	(50)	CHARACTER	1	MFBREASN	REASON FOR MISSED FDB L=LIMIT EXCEEDED O=AD RECORD IN USE
81	(51)	CHARACTER	1	MFBORIGIN	ORIGIN
82	(52)	SIGNED	2	MFBRUNID	RUN CYCLE ID
84	(54)	CHARACTER	6	MFBPLDEAD	PLANNED DEADLINE
84	(54)	CHARACTER	2	MFBPLDD	DAY OFFSET DD
86	(56)	CHARACTER	2	MFBPLDH	HOURS HH
88	(58)	CHARACTER	2	MFBPLDM	MINUTES MM
90	(5A)	CHARACTER	10	MFBACDEAD	ACTUAL DEADLINE
90	(5A)	CHARACTER	4	*	
94	(5E)	CHARACTER	2	MFBACDD	DD
96	(60)	CHARACTER	2	MFBACDH	HOURS HH
98	(62)	CHARACTER	2	MFBACDM	MINUTES MM
100	(64)	CHARACTER	14	*	TO ADD UP TO KEYSIZE

Cross reference

Name	Hex Offset	Hex Value	Level
MFBACDD	5E		4
MFBACDEAD	5A		3
MFBACDH	60		4
MFBACDM	62		4
MFBACDUR	4A		3
MFBACHH	4A		4
MFBACMM	4E		4
MFBADID	26		4
MFBAREA	0		1
MFBIADD	3A		5

Name	Hex Offset	Hex Value	Level
MFBIADT	36		4
MFBIAHH	3C		5
MFBIAHM	3C		4
MFBIAMM	3E		5
MFBKEY	0		2
MFBOCCID	26		3
MFBOP	40		4
MFBOPID	40		3
MFBOPNO	44		4
MFBORIGIN	51		3
MFBPLDD	54		4
MFBPLDH	56		4
MFBPLDM	58		4
MFBPLDEAD	54		3
MFBPLDUR	46		3
MFBPLHH	46		4
MFBPLMM	48		4
MFBREASN	50		3
MFBRUNID	52		3
MFBTYPE	0		3

MTD - Track information

Name : DCLMTD

Function:
Contains data used to update the current plan after dialog modifications or after a restart. The MTD is included in some MT0 records.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	MTD	MCP TRACK ENTRY
0	(0)	CHARACTER	4	MTDHEAD	HEADER OF ENTRY
0	(0)	SIGNED	2	MTDTYPE	TYPE OF ENTRY
2	(2)	SIGNED	2	MTDOPER	OPERATION NUMBER
4	(4)	CHARACTER	*	MTDBODY	TRACKING DATA BODY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	64	MTD_ADD_OP	ADDCHANGE OPER ENTRY (1)
0	(0)	CHARACTER	8	MTDJOBN	JOB NAME (ALSO IN JOBOPTS)
8	(8)	CHARACTER	4	MTDWSNM	WORKSTATION NAME
12	(C)	CHARACTER	24	MTDTEXT	OPERATION TEXT
36	(24)	SIGNED	2	*	free d88m
38	(26)	SIGNED	2	MTDPSUSE	PARALLEL SERVERS

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
40	(28)	SIGNED	2	MTDR1USE	RESOURCE R1 NEEDED
42	(2A)	SIGNED	2	MTDR2USE	RESOURCE R2 NEEDED
44	(2C)	CHARACTER	16	MTD_userdata	operation userdata NVIA
60	(3C)	SIGNED	4	MTDEDUR	ESTIMATED DURATION d88m
64	(40)	CHARACTER		MTD_ADD_OP_END	END OF ENTRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	52	MTD_ADD_SR	ADDCHANGE SPEC RES (3)
0	(0)	CHARACTER	44	MTD_DEL_SR	DELETE SEPC RES (4)
0	(0)	CHARACTER	44	MTDRESN	SPECIAL RESOURCE NAME
44	(2C)	CHARACTER		MTD_DEL_SR_END	END OF ENTRY
44	(2C)	CHARACTER	1	MTDREST	RESOURCE TYPE (SHREXCL)
45	(2D)	CHARACTER	1	MTDRESE	On error flag
46	(2E)	CHARACTER	1	MTDRESC	ON Complete Flag
47	(2F)	CHARACTER	1	*	free
48	(30)	SIGNED	4	MTDRESQ	RESOURCE quantity
52	(34)	CHARACTER		MTD_ADD_SR_END	END OF ENTRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	35	MTD_OPTS	JOB OPTIONS (5)
0	(0)	CHARACTER	8	MTDOPTJN	JOB NAME
8	(8)	CHARACTER	1	MTDCLASS	JOBSYSOUT CLASS
9	(9)	CHARACTER	1	MTDOPTS	JOB OPTIONS
		1...		MTDAEC	AUTOMATIC TRACKING
		.1..		MTDSUB	AUTOMATIC SUBMIT
		..1.		MTDAJR	AUTOMATIC HOLDRELEASE
		...1		MTDTJT	TIME JOB
		.... 1...		MTDCAN	SUPPRESS IF LATE
		.... .1..		MTDHRC	HIGHEST RETCODE SET
		.... ..1.		mtddwto	deadline wto TNSA
		1		mtdcondrjob	cond recovery job
10	(A)	CHARACTER	1	MTDOPTS2	MORE OPTIONS WLRA
		1...		MTDRESTA	RESTARTABLE WLRA
		.1..		MTDDIRES	RESTARTABLE BY INSTALL
		..1.		MTDRERUT	REROUTEABLE WLRA
		...1		MTDDIRER	REROUTEABLE BY INSTALL
		.... 1...		mtdexpjcl	Expanded JCL used D90C
		.... .1..		mtdusrsys	User SYSOUT used D90CCATA
		.... ..1.		mtdclasst	class present
		1		mtdformt	form present
11	(B)	CHARACTER	4	MTDHRCNE	HIGHEST RETURN CODE
11	(B)	CHARACTER	2	*	HIGHEST RETURN CODE
13	(D)	CHARACTER	2	MTDHRCNX	HIGHEST RETURN CODE
15	(F)	CHARACTER	8	MTDFORM	FORM NUMBER D23a-start

2 MTDWLMCrInd CHAR1 WLM critical indicatorxxWLM1a

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
23	(17)	CHARACTER	3	MTDOPTS3	MORE OPTIONS D90C

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
23	(17)	CHARACTER	2	MTDWLM	WLM fields
23	(17)	CHARACTER	1	MTDJOBCRT	Critical Job
24	(18)	CHARACTER	1	MTDJOBPOL	Late Job Policy
25	(19)	CHARACTER	1	MTDOPTS5	
		1...		MTDMON	Oper monitoring flag
		.1..		MTDCHGMON	Set to ON if monitoring flag value has changed
		..1.		MTDSCRIPT	free
		...1 1111		*	free D23a---end
26	(1A)	CHARACTER	9	MTDOPTS4	MORE OPTIONS D90C32C
26	(1A)	CHARACTER	1	mtdrcln	Restart & Cleanup D90C
27	(1B)	CHARACTER	8	mtdWLMclass	WLM Service Class D90C
35	(23)	CHARACTER		MTD_OPTS_END	END OF ENTRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	1	MTD_STATUS	JOB STATUS (6)
0	(0)	CHARACTER	1	MTDSTAT	NEW OPERATION STATUS
1	(1)	CHARACTER		MTD_STATUS_END	END OF ENTRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	20	MTD_TIMES	CHG OPERATION TIMES (7)
0	(0)	CHARACTER	10	MTDIA	OPR IA DATE AND TIME
0	(0)	CHARACTER	6	MTDIAD	DATE
6	(6)	CHARACTER	4	MTDIAT	TIME
10	(A)	CHARACTER	10	MTDDL	OPR DL DATE AND TIME
10	(A)	CHARACTER	6	MTDDLD	DATE
16	(10)	CHARACTER	4	MTDDLT	TIME
20	(14)	CHARACTER		MTD_TIMES_END	END OF ENTRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	36	MTD_EXT_DEP	ADDDELCH EXT DEP (8,9)
0	(0)	CHARACTER	5	MTD_INT_DEP	ADDDELCH INT DEP (10,11)
0	(0)	SIGNED	2	MTDDEPN	PREDECESSOR OPER NUMBER
2	(2)	CHARACTER	1	MTDDEPT	DEPENDENCY TYPE
3	(3)	CHARACTER		MTD_DEL_INT_END	END OF ENTRY
3	(3)	SIGNED	2	MTDTRST	TRANSPORT TIME
5	(5)	CHARACTER		MTD_ADD_INT_END	END OF ENTRY
5	(5)	CHARACTER	16	MTDDAID	PREDSUCC APPLIC NAME
21	(15)	CHARACTER	10	MTDDIA	PREDSUCC IINPUT ARRIVAL
21	(15)	CHARACTER	6	MTDDIAD	PREDSUCC IA DATE
27	(1B)	CHARACTER	4	MTDDIAT	PREDSUCC IA TIME
31	(1F)	CHARACTER	1	MTDFLAGS	FLAGS
		1...		MTDCLPRE	GROUP ADD CLOS PREC
		.1..		MTDMANDP	MANDATORY PEND PRED
		..1.		MTDSAME	mcrit = same
		...1		MTDABS	mcrit = absolute
		.... 1...		MTDREL	mcrit = relative
		.... .1..		MTDISMANDC	PIF type MAND C
		11		*	FREE

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
32	(20)	SIGNED	4	MTD_IVL	IVL RANGE (MAND PRED)
32	(20)	SIGNED	2	MTD_IVL_L	- left
34	(22)	SIGNED	2	MTD_IVL_R	- right
36	(24)	CHARACTER		MTD_DEL_EXT_END	END OF ENTRY
36	(24)	CHARACTER		MTD_ADD_EXT_END	END OF ENTRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	4	MTD_ERC	ERROR CODE CHANGE
0	(0)	CHARACTER	4	MTDERRC	NEW ERROR CODE
4	(4)	CHARACTER		MTD_ERC_END	END OF ENTRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	62	MTD_CH_CM	DELCHANGE CM CATA
0	(0)	CHARACTER	1	MTD_CH_CMACT	Action D=DISCARD CATA
1	(1)	CHARACTER	8	MTDSTEPNM	Stepname CATA
9	(9)	CHARACTER	8	MTDPROCSTP	Proc stepname or blnk
17	(11)	CHARACTER	44	MTDDSN	Dataset name CATA
61	(3D)	CHARACTER	1	MTD_disc_all	Y = discard all ds CATA
62	(3E)	CHARACTER		MTD_CH_CM_END	END OF ENTRY CATA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	14	MTD_HOP	HOP MTD part HOPA
0	(0)	CHARACTER	2	MTD_HOP_REQ	MH = Man Hold HOPA MR = Man Release HOPA NP = Nop UN = Un-nop HOPA EX = Execute HOPA BD = Rebind
2	(2)	CHARACTER	8	MTD_HOP_JBNAME	Job name HOPA
10	(A)	CHARACTER	4	MTD_HOP_WSNAME	Workstation name HOPA
14	(E)	CHARACTER		MTD_HOP_END	end of entry HOPA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	64	MTD_SLR	SLR MTD part
0	(0)	CHARACTER	8	MTD_SLR_SSTEP	Selected step
0	(0)	CHARACTER	8	mtd_slr_exdest	exec destination ---------------------------- NOTE mtd_slr_exdest is only used when adding opers from the DB2 database, so called history add. You can recognize this by inv. flag Mt0hi_add. If set, this mtd type is used to pass the exec destination from the dialog to MCP processor ----------------------------
8	(8)	CHARACTER	8	MTD_SLR_SPSTEP	Selected proc step
8	(8)	CHARACTER	8	mtd_slr_oudest	exec destination p919
16	(10)	CHARACTER	8	MTD_SLR_SESTEP	Selected end step
24	(18)	CHARACTER	8	MTD_SLR_SEPSTEP	Selected end procstep
32	(20)	CHARACTER	8	MTD_SLR_CSTEP	CM initiation step
40	(28)	CHARACTER	8	MTD_SLR_CPSTEP	CM initiation pstep
48	(30)	CHARACTER	8	MTD_SLR_CESTEP	CM initiation endstep
56	(38)	CHARACTER	8	MTD_SLR_CEPSTEP	CM initiation end pstp
64	(40)	CHARACTER		MTD_SLR_END	END OF ENTRY CATA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	200	MTD_ADD_EXTINFO	DELETE Ext.Job N. (17)
0	(0)	CHARACTER	200	MTD_DEL_EXTINFO	
0	(0)	CHARACTER	54	MTDEXTNAME	Extended Job Name
54	(36)	CHARACTER	16	mtdSEname	Scheduling Env. Name CCUA
70	(46)	CHARACTER	130	*	free CCUC
200	(C8)	CHARACTER		MTD_DEL_EXTINFO_END	END OF ENTRY
200	(C8)	CHARACTER		MTD_ADD_EXTINFO_END	END OF ENTRY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	336	MTD_ADD_AUTINFO	CHWA
0	(0)	CHARACTER	256	MTDCOMMTEXT	Command Text CHWA
0	(0)	CHARACTER	64	MTDCOMMTEX1	Command Text row 1 CHWA
64	(40)	CHARACTER	64	MTDCOMMTEX2	Command Text row 2 CHWA
128	(80)	CHARACTER	64	MTDCOMMTEX3	Command Text row 3 CHWA
192	(C0)	CHARACTER	64	MTDCOMMTEX4	Command Text row 4 CHWA
256	(100)	CHARACTER	8	MTDAUTOOPER	Automation Operator CHWA
264	(108)	CHARACTER	8	MTDSECELEM	Security Element CHWA
272	(110)	CHARACTER	64	MTDCOMPINFO	Completion Info CHWA CHWA
336	(150)	CHARACTER		MTD_ADD_AUTINFO_END	END OF ENTRY CHWA DELETE Aut Info (19)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE		MTD_DEL_AUTINFO	CHWA
0	(0)	CHARACTER		MTD_DEL_AUTINFO_END	END OF ENTRY CHWA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	32	MTD_ADD_COND	
0	(0)	SIGNED	2	MTDACOND_CID	
2	(2)	SIGNED	2	MTDACOND_TYPE	
4	(4)	SIGNED	2	MTDACOND_SIMPNO	
6	(6)	SIGNED	2	MTDACOND_COUNT	
8	(8)	CHARACTER	24	MTDACOND_DESC	
32	(20)	CHARACTER		MTD_ADD_COND_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	2	MTD_DEL_COND	
0	(0)	SIGNED	2	MTDDCOND_CID	
2	(2)	CHARACTER		MTD_DEL_COND_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	30	MTD_MOD_COND	
0	(0)	SIGNED	2	MTDMCOND_CID	
2	(2)	CHARACTER	24	MTDMCOND_DESC	
26	(1A)	SIGNED	2	MTDMCOND_TYPE	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
28	(1C)	SIGNED	2	MTDMCOND_COUNT	
30	(1E)	CHARACTER		MTD_MOD_COND_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	60	MTD_ESIMP	
0	(0)	CHARACTER	33	MTD_ISIMP	
0	(0)	SIGNED	2	MTDSIMP_CID	
2	(2)	SIGNED	2	MTDSIMP_OPNO	
4	(4)	CHARACTER	2	MTDSIMP_TYP	RC or ST
6	(6)	CHARACTER	2	MTDSIMP_LOG	RG, EQ, NE
8	(8)	CHARACTER	4	MTDSIMP_VALRC	RC1 value
12	(C)	CHARACTER	4	MTDSIMP_VALRC2	RC2 value
16	(10)	CHARACTER	1	MTDSIMP_VALST	ST value
17	(11)	CHARACTER	8	MTDSIMP_STEPN	Step name
25	(19)	CHARACTER	8	MTDSIMP_PROCS	Proc Step
33	(21)	CHARACTER		MTD_ADD_ISIMP_END	
33	(21)	CHARACTER		MTD_DEL_ISIMP_END	
33	(21)	CHARACTER	16	MTDSIMP_AID	
49	(31)	CHARACTER	10	MTDSIMP_IA	
49	(31)	CHARACTER	6	MTDSIMP_IAD	
55	(37)	CHARACTER	4	MTDSIMP_IAT	
59	(3B)	CHARACTER	1	MTDSIMPFLAGS	
		1...		MTDSIMPCLOPRE	
		.111 1111		*	
60	(3C)	CHARACTER		MTD_ADD_ESIMP_END	
60	(3C)	CHARACTER		MTD_DEL_ESIMP_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	90	MTD_EXT_CSUC	
0	(0)	SIGNED	2	MTDCSUC_CID	
2	(2)	SIGNED	2	MTDCSUC_OPNO	
4	(4)	CHARACTER	2	MTDCSUC_TYP	
6	(6)	CHARACTER	2	MTDCSUC_LOG	
8	(8)	CHARACTER	4	MTDCSUC_VALRC	
12	(C)	CHARACTER	4	MTDCSUC_VALRC2	
16	(10)	CHARACTER	1	MTDCSUC_VALST	
17	(11)	CHARACTER	8	MTDCSUC_STEPN	
25	(19)	CHARACTER	8	MTDCSUC_PROCS	
33	(21)	CHARACTER	1	*	
34	(22)	CHARACTER	16	MTDCSUC_AID	
50	(32)	CHARACTER	10	MTDCSUC_IA	
50	(32)	CHARACTER	6	MTDCSUC_IAD	
56	(38)	CHARACTER	4	MTDCSUC_IAT	
60	(3C)	CHARACTER	30	MTDCSUC_COND	
60	(3C)	CHARACTER	24	MTDCSUC_CONDDESC	
84	(54)	SIGNED	2	MTDCSUC_CONDTYPE	
86	(56)	SIGNED	2	MTDCSUC_CONDCNT	
88	(58)	SIGNED	2	MTDCSUC_CONDSNO	
90	(5A)	CHARACTER		MTD_EXT_CSUC_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	60	MTD_ESETST	
0	(0)	CHARACTER	34	MTD_ISETST	
0	(0)	SIGNED	2	MTDSETS_CID	
2	(2)	SIGNED	2	MTDSETS_OPNO	
4	(4)	CHARACTER	2	MTDSETS_TYP	
6	(6)	CHARACTER	2	MTDSETS_LOG	
8	(8)	CHARACTER	4	MTDSETS_VALRC	
12	(C)	CHARACTER	4	MTDSETS_VALRC2	
16	(10)	CHARACTER	1	MTDSETS_VALST	
17	(11)	CHARACTER	8	MTDSETS_STEPN	
25	(19)	CHARACTER	8	MTDSETS_PROCS	
33	(21)	CHARACTER	1	MTDSETS_NEWST	
34	(22)	CHARACTER		MTD_ISETST_END	
34	(22)	CHARACTER	16	MTDSETS_AID	
50	(32)	CHARACTER	10	MTDSETS_IA	
50	(32)	CHARACTER	6	MTDSETS_IAD	
56	(38)	CHARACTER	4	MTDSETS_IAT	
60	(3C)	CHARACTER		MTD_ESETST_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	70	MTD_ADD_USRF	
0	(0)	CHARACTER	16	MTDAUSRF_NAME	
16	(10)	CHARACTER	54	MTDAUSRF_VALUE	
70	(46)	CHARACTER		MTD_ADD_USRF_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	70	MTD_DEL_USRF	
0	(0)	CHARACTER	16	MTDDUSRF_NAME	
16	(10)	CHARACTER	54	MTDDUSRF_VALUE	
70	(46)	CHARACTER		MTD_DEL_USRF_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	70	MTD_MOD_USRF	
0	(0)	CHARACTER	16	MTDMUSRF_NAME	
16	(10)	CHARACTER	54	MTDMUSRF_VALUE	
70	(46)	CHARACTER		MTD_MOD_USRF_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	20	MTD_AMOZ_RENGINFO	
0	(0)	CHARACTER	16	MTDAMO_RENGZ_ADID	
16	(10)	CHARACTER	3	MTDAMO_RENGZ_OPNUM	
19	(13)	CHARACTER	1	MTDAMO_RENGZ_IFBIND	
20	(14)	CHARACTER		MTD_AMOZ_RENGINFO_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	73	MTD_AMOD_RENGINFO	
0	(0)	CHARACTER	16	MTDAMO_RENGD_WSN	
16	(10)	CHARACTER	16	MTDAMO_RENGD_ADID	
32	(20)	CHARACTER	40	MTDAMO_RENGD_JOBNAME	
72	(48)	CHARACTER	1	MTDAMO_RENGD_IFBIND	
73	(49)	CHARACTER		MTD_AMOD_RENGINFO_END	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE		MTD_DEL_RENGINFO	
0	(0)	CHARACTER		MTD_DEL_RENGINFO_END	

Cross reference

Name	Hex Offset	Hex Value	Level
MTD	0		1
MTD_ADD_AUTINFO	0		1
MTD_ADD_AUTINFO_END	150		2
MTD_ADD_COND	0		1
MTD_ADD_COND_END	20		2
MTD_ADD_ESIMP_END	3C		2
MTD_ADD_EXT_END	24		2
MTD_ADD_EXTINFO	0		1
MTD_ADD_EXTINFO_END	C8		2
MTD_ADD_INT_END	5		3
MTD_ADD_ISIMP_END	21		3
MTD_ADD_OP	0		1
MTD_ADD_OP_END	40		2
MTD_ADD_SR	0		1
MTD_ADD_SR_END	34		2
MTD_ADD_USRF	0		1
MTD_ADD_USRF_END	46		2
MTD_AMOD_RENGINFO	0		1
MTD_AMOD_RENGINFO_END	49		2
MTD_AMOZ_RENGINFO	0		1
MTD_AMOZ_RENGINFO_END	14		2
MTD_CH_CM	0		1
MTD_CH_CM_END	3E		2
MTD_CH_CMACT	0		2
MTD_DEL_AUTINFO	0		1
MTD_DEL_AUTINFO_END	0		2
MTD_DEL_COND	0		1
MTD_DEL_COND_END	2		2

Name	Hex Offset	Hex Value	Level
MTD_DEL_ESIMP_END	3C		2
MTD_DEL_EXT_END	24		2
MTD_DEL_EXTINFO	0		2
MTD_DEL_EXTINFO_END	C8		2
MTD_DEL_INT_END	3		3
MTD_DEL_ISIMP_END	21		3
MTD_DEL_RENGINFO	0		1
MTD_DEL_RENGINFO_END	0		2
MTD_DEL_SR	0		2
MTD_DEL_SR_END	2C		3
MTD_DEL_USRF	0		1
MTD_DEL_USRF_END	46		2
MTD_disc_all	3D		2
MTD_ERC	0		1
MTD_ERC_END	4		2
MTD_ESETST	0		1
MTD_ESETST_END	3C		2
MTD_ESIMP	0		1
MTD_EXT_CSUC	0		1
MTD_EXT_CSUC_END	5A		2
MTD_EXT_DEP	0		1
MTD_HOP	0		1
MTD_HOP_END	E		2
MTD_HOP_JBNAME	2		2
MTD_HOP_REQ	0		2
MTD_HOP_WSNAME	A		2
MTD_INT_DEP	0		2
MTD_ISETST	0		2
MTD_ISETST_END	22		3
MTD_ISIMP	0		2
MTD_IVL	20		2
MTD_IVL_L	20		3
MTD_IVL_R	22		3
MTD_MOD_COND	0		1
MTD_MOD_COND_END	1E		2
MTD_MOD_USRF	0		1
MTD_MOD_USRF_END	46		2
MTD_OPTS	0		1
MTD_OPTS_END	23		2
MTD_SLR	0		1
MTD_SLR_CEPSTEP	38		2

Name	Hex Offset	Hex Value	Level
MTD_SLR_CESTEP	30		2
MTD_SLR_CPSTEP	28		2
MTD_SLR_CSTEP	20		2
MTD_SLR_END	40		2
mtd_slr_exdest	0		3
mtd_slr_oudest	8		3
MTD_SLR_SEPSTEP	18		2
MTD_SLR_SESTEP	10		2
MTD_SLR_SPSTEP	8		2
MTD_SLR_SSTEP	0		2
MTD_STATUS	0		1
MTD_STATUS_END	1		2
MTD_TIMES	0		1
MTD_TIMES_END	14		2
MTD_userdata	2C		2
MTDABS	1F	10	3
MTDACOND_CID	0		2
MTDACOND_COUNT	6		2
MTDACOND_DESC	8		2
MTDACOND_SIMPNO	4		2
MTDACOND_TYPE	2		2
MTDAEC	9	80	3
MTDAJR	9	20	3
MTDAMO_RENGD_ADID	10		2
MTDAMO_RENGD_IFBIND	48		2
MTDAMO_RENGD_JOBNAME	20		2
MTDAMO_RENGD_WSN	0		2
MTDAMO_RENGZ_ADID	0		2
MTDAMO_RENGZ_IFBIND	13		2
MTDAMO_RENGZ_OPNUM	10		2
MTDAUSRF_NAME	0		2
MTDAUSRF_VALUE	10		2
MTDAUTOOPER	100		2
MTDBODY	4		2
MTDCAN	9	08	3
MTDCHGMON	19	40	4
MTDCLASS	8		2
mtdclasst	A	02	3
MTDCLPRE	1F	80	3
MTDCOMMTEXT	0		2
MTDCOMMTEX1	0		3

Name	Hex Offset	Hex Value	Level
MTDCOMMTEX2	40		3
MTDCOMMTEX3	80		3
MTDCOMMTEX4	C0		3
MTDCOMPINFO	110		2
mtdcondrjob	9	01	3
MTDCSUC_AID	22		2
MTDCSUC_CID	0		2
MTDCSUC_COND	3C		2
MTDCSUC_CONDCNT	56		3
MTDCSUC_CONDDESC	3C		3
MTDCSUC_CONDSNO	58		3
MTDCSUC_CONDTYPE	54		3
MTDCSUC_IA	32		2
MTDCSUC_IAD	32		3
MTDCSUC_IAT	38		3
MTDCSUC_LOG	6		2
MTDCSUC_OPNO	2		2
MTDCSUC_PROCS	19		2
MTDCSUC_STEPN	11		2
MTDCSUC_TYP	4		2
MTDCSUC_VALRC	8		2
MTDCSUC_VALRC2	C		2
MTDCSUC_VALST	10		2
MTDDAID	5		2
MTDDCOND_CID	0		2
MTDDEPN	0		3
MTDDEPT	2		3
MTDDIA	15		2
MTDDIAD	15		3
MTDDIAT	1B		3
MTDDIRER	A	10	3
MTDDIRES	A	40	3
MTDDL	A		2
MTDDLD	A		3
MTDDLT	10		3
MTDDSN	11		2
MTDDUSRF_NAME	0		2
MTDDUSRF_VALUE	10		2
mtddwto	9	02	3
MTDEDUR	3C		2
MTDERRC	0		2

Name	Hex Offset	Hex Value	Level
mtdexpjcl	A	08	3
MTDEXTNAME	0		3
MTDFLAGS	1F		2
MTDFORM	F		2
mtdformt	A	01	3
MTDHEAD	0		2
MTDHRC	9	04	3
MTDHRCNE	B		2
MTDHRCNX	D		3
MTDIA	0		2
MTDIAD	0		3
MTDIAT	6		3
MTDISMANDC	1F	04	3
MTDJOBCRT	17		4
MTDJOBN	0		2
MTDJOBPOL	18		4
MTDMANDP	1F	40	3
MTDMCOND_CID	0		2
MTDMCOND_COUNT	1C		2
MTDMCOND_DESC	2		2
MTDMCOND_TYPE	1A		2
MTDMON	19	80	4
MTDMUSRF_NAME	0		2
MTDMUSRF_VALUE	10		2
MTDOPER	2		3
MTDOPTJN	0		2
MTDOPTS	9		2
MTDOPTS2	A		2
MTDOPTS3	17		2
MTDOPTS4	1A		2
MTDOPTS5	19		3
MTDPROCSTP	9		2
MTDPSUSE	26		2
mtdrcln	1A		3
MTDREL	1F	08	3
MTDRERUT	A	20	3
MTDRESC	2E		2
MTDRESE	2D		2
MTDRESN	0		3
MTDRESQ	30		2
MTDREST	2C		2

Name	Hex Offset	Hex Value	Level
MTDRESTA	A	80	3
MTDR1USE	28		2
MTDR2USE	2A		2
MTDSAME	1F	20	3
MTDSCRIPT	19	20	4
MTDSECELEM	108		2
mtdSEname	36		3
MTDSETS_AID	22		2
MTDSETS_CID	0		3
MTDSETS_IA	32		2
MTDSETS_IAD	32		3
MTDSETS_IAT	38		3
MTDSETS_LOG	6		3
MTDSETS_NEWST	21		3
MTDSETS_OPNO	2		3
MTDSETS_PROCS	19		3
MTDSETS_STEPN	11		3
MTDSETS_TYP	4		3
MTDSETS_VALRC	8		3
MTDSETS_VALRC2	C		3
MTDSETS_VALST	10		3
MTDSIMP_AID	21		2
MTDSIMP_CID	0		3
MTDSIMP_IA	31		2
MTDSIMP_IAD	31		3
MTDSIMP_IAT	37		3
MTDSIMP_LOG	6		3
MTDSIMP_OPNO	2		3
MTDSIMP_PROCS	19		3
MTDSIMP_STEPN	11		3
MTDSIMP_TYP	4		3
MTDSIMP_VALRC	8		3
MTDSIMP_VALRC2	C		3
MTDSIMP_VALST	10		3
MTDSIMPCLOPRE	3B	80	3
MTDSIMPFLAGS	3B		2
MTDSTAT	0		2
MTDSTEPNM	1		2
MTDSUB	9	40	3
MTDTEXT	C		2
MTDTJT	9	10	3

Name	Hex Offset	Hex Value	Level
MTDTRST	3		3
MTDTYPE	0		3
mtdusrsys	A	04	3
MTDWLM	17		3
mtdWLMclass	1B		3
MTDWSNM	8		2

MT0 - MCP tracking information entry

Name : DCLMT0

Function:
Contains data used to update the CP with changes from the MCP dialog, PIF, API, AR, ETT.
It is also used to map the tracklog record TRL24 after a successful MCP update.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	41	mt0	mcp track entry
0	(0)	CHARACTER	4	mt0eye	eye catcher
4	(4)	CHARACTER	2	mt0ver	version
6	(6)	BITSTRING	1	mt0flgs	flag bits
		1...		mt0chk	1=check request
		.1..		mt0error	1=error is detected
		..1.		mt0drlst	1=use last occ in cp for dep resolution
		...1		mt0drlat	1=use latest occ prec added occ for dep res
		.... 1...		mt0continued	another mt0 follows
		.... .1..		mt0aerr	accept error
		.... ..1.		mt0pifad	pif supplied ad
		1		mt0fopd	faked op dates
7	(7)	CHARACTER	1	mt0type	type of change
8	(8)	CHARACTER	1	mt0caller	function calling mcp a-ar,e-ett,p-pif
9	(9)	BITSTRING	1	mt0flags	additional flags
		1...		mt0turnov	1=turnover in progress
		.1..		mt0restart	1=cp restart in progress
		..1.		mt0autodep	1=add ext dependencies
		...1		mt0resolve	1=deps must be resolved
		.... 1...		mt0cmrer	1=potential cm before CATA oper is rerun CATA
		.... .1..		mt0ignore	Ignore this MT0
		.... ..1.		mt0gr_add	This MT0 is part of a group add trnx
		1		mt0notlt	1=Dont use LT file
10	(A)	CHARACTER	10	mt0cpe	current plan end
10	(A)	CHARACTER	6	mt0cped	date and time
16	(10)	CHARACTER	4	mt0cpet	set by subsyst if err -------------------------- Mt0CPe is used to hold old IA when adding history occurrences to current plan -------------------------
20	(14)	CHARACTER	8	mt0msgid	msg, if accepted err
20	(14)	CHARACTER	8	mt0stmp	time stamp of update
20	(14)	SIGNED	4	mt0mivloff	modify ivl offsett

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
20	(14)	CHARACTER	4	mt0date	date of update
24	(18)	CHARACTER	4	mt0time	time of update
28	(1C)	SIGNED	4	mt0lgth	length of entire record -------------------------
32	(20)	CHARACTER	8	mt0OCCtok	MT0 OCC token D90A Used to decide if when D90A adding an OCC the old D90A token must be used D90A D90A '00'x - first add D90A D90A Filled - add due to D90A JT reapply use old token D90A -------------------------
40	(28)	BITSTRING	1	mt0flag3	flags D90A
		1...		mt0PSU_Appl	1 = from Apply Logic D90A
		.1..		mt0chg	COZA
		..1.		mt0chstrer	change status for rerun
		...1		mt0CAutoSuc	cond succ auto solve
		.... 1...		mt0inGroup	on = in a group
		.... .1..		mt0SJR	on = SJR
		.... ..1.		mt0xRerun	on = change occ for rerun
		1		*	free
41	(29)	CHARACTER		mt0body	body of change

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
10	(A)	STRUCTURE	8	mt0rqsst	start building MT0, or 0s
10	(A)	CHARACTER	4	mt0rqsstd	date
14	(E)	CHARACTER	4	mt0rqsstt	time

MT0 Body variable part

For ws changes

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	8	mt0ws	
0	(0)	CHARACTER	4	mt0wsn	work station name
4	(4)	CHARACTER	1	mt0rep	new reporting attribute
5	(5)	CHARACTER	1	mt0wcf	control flags
		1...		mt0wcp	control on servers
		.1..		mt0wc1	control on resource 1
		..1.		mt0wc2	control on resource 2
		...1		mt0tws	Fault-Tolerance ws
		 1111	*	spare	
6	(6)	SIGNED	2	mt0wi	number of intervals
8	(8)	CHARACTER		mt0mws	ws open intervals see dclivl(ivldata)

For ws status variation

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	12	mt0vary	
0	(0)	CHARACTER	4	mt0v_wsn	work station name wlra
4	(4)	CHARACTER	1	mt0v_stat	new ws status wlra
5	(5)	BITSTRING	1	mt0v_opt	failure options wlra
		1...		mt0v_opt_fleav	leave started wlra
		.1..		mt0v_opt_ferr	error set wlra
		..1.		mt0v_opt_frest	restart wlra
		...1		mt0v_opt_rleav	leave scheduled wlra
		.... 1...		mt0v_opt_rreru	reroute wlra
		.... .1..		mt0v_opt_rimmd	immediately available wlra
		11		*	reserved wlra
6	(6)	CHARACTER	4	mt0v_aws	alternate ws name wlra
10	(A)	CHARACTER	1	mt0v_link	link status D94A
11	(B)	BITSTRING	1	mt0v_var	status and link vary D94A
		1...		mt0v_var_link	link vary D94A
		.1..		mt0v_var_status	status vary D94A
		..11 1111		*	reserved D94A

For WS22 changes $COBA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	16	mt0ws22	COBA
0	(0)	CHARACTER	4	mt0wsn22	work station name COBA
4	(4)	CHARACTER	8	mt0dest22	work station destinationCOBA
12	(C)	CHARACTER	1	mt0rep22	new reporting attribute COBA = not used COBA
13	(D)	CHARACTER	1	mt0wcf22	control flags COBA
		1...		mt0wcp22	control on servers COBA
		.1..		mt0wc122	control on resource 1 COBA
		..1.		mt0wc222	control on resource 2 COBA
		...1		mt0tws22	Fault-Tolerance ws COBA
		 1111	*	spare COBA	
14	(E)	SIGNED	2	mt0wi22	number of intervals COBA
16	(10)	CHARACTER		mt0mws22	ws open intervals COBA see dclivl(ivldata) COBA

For WS22 status variation $COBA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	20	mt0vary22	COBA
0	(0)	CHARACTER	4	mt0v_wsn22	work station name COBA
4	(4)	CHARACTER	8	mt0v_dest22	work station destination COBA
12	(C)	CHARACTER	1	mt0v_stat22	new ws status COBA
13	(D)	BITSTRING	1	mt0v_opt22	failure options COBA
		1...		mt0v_opt_fleav22	leave started COBA
		.1..		mt0v_opt_ferr22	error set COBA
		..1.		mt0v_opt_frest22	restart COBA
		...1		mt0v_opt_rleav22	leave scheduled COBA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.... 1...		mt0v_opt_rreru22	reroute COBA
		.... .1..		mt0v_opt_rimmd22	immediately available COBA
		11	*	reserved COBA	
14	(E)	CHARACTER	4	mt0v_aws22	alternate ws name COBA
18	(12)	CHARACTER	1	mt0v_link22	link status COBA
19	(13)	BITSTRING	1	mt0v_var22	status and link vary COBA
		1...		mt0v_var_link22	link vary COBA
		.1..		mt0v_var_stat22	status vary COBA
		..11 1111		*	reserved COBA

For Occurrence changes adddeletemodify

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	96	mt0occ	addchange oper entry
0	(0)	CHARACTER	26	mt0key	occurrence key
0	(0)	CHARACTER	16	mt0aid	application name
16	(10)	CHARACTER	10	mt0im	modified input arrival,
16	(10)	CHARACTER	6	mt0imd	the new value if to be
22	(16)	CHARACTER	4	mt0imt	changed in change occ req
26	(1A)	CHARACTER	10	mt0ia	occurrence identifier ia
26	(1A)	CHARACTER	6	mt0iad	same as mt0im if not a
32	(20)	CHARACTER	4	mt0iat	change occurrence request
36	(24)	CHARACTER	10	mt0dl	occurrence deadline
36	(24)	CHARACTER	6	mt0dld	deadline date
42	(2A)	CHARACTER	4	mt0dlt	deadline time
46	(2E)	CHARACTER	1	mt0pri	priority
47	(2F)	CHARACTER	4	mt0err	error code
51	(33)	BITSTRING	1	mt0ocflg	general flag
		1...		mt0imset	pif provided IANEW'
		.1..		mt0dlset	pif provided DEADLINE
		..1.		mt0Reas	reason block exists
		...1		mt0hi_add	History occurrence add
		...1		mt0hi_chg	History occurrence chg
		.... 1...		mt0Toler	tolerate d19a
		.... .1..		mt0chgallmon	chg monitoring flag for all operation in occur
		.... ..1.		mt0mon	oper monit flag value if mt0chgallmon=ON
		1		mt0GroupM	ON if Occ of 01Cd19c Group is modified
52	(34)	CHARACTER	16	mt0jvt	jcl var table name jcla
68	(44)	CHARACTER	16	mt0GroupDef	Occurrence Group Def ADGA
84	(54)	BITSTRING	1	mt0ocflg1	
		1...		mt0mheld	set all the operation = mt0mheldV value
		.1..		mt0mheldV	mheld value
		..11 1111		*	free
85	(55)	CHARACTER	11	mt0cflg2	free
96	(60)	CHARACTER		mt0ocd	operation details see dclmtd

For Occurrence RERUN

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	336	mt0rstrt	section on restart
0	(0)	CHARACTER	8	mt0rsJob	job name
8	(8)	CHARACTER	4	mt0rsErr	error code
12	(C)	CHARACTER	16	mt0rsUsr	user data field
28	(1C)	CHARACTER	300	mt0rsReason	reason for rerun
328	(148)	CHARACTER	8	mt0rsPanel	panel where reason text entered
336	(150)	CHARACTER		mt0rod	operation details

For Occurrence ADD

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	206	mt0aoc	added occurrence info
0	(0)	CHARACTER	30	mt0txt	application text
30	(1E)	CHARACTER	8	mt0grp	authority group
38	(26)	CHARACTER	16	mt0oid	application owner
54	(36)	CHARACTER	30	mt0otx	owner text
84	(54)	CHARACTER	16	mt0cal	calendar name
100	(64)	CHARACTER	44	mt0ettcrit	ETT criteria
144	(90)	CHARACTER	1	mt0etttyp	ETT type J or R
145	(91)	CHARACTER	8	mt0ettjob	ETT job name
153	(99)	CHARACTER	8	mt0ettjid	ETT job id
161	(A1)	UNSIGNED	1	mt0ettgrootl	ETT gdg root len
162	(A2)	CHARACTER	44	mt0ettevnam	ETT full event name
206	(CE)	CHARACTER		mt0aod	operation details see dclmtd

Occurrence Group Process MT0

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	356	Mt0Group	Group Mt0
0	(0)	CHARACTER	4	Mt0GroupReq	Request type, see constants
4	(4)	CHARACTER	16	Mt0GroupId	NameReference of this Group
20	(14)	CHARACTER	16	Mt0GroupCal	Group Calendar
36	(24)	CHARACTER	4	Mt0GroupErr	Error returned by Group MCP
40	(28)	CHARACTER	4	Mt0GroupCause	Cause of error set by MC0
44	(2C)	CHARACTER	302	Mt0GroupMt0	Global change data &period..in group manner &period..for all OCCs
346	(15A)	CHARACTER	2	*	free
348	(15C)	SIGNED	4	Mt0GroupNum	Number of Occs in list
352	(160)	BITSTRING	4	Mt0GroupFlag	Processing flags - group Mt0
		1...		Mt0GroupF_Init	on LogAhead record for group
		.1..		Mt0GroupF_Wait	on Wait for completion req
		..1.		Mt0GroupF_Comp	on This is a completion req
		...1		Mt0GroupF_Err	on Updates failed
		.... 1...		Mt0GroupF_Mt0	on Induced Mt0s follows
		.... .1..		Mt0GroupF_ADDc	on CANCELLED compl ADD
		.... ..1.		Mt0GroupF_ADDd	on DELETE in compl ADD
		1		Mt0GroupF_OPCt	on OPC terminated conversat
353	(161)	1...		MT0GroupF_RDH	on Some reldelHold failed

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.1..		Mt0GroupF_Dep	on Resol deps grp inter - global data changes flags
		..1.		Mt0GroupFC_Group	on change Group Def id global
		...1		Mt0GroupFC_IA	on change IA globally
		.... 1...		Mt0GroupFC_DL	on change DeadLine globally
		.... .1..		Mt0GroupFC_Pri	on Change priority globally
		.... ..1.		Mt0GroupFC_Err	on Change error code globally
		1		Mt0GroupFC_JVT	on Change JCL Var tab globally
354	(162)	1...		*	reserved
		.1..		*	reserved
		..1.		*	reserved
		...1		*	reserved
		.... 1...		*	reserved
356	(164)	CHARACTER		Mt0GroupEnd	End of header

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	28	Mt0GroupList (*)	List of Occ's subject to Mt0 based on end of header
0	(0)	CHARACTER	16	Mt0GroupOcc	Occurrence AD name
16	(10)	CHARACTER	10	Mt0GroupIa	IA of group OCC
16	(10)	CHARACTER	6	Mt0GroupIaDate	Input Arrival date
22	(16)	CHARACTER	4	Mt0GroupIaTime	&period..time
26	(1A)	CHARACTER	1	*	Reserved
27	(1B)	BITSTRING	1	Mt0GroupOccFlg	Processing indicators
		1...		Mt0GroupOcc_Bad	On This OCC causes error
		.1..		Mt0GroupOcc_Mt0	On Specific dialog MT0 follows
		..1.		Mt0GroupOcc_Mov	On OCC modified vs ADD Mt0
		...1		Mt0GroupOcc_Del	On OCC is del&colon.d vs ADD Mt0
		.... 1...		Mt0GroupOcc_Dep	On&colon.OCC dep. ignored ARQA
		.... .1..		Mt0GroupOcc_Unr	On&colon.OCC dep unresolved

Cross reference

Name	Hex Offset	Hex Value	Level
mt0	0		1
mt0aerr	6	04	3
mt0aid	0		3
mt0aoc	0		1
mt0aod	CE		2
mt0autodep	9	20	3
mt0body	29		2
mt0cal	54		2
mt0caller	8		2
mt0CAutoSuc	28	10	3
mt0cflg2	55		2
mt0chg	28	40	3
mt0chgallmon	33	04	3

Name	Hex Offset	Hex Value	Level
mt0chk	6	80	3
mt0chstrer	28	20	3
mt0cmrer	9	08	3
mt0continued	6	08	3
mt0cpe	A		2
mt0cped	A		3
mt0cpet	10		3
mt0date	14		5
mt0dest22	4		2
mt0dl	24		2
mt0dld	24		3
mt0dlset	33	40	3
mt0dlt	2A		3
mt0drlat	6	10	3
mt0drlst	6	20	3
mt0err	2F		2
mt0error	6	40	3
mt0ettcrit	64		2
mt0ettevnam	A2		2
mt0ettgrootl	A1		2
mt0ettjid	99		2
mt0ettjob	91		2
mt0etttyp	90		2
mt0eye	0		2
mt0flags	9		2
mt0flag3	28		2
mt0flgs	6		2
mt0fopd	6	01	3
mt0gr_add	9	02	3
Mt0Group	0		1
Mt0GroupCal	14		2
Mt0GroupCause	28		2
mt0GroupDef	44		2
Mt0GroupEnd	164		2
Mt0GroupErr	24		2
Mt0GroupF_ADDc	160	04	3
Mt0GroupF_ADDd	160	02	3
Mt0GroupF_Comp	160	20	3
Mt0GroupF_Dep	161	40	3
Mt0GroupF_Err	160	10	3
Mt0GroupF_Init	160	80	3

Name	Hex Offset	Hex Value	Level
Mt0GroupF_Mt0	160	08	3
Mt0GroupF_OPCt	160	01	3
MT0GroupF_RDH	161	80	3
Mt0GroupF_Wait	160	40	3
Mt0GroupFC_DL	161	08	3
Mt0GroupFC_Err	161	02	3
Mt0GroupFC_Group	161	20	3
Mt0GroupFC_IA	161	10	3
Mt0GroupFC_JVT	161	01	3
Mt0GroupFC_Pri	161	04	3
Mt0GroupFlag	160		2
Mt0GroupIa	10		2
Mt0GroupIaDate	10		3
Mt0GroupIaTime	16		3
Mt0GroupId	4		2
Mt0GroupList	0		1
mt0GroupM	33	01	3
Mt0GroupMt0	2C		2
Mt0GroupNum	15C		2
Mt0GroupOcc	0		2
Mt0GroupOcc_Bad	1B	80	3
Mt0GroupOcc_Del	1B	10	3
Mt0GroupOcc_Dep	1B	08	3
Mt0GroupOcc_Mov	1B	20	3
Mt0GroupOcc_Mt0	1B	40	3
Mt0GroupOcc_Unr	1B	04	3
Mt0GroupOccFlg	1B		2
Mt0GroupReq	0		2
mt0grp	1E		2
mt0hi_add	33	10	3
mt0hi_chg	33	10	4
mt0ia	1A		2
mt0iad	1A		3
mt0iat	20		3
mt0ignore	9	04	3
mt0im	10		3
mt0imd	10		4
mt0imset	33	80	3
mt0imt	16		4
mt0inGroup	28	08	3
mt0jvt	34		2

Name	Hex Offset	Hex Value	Level
mt0key	0		2
mt0lgth	1C		2
mt0mheld	54	80	3
mt0mheldV	54	40	3
mt0mivloff	14		4
mt0mon	33	02	3
mt0msgid	14		2
mt0mws	8		2
mt0mws22	10		2
mt0notlt	9	01	3
mt0occ	0		1
mt0OCCtok	20		2
mt0ocd	60		2
mt0ocflg	33		2
mt0ocflg1	54		2
mt0oid	26		2
mt0otx	36		2
mt0pifad	6	02	3
mt0pri	2E		2
mt0PSU_Appl	28	80	3
mt0Reas	33	20	3
mt0rep	4		2
mt0rep22	C		2
mt0resolve	9	10	3
mt0restart	9	40	3
mt0rod	150		2
mt0rqsst	A		1
mt0rqsstd	A		2
mt0rqsstt	E		2
mt0rsErr	8		2
mt0rsJob	0		2
mt0rsPanel	148		2
mt0rsReason	1C		2
mt0rstrt	0		1
mt0rsUsr	C		2
mt0SJR	28	04	3
mt0stmp	14		3
mt0time	18		4
mt0Toler	33	08	3
mt0turnov	9	80	3
mt0tws	5	10	3

Name	Hex Offset	Hex Value	Level
mt0tws22	D	10	3
mt0txt	0		2
mt0type	7		2
mt0v_aws	6		2
mt0v_aws22	E		2
mt0v_dest22	4		2
mt0v_link	A		2
mt0v_link22	12		2
mt0v_opt	5		2
mt0v_opt_ferr	5	40	3
mt0v_opt_ferr22	D	40	3
mt0v_opt_fleav	5	80	3
mt0v_opt_fleav22	D	80	3
mt0v_opt_frest	5	20	3
mt0v_opt_frest22	D	20	3
mt0v_opt_rimmd	5	04	3
mt0v_opt_rimmd22	D	04	3
mt0v_opt_rleav	5	10	3
mt0v_opt_rleav22	D	10	3
mt0v_opt_rreru	5	08	3
mt0v_opt_rreru22	D	08	3
mt0v_opt22	D		2
mt0v_stat	4		2
mt0v_stat22	C		2
mt0v_var	B		2
mt0v_var_link	B	80	3
mt0v_var_link22	13	80	3
mt0v_var_status	B	40	3
mt0v_var_stat22	13	40	3
mt0v_var22	13		2
mt0v_wsn	0		2
mt0v_wsn22	0		2
mt0vary	0		1
mt0vary22	0		1
mt0ver	4		2
mt0wi	6		2
mt0wi22	E		2
mt0wcf	5		2
mt0wcf22	D		2
mt0wcp	5	80	3
mt0wcp22	D	80	3

Name	Hex Offset	Hex Value	Level
mt0wc1	5	40	3
mt0wc122	D	40	3
mt0wc2	5	20	3
mt0wc222	D	20	3
mt0ws	0		1
mt0wsn	0		2
mt0wsn22	0		2
mt0ws22	0		1
mt0xRerun	28	02	3

NJE - JES NJE record

Name : DCLNJE

Function:
This segment declares a JES NJE node name entry record in the scheduler Side information file (EQQSIDS)

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	80	NJE	NJE NODE NAME RECORD
0	(0)	CHARACTER	64	NJEKEY	KEY
0	(0)	CHARACTER	1	NJETYPE	RECORD TYPE 4
1	(1)	CHARACTER	8	NJENAME	NAME OF JES NJE NODE
9	(9)	CHARACTER	55	*	RESERVED (ZEROES)
64	(40)	CHARACTER	16	NJEDATA	DATA
64	(40)	CHARACTER	4	NJEDESC	RECORD DESCRIPTOR = 'NJE '
68	(44)	CHARACTER	2	NJEVERS	RECORD VERSION = '01'
70	(46)	CHARACTER	1	NJEJES	JESTYPE 'A'=JES2 'B'=JES3
71	(47)	BITSTRING	1	*	RESERVED
72	(48)	CHARACTER	8	*	RESERVED
80	(50)	CHARACTER		NJEEND	END OF NJE

Cross reference

Name	Hex Offset	Hex Value	Level
NJE	0		1
NJEDATA	40		2
NJEDESC	40		3
NJEEND	50		2
NJEJES	46		3
NJEKEY	0		2
NJENAME	1		3
NJETYPE	0		3
NJEVERS	44		3

NMM - Normal mode manager parameters

Name : DCLNMM

Function:
This segment defines the Normal Mode Manager parameter block. The NMM block is created by the Normal Mode Manager task and is addressable from the MCA after NMM has started successfully.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	504	nmm	manager block
0	(0)	CHARACTER	4	nmmdesc	block descriptor = 'NMM '
4	(4)	CHARACTER	2	nmmver	block mapping version
6	(6)	BITSTRING	4	nmmflags	nmm status switches LOGC
6	(6)	BITSTRING	1	nmmflag1	nmm status flag1
		1...		nmcpopen	current plan is open
		.1..		nmjsopen	JS file is open
		..1.		nmltopen	LTP file is open
		...1		nmsiopen	SI file is open
		.... 1...		nmadopen	AD file is open
		.... .1..		nmjscopy	will copy js file OBRC
		.... ..1.		nmbackup	will take plan backup OBRC
		1		nmturnsw	turnover in progress
7	(7)	BITSTRING	1	nmmflag2	nmm status flag2
		1...		nmrfresh	refresh is requested
		.1..		nmcpeq	CP data sets are equal
		..1.		nmncp	NCP to be copied
		...1		nmcplock	current plan is locked
		.... 1...		nmchlock	checkpoint is locked
		.... .1..		nmmaract	on AR is activated
		.... ..1.		nmjsforce	force copy of js file OBRA
		1		nmmARC	on do Data Store clean up of hanging JOBLOGD55A requests (dccln) D55A
8	(8)	BITSTRING	1	nmmflag3	NMM msgeqqn051 reas flag
		1...		nmmlim	backup(nn) limit reached
		.1..		nmmto1	DP turnover1 (start)
		..1.		nmmto2	DP turnover2 (end)
		...1		nmmcmd	BACKUP cmd issued
		.... 1...		nmmrst	NMM recovery restart
		.... .1..		nmmstop	NMM normal termination
		.... ..1.		nmmjte	JT IO error
		1		nmmedp	edp full AGZCI5A
9	(9)	BITSTRING	1	nmmflag4	NMM status flag 4
		1...		nmcxopen	current plan ext is open
		.1..		nmrdopen	current plan ext is open
		..1.		nmltlock	reserved
		...1		nmadlock	reserved
		.... 1...		nmwslock	reserved
		.... .1..		nmrdlock	reserved
		.... ..1.		nmmsymst	1 symphony is arriving
		1		nmocxwsa	check CX open flag AQIA
10	(A)	CHARACTER	1	nmmstat	contains status of nmm
11	(B)	BITSTRING	2	nmmjtflg	JT logging flags LOGA
		1...		nmjtfirst	write 1st record on JTlog

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.1..		nmjtlast	write last record on JTlog
		..1.		nmjtarcf	archiving finished posted
		...1		nmjtarce	perm error on archive log
		.... 1...		nmjtevcp	cp updates on jtlog ds
		.... .1..		nmjtevany	JTlog ds is not empty
		.... ..1.		nmjtlock	JT archive ds is locked
		1		nmjtioer	write error recovery in progr
12	(C)	1...		nmjtturn	turnover has been requested
		.1..		nmjtauxt	exit11 status act-inact
		..1.		nmsymback	sym renew backup
		...1		nmmMCP	Tie change for MCP DAWA
		.... 1...		nmmUrgDQE	dont send Urg DQE DBWA
		.... .1..		nmmPermBP	1 = permanent buf pool DCOA
		11		*	free 21CCZDCDBWCDCOC
13	(D)	BITSTRING	1	nmmnoerr	noerror processing flags
		1...		nmnoproc	noerror processing requested
		.1..		nmnomemb	noerror member proc reqstd
		..1.		nmnolist	noerror table list reqstdCPWA
		...1		nmnotry	noerror stmt trial CTXA
		.... 111.		*	free CPWCCTXC
		1		nmskipin	SKIPINC command BLBA
14	(E)	BITSTRING	1	nmmflag5	D90A
		1...		nmProtDD	D90A
		.1..		nmProtDS	D90A
		..1.		nmturn2sw	TWS syncron. required d94a
		...1		nmtwsf	TWS Syncron. ended d94a
		.... 1...		nmtwsyn	TWS Syncron. waiting d94a
		.... .1..		nmrfrsym	TWS being stopped for rfr
		.... ..1.		nmmSpecial	TWS Special Event
		1		nmtwsq	TWS quit ended
15	(F)	CHARACTER	1	nmmTWSstatus	Enabler Status queue

normal mode manager control block addresses follow

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
16	(10)	ADDRESS	4	nmmmca	mca address
20	(14)	ADDRESS	4	nmmcioc	checkpoint io comm area
24	(18)	ADDRESS	4	nmmjtdcb	dcb address, jt log file
28	(1C)	ADDRESS	4	nmmjtbuf	buffer addr curr jt file
32	(20)	ADDRESS	4	nmmopc	status record address
36	(24)	ADDRESS	4	nmmhdr	cp hdr record address
40	(28)	ADDRESS	4	nmmemp	ev mgr parameter block
44	(2C)	ADDRESS	4	nmmcpbuf	buffer addr curr cp file
48	(30)	ADDRESS	4	nmmmcjta	addr of mcp jt event
52	(34)	ADDRESS	4	nmmmcmta	addr of mt0 for mcp jt
56	(38)	ADDRESS	4	nmmemdqp	ev manager suspend queue
60	(3C)	ADDRESS	4	nmmjtabp	JTlog file table addr
64	(40)	ADDRESS	4	nmmstep	nmm subtask entry addr
68	(44)	ADDRESS	4	NmmXCM	XCM cache address ADGA
72	(48)	ADDRESS	4	nmmDCXCM	CM XCM cache addresss
76	(4C)	ADDRESS	4	nmmqrsp	RS main block address

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
80	(50)	ADDRESS	4	nmmrswp	RS record work area
84	(54)	ADDRESS	4	nmmjtbp	ptr to JTB CRFACWWC
88	(58)	ADDRESS	4	nmmxdbuf	buffer addr curr xd file
92	(5C)	SIGNED	4	nmmxdsiz	xd record file size
96	(60)	ADDRESS	4	nmmErrStp	error statement 61CDEWC

22A general purpose flags

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
100	(64)	BITSTRING	4	nmmflag6	
		1...		nmjvopen	JV logical file open
		.1..		nmclean	NMWRK cleanup
		..1.		nmmSyncRcvAsked	EME2ERcvr Sync asked BRDa
		...1		nmmSyncRcvReceived	EME2ERcvr Sync event BRDa received by EM BRDa
		.... 1...		nmmTNodeMsg	N127N128 issued cghaDAWC
		.... .1..		NMMqdlog	QDLOG flag
		.... ..1.		NMMbulk	bulkdisc event
		1		NMMSCPop	SCP open bit
101	(65)	1...		NMMSCPtoClose	going to close SCP
		.1..		nmmbatch2	CRVA
		..1.		nmxdopen	current XD is open
		...1		nmxdpendlt	ON some xd02 pend ltp
		.... 1...		nmxdpendcp	ON some xd02 pend ltp
		.... .1..		nmCKPTinit	ON started CKPT init
101	(65)	BITSTRING	2	*	free

various ecbs for NMM and its subtasks follow

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
104	(68)	BITSTRING	4	nmmarecb	auto recovery ecb
108	(6C)	BITSTRING	4	nmmwaecb	workstation analyzer ecb
112	(70)	BITSTRING	4	nmmwecb	nmm work required ecb
116	(74)	BITSTRING	4	nmmtmecb	nmm timer ecb
120	(78)	BITSTRING	4	nmmjtlog	JTlog archiving required
124	(7C)	BITSTRING	4	nmmjtfin	JTlog archiving finished
128	(80)	BITSTRING	4	nmmtwsecb	TWS syncronization ecb
132	(84)	BITSTRING	4	nmmtwsquit	TWS immediate stop

normal mode manager queue and chain headers follow

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
136	(88)	ADDRESS	4	nmmareas	addr of first alloc area
140	(8C)	ADDRESS	4	nmmdoa1	address of first doa
144	(90)	ADDRESS	4	nmmwqa1	address of first wqa

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
148	(94)	ADDRESS	4	nmmtjt1	addr of doa 1st time job
152	(98)	ADDRESS	4	nmmque1	addr of 1st req que entry
156	(9C)	ADDRESS	4	nmmdra1	address of first dra
160	(A0)	ADDRESS	4	nmmdoafr	first free doa
164	(A4)	ADDRESS	4	nmmwqafr	first free wqa
168	(A8)	ADDRESS	4	nmmquefr	first free res que entry
172	(AC)	ADDRESS	4	nmmdrafr	first free dra
176	(B0)	ADDRESS	4	nmmivlfr	first free ivl
180	(B4)	ADDRESS	4	nmmdsefr	first free DSEINFO CCUA
184	(B8)	BITSTRING	4	nmmSyncEMReceiver	TWS immediate stop BRDa
188	(BC)	BITSTRING	4	nmmBulkStartedEcb	Sincro with monitor task
192	(C0)	ADDRESS	4	nmmTNOfr	first free tie node DAWA
196	(C4)	BITSTRING	4	* (9)	free 34aDAWC

miscellaneous saved nmm values

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
232	(E8)	SIGNED	4	nmmjtnot	track addr last JT note
236	(EC)	SIGNED	4	nmmjtbsz	size of current JT buff
240	(F0)	SIGNED	4	nmmbsize	current plan record size
244	(F4)	CHARACTER	4	nmmbkpdt	date, 1st event after bkp
248	(F8)	CHARACTER	4	nmmbkptm	time, 1st event after bkp
252	(FC)	SIGNED	4	nmmjsopt	max jsfile size in bytes '-1' = auto copy active OBRA
256	(100)	SIGNED	4	nmmjscas	js file cylinder size
260	(104)	SIGNED	4	nmmjscnt	num inserts on jsfile
264	(108)	SIGNED	4	nmmtqlim	backup trk records limit '-1' = auto bkp active
268	(10C)	SIGNED	4	nmmtqtot	trk records since backup
272	(110)	CHARACTER	8	nmmtoken_full	operation full token
272	(110)	CHARACTER	4	nmmtoken_pre	prefix
276	(114)	UNSIGNED	4	nmmtoken	opctoken
280	(118)	SIGNED	4	nmmjbufz	reserved
284	(11C)	SIGNED	4	nmmenqcnt	numb of enq attempts d94a
288	(120)	SIGNED	2	nmmopcsz	size of status rec buffer
290	(122)	SIGNED	2	nmmhdrsz	size of cp hdr rec buffer
292	(124)	SIGNED	2	nmmdlfq	number of dlf enqs
294	(126)	SIGNED	2	nmmnsubs	number of nmm subtasks
296	(128)	CHARACTER	8	nmmerrnm	noerror parmlib memname
304	(130)	CHARACTER	64	nmmexit	timerexit
368	(170)	ADDRESS	4	nmmhtcp	ptr to HTC block
372	(174)	SIGNED	4	nmmhtcl	length of HTC block
376	(178)	ADDRESS	4	nmmlstTIEfp	ptr to late tie chain DAWA
380	(17C)	ADDRESS	4	nmmlstTIElp	ptr to last late tie DAWA
384	(180)	ADDRESS	4	nmmTNOflp	ptr to first leaf TNode DAWA
388	(184)	SIGNED	2	nmmlastipo	last ins tie relative posDAWA
390	(186)	SIGNED	2	nmmTNodeno	nbr of TNodes DAWA
392	(188)	ADDRESS	4	nmmhtsp	ptr to HTS block
396	(18C)	SIGNED	4	nmmhtsl	length of HTS block
400	(190)	CHARACTER	4	nmmWQAname	WSA name for cmd
404	(194)	CHARACTER	8	nmmWQAdest	WSA dest for cmd

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
412	(19C)	CHARACTER	4	*	free

R4A normal mode manager cell pools

R4A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
416	(1A0)	SIGNED	4	nmmtiepid	Time Queue cell pool id
420	(1A4)	ADDRESS	4	nmmtieptr	adress of first tie
424	(1A8)	ADDRESS	4	nmmDSRfr	first free DSRinfo
428	(1AC)	CHARACTER	16	NMMtokTCB	TCB token of the NMM task
444	(1BC)	CHARACTER	8	NMMDDmem	PROTxx command D90A
452	(1C4)	CHARACTER	8	NMMDSmem	PROTxx command D90A
460	(1CC)	CHARACTER	1	nmmAsymst	Symphony asynchro phase
461	(1CD)	CHARACTER	8	NMMSKIPmem	SKIPINCL member name BLBA
469	(1D5)	BITSTRING	1	nmmZCMDflag	noerror processing flags
		1...		*	- DSPJTBL not moved ... used
		.1..		*	- DSPJIDX not moved ... used
		..1.		nmmDWQA	- DWQA requested
		...1		nmmSWQA	- SWQA requested
		.... 1...		nmmDSPTNOD	- DSPTNOD requested
		.... .1..		nmmDSPLTIE	- DSPLTIE requested
		.... ..1.		*	-
		1		*	-
470	(1D6)	CHARACTER	2	*	
472	(1D8)	ADDRESS	4	nmmTNOfp	first Tie Node ptr DAWA

$CHWA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
476	(1DC)	ADDRESS	4	nmmdsafr	first free DSAINFO CHWA
480	(1E0)	ADDRESS	4	nmmTNOllp	ptr to last leaf TNode DAWC
484	(1E4)	ADDRESS	4	nmmCToken	C Interface Token CQCa
488	(1E8)	CHARACTER	16	*	free CQCc
504	(1F8)	CHARACTER		nmmend	end of nmm block

Cross reference

Name	Hex Offset	Hex Value	Level
nmadlock	9	10	4
nmadopen	6	08	4
nmbackup	6	02	4
nmchlock	7	08	4
nmCKPTinit	65	04	3
nmclean	64	40	3

Name	Hex Offset	Hex Value	Level
nmcpeq	7	40	4
nmcplock	7	10	4
nmcpopen	6	80	4
nmcxopen	9	80	4
nmjscopy	6	04	4
nmjsforce	7	02	4
nmjsopen	6	40	4
nmjtarce	B	10	3
nmjtarcf	B	20	3
nmjtauxt	C	40	3
nmjtevany	B	04	3
nmjtevcp	B	08	3
nmjtfirst	B	80	3
nmjtioer	B	01	3
nmjtlast	B	40	3
nmjtlock	B	02	3
nmjtturn	C	80	3
nmjvopen	64	80	3
nmltlock	9	20	4
nmltopen	6	20	4
nmm	0		1
nmmaract	7	04	4
nmmARC	7	01	4
nmmareas	88		2
nmmarecb	68		2
nmmAsymst	1CC		2
nmmbatch2	65	40	3
nmmbkpdt	F4		2
nmmbkptm	F8		2
nmmbsize	F0		2
NMMbulk	64	02	3
nmmBulkStartedEcb	BC		2
nmmcioc	14		2
nmmcmd	8	10	4
nmmcpbuf	2C		2
nmmCToken	1E4		2
nmmDCXCM	48		2
NMMDDmem	1BC		2
nmmdesc	0		2
nmmdlfq	124		2
nmmdoafr	A0		2

Name	Hex Offset	Hex Value	Level
nmmdoa1	8C		2
nmmdrafr	AC		2
nmmdra1	9C		2
nmmdsafr	1DC		2
nmmdsefr	B4		2
NMMDSmem	1C4		2
nmmDSPLTIE	1D5	04	3
nmmDSPTNOD	1D5	08	3
nmmDSRfr	1A8		2
nmmDWQA	1D5	20	3
nmmedp	8	01	4
nmmemdqp	38		2
nmmemp	28		2
nmmend	1F8		2
nmmenqcnt	11C		2
nmmerrnm	128		2
nmmErrStp	60		2
nmmexit	130		2
nmmflags	6		2
nmmflag1	6		3
nmmflag2	7		3
nmmflag3	8		3
nmmflag4	9		3
nmmflag5	E		2
nmmflag6	64		2
nmmhdr	24		2
nmmhdrsz	122		2
nmmhtcl	174		2
nmmhtcp	170		2
nmmhtsl	18C		2
nmmhtsp	188		2
nmmivlfr	B0		2
nmmjbufz	118		2
nmmjscas	100		2
nmmjscnt	104		2
nmmjsopt	FC		2
nmmjtabp	3C		2
nmmjtbp	54		2
nmmjtbsz	EC		2
nmmjtbuf	1C		2
nmmjtdcb	18		2

Name	Hex Offset	Hex Value	Level
nmmjte	8	02	4
nmmjtfin	7C		2
nmmjtflg	B		2
nmmjtlog	78		2
nmmjtnot	E8		2
nmmlastipo	184		2
nmmlim	8	80	4
nmmlstTIEfp	178		2
nmmlstTIElp	17C		2
nmmmca	10		2
nmmmcjta	30		2
nmmmcmta	34		2
nmmMCP	C	10	3
nmmnoerr	D		2
nmmnsubs	126		2
nmmopc	20		2
nmmopcsz	120		2
nmmPermBP	C	04	3
NMMqdlog	64	04	3
nmmqrsp	4C		2
nmmquefr	A8		2
nmmque1	98		2
nmmrst	8	08	4
nmmrswp	50		2
NMMSCPop	64	01	3
NMMSCPtoClose	65	80	3
NMMSKIPmem	1CD		2
nmmSpecial	E	02	3
nmmstat	A		2
nmmstep	40		2
nmmstop	8	04	4
nmmSWQA	1D5	10	3
nmmsymst	9	02	4
nmmSyncEMReceiver	B8		2
nmmSyncRcvAsked	64	20	3
nmmSyncRcvReceived	64	10	3
nmmtiepid	1A0		2
nmmtieptr	1A4		2
nmmtjt1	94		2
nmmtmecb	74		2
nmmTNodeMsg	64	08	3

Name	Hex Offset	Hex Value	Level
nmmTNodeno	186		2
nmmTNOflp	180		2
nmmTNOfp	1D8		2
nmmTNOfr	C0		2
nmmTNOllp	1E0		2
nmmtoken	114		3
nmmtoken_full	110		2
nmmtoken_pre	110		3
NMMtokTCB	1AC		2
nmmto1	8	40	4
nmmto2	8	20	4
nmmtqlim	108		2
nmmtqtot	10C		2
nmmtwsecb	80		2
nmmtwsquit	84		2
nmmTWSstatus	F		2
nmmUrgDQE	C	08	3
nmmver	4		2
nmmwaecb	6C		2
nmmwecb	70		2
nmmWQAdest	194		2
nmmwqafr	A4		2
nmmWQAname	190		2
nmmwqa1	90		2
NmmXCM	44		2
nmmxdbuf	58		2
nmmxdsiz	5C		2
nmmZCMDflag	1D5		2
nmncp	7	20	4
nmnolist	D	20	3
nmnomemb	D	40	3
nmnoproc	D	80	3
nmnotry	D	10	3
nmocxwsa	9	01	4
nmProtDD	E	80	3
nmProtDS	E	40	3
nmrdlock	9	04	4
nmrdopen	9	40	4
nmrfresh	7	80	4
nmrfrsym	E	04	3
nmsiopen	6	10	4

Name	Hex Offset	Hex Value	Level
nmskipin	D	01	3
nmsymback	C	20	3
nmturnsw	6	01	4
nmturn2sw	E	20	3
nmtwsf	E	10	3
nmtwsq	E	01	3
nmtwsyn	E	08	3
nmwslock	9	08	4
nmxdopen	65	20	3
nmxdpendcp	65	08	3
nmxdpendlt	65	10	3

NNN - Node name record

Name : DCLNNN

Function:
This segment declares the scheduler node name checkpoint record. This record is always the second record on the first track of the checkpoint data set.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	8024	NNN	NODE NAME RECORD
0	(0)	CHARACTER	4	NNNDESC	BLOCK DESCRIPTOR = 'NNN '
4	(4)	CHARACTER	2	NNNVER	BLOCK MAPPING VERSION
6	(6)	BITSTRING	2	*	RESERVED
8	(8)	SIGNED	4	NNNNODES	NUMBER OF NODES IN RECORD
12	(C)	SIGNED	4	* (3)	RESERVED (INIT TO ZERO)
24	(18)	CHARACTER	8	NNNNAMES (1000)	KNOWN NODE NAMES
8024	(1F58)	CHARACTER		NNNEND	END OF NNN

OPC - OPCSTAT record

Name : DCLOPC

Function:
This is the OPCSTAT record mapping. This record is first in the checkpoint data set.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	104	OPCSTAT	STATUS RECORD
0	(0)	CHARACTER	10	OPCKEY	KEY ALWAYS='0000000000'
10	(A)	CHARACTER	6	*	RESERVED
16	(10)	SIGNED	2	OPC#REPL	NUMBER OF REPLANS SINCE LAST PLAN NEXT
18	(12)	CHARACTER	10	OPCDPEN	DP PLAN PERIOD END
18	(12)	CHARACTER	6	OPCDPEND	DP PLAN PERIOD END DATE

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
24	(18)	CHARACTER	4	OPCDPENT	DP PLAN PERIOD END TIME
28	(1C)	CHARACTER	8	OPCCPDDN	CURRENT PLAN DDN
36	(24)	CHARACTER	8	OPCOCPDD	OLD CURRENT PLAN DDN
44	(2C)	CHARACTER	2	OPCJTCUR	CURRENT JTLOG NUMBER
46	(2E)	CHARACTER	2	OPCJTARC	LAST ARCHIVED JTLOG
48	(30)	CHARACTER	2	OPCJTMAX	MAX JTLOG NUMBER
50	(32)	CHARACTER	1	OPCJTOPE	ARCHIVE JTLOG OPEN
51	(33)	CHARACTER	1	OPCSYMRF	SYMPHONY AFTER REFRESH
52	(34)	CHARACTER	8	*	RESERVED
60	(3C)	CHARACTER	1	OPCCP	CURRENT PLAN EXISTS, Y|N
61	(3D)	CHARACTER	1	OPCJTDL	BACKUP JT CREATED, Y|N
62	(3E)	CHARACTER	1	OPCTUNCP	TURNOVER FLAG
63	(3F)	CHARACTER	8	OPCCURJS	DDN FOR JSFILE IN USE
71	(47)	CHARACTER	1	OPCBACKINGUP	BACKUP STARTED Y|N
72	(48)	CHARACTER	10	OPCLEVEL	CKPT BLOCK CODE LEVEL
72	(48)	CHARACTER	2	OPCVER	BLOCK MAPPING VERSION
74	(4A)	CHARACTER	8	OPCFMID	TRACKER FMID
82	(52)	CHARACTER	1	OPCCKPTR	CKPT REFRESH Y|N
84	(54)	SIGNED	4	OPCSYMRUNNUM	SYMPHONY CURRENT RUN NUMBER
88	(58)	CHARACTER	1	OPCTUNSYM	SYMPHONY TURNOVER FLAG
89	(59)	CHARACTER	1	OPCTWSYN	Y=TIVOLI WORKLOAD SCHEDULER MACRO FAILED
90	(5A)	CHARACTER	1	OPCTUNASYM	SYMPHONY TURNOVER ASYNCHR
91	(5B)	CHARACTER	1	OPCTWSJBNM	TWSJOBNAME IN JTOPTS 'O' = OCCNAME 'E' = EXTNAME 'X' = EXTNOCC 'J' = JOBNAME
92	(5C)	SIGNED	2	OPCSUPOL	SUPPRESS POLICY FROM OPT
94	(5E)	BITSTRING	2	OPCFLAGS	BIT BATTERY
		1...	2	OPCNOPTIME	TOPOLOGY NOPTIMEDEP
94	(5E)	BITSTRING	1	*	FREE
96	(60)	CHARACTER	8	*	FREE

Cross reference

Name	Hex Offset	Hex Value	Level
OPC#REPL	10		2
OPCBACKINGUP	47		2
OPCCKPTR	52		2
OPCCP	3C		2
OPCCPDDN	1C		2
OPCCURJS	3F		2
OPCDPEN	12		2
OPCDPEND	12		3
OPCDPENT	18		3
OPCFLAGS	5E		2
OPCFMID	4A		3

Name	Hex Offset	Hex Value	Level
OPCJTARC	2E		2
OPCJTCUR	2C		2
OPCJTDL	3D		2
OPCJTMAX	30		2
OPCJTOPE	32		2
OPCKEY	0		2
OPCLEVEL	48		2
OPCNOPTIME	5E	80	3
OPCOCPDD	24		2
OPCSTAT	0		1
OPCSUPOL	5C		2
OPCSYMRF	33		2
OPCSYMRUNNUM	54		2
OPCTUNASYM	5A		2
OPCTUNCP	3E		2
OPCTUNSYM	58		2
OPCTWSJBNM	5B		2
OPCTWSYN	59		2
OPCVER	48		3

OPERR - Ops in error report record in DP

Name : DCLOPERR

Function:
Describes layout of oper in error records representing report data on occurrences which have one/more operations in error, when the last back-up of current plan was created. The records have the standard recsize and keysize of DPIN records.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	266	OPEAREA	OPE RECORD
0	(0)	CHARACTER	114	OPEKEY	KEY
0	(0)	SIGNED	2	OPETYPE	TYPE IS ALWAYS 6
2	(2)	CHARACTER	2	*	ADJUST TO ADID OFFS
4	(4)	CHARACTER	34	*	ADJUST KEY SIZE
38	(26)	CHARACTER	26	OPEOCC	OCCURANCE IDENTIFICATION
38	(26)	CHARACTER	16	OPEADID	APPLICATION IDENTIFICATION
54	(36)	CHARACTER	10	OPEOPIA	PLANNED OCC INPUT ARRIVAL
54	(36)	CHARACTER	6	OPEOPIAD	DAY (YYMMDD)
54	(36)	CHARACTER	4	OPEOPIAY	
58	(3A)	CHARACTER	2	OPEOPIAE	(DD)
60	(3C)	CHARACTER	4	OPEOPIAT	TIME (HHMM)
60	(3C)	CHARACTER	2	OPEOPIAH	TIME (HH)
62	(3E)	CHARACTER	2	OPEOPIAM	TIME (MM)
64	(40)	CHARACTER	7	OPEWSID	WSID OF OPERATION

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
64	(40)	CHARACTER	3	OPEWSIDN	OPERATION NUMBER
67	(43)	CHARACTER	4	OPEWSIDS	WORKSTATION NAME
71	(47)	CHARACTER	43	*	TO FILL UP THE KEY
114	(72)	CHARACTER	152	OPEDATA	DATA PART OF OPE RECORD
114	(72)	CHARACTER	24	OPEOTXT	APPLICATION TEXT
138	(8A)	CHARACTER	1	OPEOPRI	PRIORITY
139	(8B)	CHARACTER	10	OPEOAIA	ACTUAL OCC INPUT ARRIVAL
139	(8B)	CHARACTER	6	OPEOAIAD	DAY (YYMMDD)
139	(8B)	CHARACTER	4	OPEOAIAY	(YYMM)
143	(8F)	CHARACTER	2	OPEOAIAE	(DD)
145	(91)	CHARACTER	4	OPEOAIAT	TIME (HHMM)
145	(91)	CHARACTER	2	OPEOAIAH	TIME (HH)
147	(93)	CHARACTER	2	OPEOAIAM	TIME (MM)
149	(95)	CHARACTER	10	OPEOPC	PLANNED OCC. COMPLETION
149	(95)	CHARACTER	6	OPEOPCDT	DAY (YYMMDD)
149	(95)	CHARACTER	4	OPEOPCY	(YYMM)
153	(99)	CHARACTER	2	OPEOPCD	(DD)
155	(9B)	CHARACTER	4	OPEOPCT	TIME (HHMM)
155	(9B)	CHARACTER	2	OPEOPCH	TIME (HH)
157	(9D)	CHARACTER	2	OPEOPCM	TIME (MM)
159	(9F)	CHARACTER	24	OPEOPTXT	OPERATION TEXT
183	(B7)	CHARACTER	8	OPEJBNM	JOBNAME
191	(BF)	CHARACTER	10	OPESPIA	SPECIFIC OPER INPUT ARR.
191	(BF)	CHARACTER	6	OPESPIAD	DAY (YYMMDD) | BLANK
191	(BF)	CHARACTER	4	OPESPIAY	(YYMM)
195	(C3)	CHARACTER	2	OPESPIAE	(DD)
197	(C5)	CHARACTER	4	OPESPIAT	TIME (HHMM) | BLANK
197	(C5)	CHARACTER	2	OPESPIAH	TIME (HH)
199	(C7)	CHARACTER	2	OPESPIAM	TIME (MM)
201	(C9)	CHARACTER	10	OPEPAIA	ACTUAL OPER INPUT ARRIVAL
201	(C9)	CHARACTER	6	OPEPAIAD	DAY (YYMMDD)
201	(C9)	CHARACTER	4	OPEPAIAY	(YYMM)
205	(CD)	CHARACTER	2	OPEPAIAE	(DD)
207	(CF)	CHARACTER	4	OPEPAIAT	TIME (HHMM)
207	(CF)	CHARACTER	2	OPEPAIAH	TIME (HH)
209	(D1)	CHARACTER	2	OPEPAIAM	TIME (MM)
211	(D3)	CHARACTER	10	OPEPPC	LATEST OUT
211	(D3)	CHARACTER	6	OPEPPCDT	DAY (YYMMDD)
211	(D3)	CHARACTER	4	OPEPPCY	(YYMM)
215	(D7)	CHARACTER	2	OPEPPCD	(DD)
217	(D9)	CHARACTER	4	OPEPPCT	TIME (HHMM)
217	(D9)	CHARACTER	2	OPEPPCH	TIME (HH)
219	(DB)	CHARACTER	2	OPEPPCM	TIME (MM)
221	(DD)	CHARACTER	10	OPEPSC	SPECIFIC OPER COMPLETION
221	(DD)	CHARACTER	6	OPEPSCT	DAY (YYMMDD) |BLANK
221	(DD)	CHARACTER	4	OPEPSCY	(YYMM)
225	(E1)	CHARACTER	2	OPEPSCD	(DD)
227	(E3)	CHARACTER	4	OPEPACT	TIME (HHMM) |BLANK
227	(E3)	CHARACTER	2	OPEPSCH	TIME (HH)
229	(E5)	CHARACTER	2	OPEPSCM	TIME (MM)
231	(E7)	CHARACTER	4	OPEODUR	OPERATION DURATION
231	(E7)	CHARACTER	2	OPEODURH	TIME (HH)
233	(E9)	CHARACTER	2	OPEODURM	(MM)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
235	(EB)	CHARACTER	10	OPEERR	TIME FOR OPER ERROR
235	(EB)	CHARACTER	6	OPEERRDT	DAY (YYMMDD)
235	(EB)	CHARACTER	4	OPEERRY	(YYMM)
239	(EF)	CHARACTER	2	OPEERRD	(DD)
241	(F1)	CHARACTER	4	OPEERRT	TIME (HHMM)
241	(F1)	CHARACTER	2	OPEERRTH	TIME (HH)
243	(F3)	CHARACTER	2	OPEERRTM	TIME (MM)
245	(F5)	CHARACTER	4	OPEOPERR	OPERATION ERROR CODE
249	(F9)	CHARACTER	16	OPEOPUDATA	OPERATION USERDATA
265	(109)	CHARACTER	1	OPECMSTAT	CAT MGT STATUS

Cross reference

Name	Hex Offset	Hex Value	Level
OPEADID	26		4
OPEAREA	0		1
OPECMSTAT	109		3
OPEDATA	72		2
OPEERR	EB		3
OPEERRD	EF		5
OPEERRDT	EB		4
OPEERRT	F1		4
OPEERRTH	F1		5
OPEERRTM	F3		5
OPEERRY	EB		5
OPEJBNM	B7		3
OPEKEY	0		2
OPEOAIA	8B		3
OPEOAIAD	8B		4
OPEOAIAE	8F		5
OPEOAIAH	91		5
OPEOAIAM	93		5
OPEOAIAT	91		4
OPEOAIAY	8B		5
OPEOCC	26		3
OPEODUR	E7		3
OPEODURH	E7		4
OPEODURM	E9		4
OPEOPC	95		3
OPEOPCD	99		5
OPEOPCDT	95		4
OPEOPCH	9B		5
OPEOPCM	9D		5

Name	Hex Offset	Hex Value	Level
OPEOPCT	9B		4
OPEOPCY	95		5
OPEOPERR	F5		3
OPEOPIA	36		4
OPEOPIAD	36		5
OPEOPIAE	3A		6
OPEOPIAH	3C		6
OPEOPIAM	3E		6
OPEOPIAT	3C		5
OPEOPIAY	36		6
OPEOPRI	8A		3
OPEOPTXT	9F		3
OPEOPUDATA	F9		3
OPEOTXT	72		3
OPEPACT	E3		4
OPEPAIA	C9		3
OPEPAIAD	C9		4
OPEPAIAE	CD		5
OPEPAIAH	CF		5
OPEPAIAM	D1		5
OPEPAIAT	CF		4
OPEPAIAY	C9		5
OPEPPC	D3		3
OPEPPCD	D7		5
OPEPPCDT	D3		4
OPEPPCH	D9		5
OPEPPCM	DB		5
OPEPPCT	D9		4
OPEPPCY	D3		5
OPEPSC	DD		3
OPEPSCD	E1		5
OPEPSCH	E3		5
OPEPSCM	E5		5
OPEPSCT	DD		4
OPEPSCY	DD		5
OPESPIA	BF		3
OPESPIAD	BF		4
OPESPIAE	C3		5
OPESPIAH	C5		5
OPESPIAM	C7		5
OPESPIAT	C5		4

Name	Hex Offset	Hex Value	Level
OPESPIAY	BF		5
OPETYPE	0		3
OPEWSID	40		3
OPEWSIDN	40		4
OPEWSIDS	43		4

OPI - Operator instruction

Name : DCLOPI

Function:
This segment declares an operator instruction record, physically located in EQQOIDS.
If the valid_from key is blank and the valid_to key is set to the highest value (all hexadecimal ’F’), the operator instruction is called 'permanent', else it is called 'temporary'.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	*	OPI	
0	(0)	CHARACTER	78	OPIBODY	
0	(0)	CHARACTER	28	OPIKEY	
0	(0)	CHARACTER	18	OPIADOP	
0	(0)	CHARACTER	16	OPIADID	APPLICATION NAME
16	(10)	SIGNED	2	OPIOPNO	OPERATION NUMBER
18	(12)	CHARACTER	10	OPIVALTO	VALID TO DATE+TIME
18	(12)	CHARACTER	6	OPIVALT1	DATE
24	(18)	CHARACTER	4	OPIVALT2	TIME
28	(1C)	CHARACTER	4	OPIEYE	EYE CATCHER
32	(20)	UNSIGNED	1	OPIVERS	VERSION NUMBER
33	(21)	CHARACTER	10	OPIVALFR	VALID FROM DATE+TIME
33	(21)	CHARACTER	6	OPIVALF1	DATE
39	(27)	CHARACTER	4	OPIVALF2	TIME
43	(2B)	CHARACTER	4	OPIWSID	NOT USED
47	(2F)	CHARACTER	8	OPIOWNER	LAST UPDATE USER
55	(37)	CHARACTER	10	OPIUPDAT	LAST UPDATE, DATE+TIME
55	(37)	CHARACTER	6	OPIUPDT1	DATE
61	(3D)	CHARACTER	4	OPIUPDT2	TIME
65	(41)	CHARACTER	1	OIFLAGS	FLAGS
		1...		OPINAC	DO NO CHECK AGAINST AD
		.1..		OPIBEING	UNDER CREATION
66	(42)	CHARACTER	2	*	FREE
68	(44)	CHARACTER	8	OPILUTS	LAST UPDATE TIMESTAMP
76	(4C)	SIGNED	2	OPILINES	NUMBER OF LINES COMING
78	(4E)	CHARACTER	72	OPITAB (*)	1 LINE WITH OPERATOR INSTR.

Cross reference

Name	Hex Offset	Hex Value	Level
OIFLAGS	41		3
OPI	0		1

Name	Hex Offset	Hex Value	Level
OPIADID	0		5
OPIADOP	0		4
OPIBEING	41	40	4
OPIBODY	0		2
OPIEYE	1C		3
OPIKEY	0		3
OPILINES	4C		3
OPILUTS	44		3
OPINAC	41	80	4
OPIOPNO	10		5
OPIOWNER	2F		3
OPITAB	4E		2
OPIUPDAT	37		3
OPIUPDT1	37		4
OPIUPDT2	3D		4
OPIVALFR	21		3
OPIVALF1	21		4
OPIVALF2	27		4
OPIVALTO	12		4
OPIVALT1	12		5
OPIVALT2	18		5
OPIVERS	20		3
OPIWSID	2B		3

OPT - Subsystem options

Name : DCLOPT

Function:
This segment declares options and parameter values supplied to the subsystem through init statements.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	688	opt	opcesa options block
0	(0)	CHARACTER	4	optdesc	block descriptor = 'OPT '
4	(4)	CHARACTER	2	optver	block mapping version
6	(6)	BITSTRING	4	optflags	yesno type options
		1...		opthost	1 this is an opc host
		.1..		optstby	1 this is a standby system
		..1.		optnec	1 start a nec task
		...1		optewtr	1 start an event wtr
		.... 1...		optjcc	1 start a jcc task
		.... .1..		optrcvy	1 start a recovery task
		.... ..1.		optjsub	1 opc will submit jobs

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		1		optnprt	1 don't compl prt opers
7	(7)	1...		optevms	1 create event statistics
		.1..		optksrx	1 keep excl resrc
		..1.		optksrs	1 keep shrd resrc
		...1		optosub	1 track only opcsubmit jobs
		.... 1...		optjopt	1 track opcsubmit others
		.... .1..		optsbno	1 no jobcard check
		.... ..1.		optsbok	1 submit only ok jobs
		1		optgsms	1 measure gs requests
8	(8)	1...		optcpnew	1 currplan(new) specified
		.1..		optcpms	1 create cp enq statistics
		..1.		optappc	1 start an appc task APPA
		...1		optdual	1 dual JT logging LOGA
		.... 1...		optres	1 restartblank opr val
		.... .1..		optially	1 operiall(y)
		.... ..1.		optdally	1 operdall(y)
		1		optsim	1 simulate mode
9	(9)	1...		optsysf	1 takeover if host sys fail
		.1..		opthostf	1 takeover if host opc fail
		..1.		optacct	1 account is reqd in jobcd
		...1		optpgmr	1 prog nm is reqd in jobcd
		.... 1...		optdb2	1 db2 is used for model db
		.... .1..		optrer	1 rerouteblank opr val
		.... ..1.		optssxb	1 SSX build requested
		1		optssxbm	1 SSX merge requested
10	(A)	BITSTRING	2	optflag2	yesno type options
		1...		optsscmp	1 load SSCM permanent
		.1..		optrodm	1 start a rodm task
		..1.		optdynev	1 dynadd res at event
		...1		optdynsu	1 dynadd res ar submit
		.... 1...		optapdst	1 APPC dests exist
		.... .1..		optPlex	1 PLEX configuration
		.... ..1.		optserver	1 a Server
		1		optwsms	1 wsa stats onoff
11	(B)	1...		optproc	1 proc substitution ACJA
		.1..		optvfail	1 varsub fail ignore ACJA
		..1.		optfltask	1 RCLEANUP = YES D90C start FL and PSU D90A
		...1		optSPIN	1 SPIN(YES) 0 SPIN(NO)
		.... 1...		optarm	SYSPLEX ARM request D44A
		.... .1..		opttplgy	topology is present d94a
		.... ..1.		optextmon	Ext mon active(YN)
		1		optngens	1&colon.don't do gen. search
12	(C)	CHARACTER	1	optOnCompl	ONCOMPLETE - Y set to YES - N set to NO - R Reset to blank - do nothing
13	(D)	CHARACTER	1	optDynOnCo	DYNONCOMPLETE - Y set to YES - N set to NO - R Reset to blank - do nothing
14	(E)	BITSTRING	1	optflag4	additional flags
		11..		optaudCP	AUDITCP statement
		1...		optcond	1 log COND status change
		.1..		optcdep	1 log CDEP status change
		..1.		optettnew	1 ETT dep from ETTYTCR1 CWRA
		...1		optCNDSub	1 Cond S on submission DAOA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.... 1...		opttaskusr	DDFA 41A CWRA
		.... .1..		opttrackW	DGPA
		.... ..1.		opttrka3j	DFSA
		1		optOutCol	62CDDFC41ACWRCDAOCDGPCDF SC
15	(F)	BITSTRING	1	optflag6	
		1...		optCSABIG	
		.1..		optzCHRC	on HIGHRC active forzCen DJLA
		..11 1111		*	reserved free63A
16	(10)	UNSIGNED	2	optX01sz	EXIT01 jcl size ACJA
18	(12)	BITSTRING	2	optxitfl	user exit flag
20	(14)	SIGNED	4	optback	nmm backup frequency '-1' = auto bkp active OBRA
24	(18)	SIGNED	4	optnrdr	number of event readers
28	(1C)	SIGNED	4	optabnum	num no error retcodes
32	(20)	SIGNED	4	optarnum	num reset error retcodes
36	(24)	SIGNED	4	opthrcnr	highest ok retcode
40	(28)	SIGNED	4	optmaxjs	max js-file size '-1' = auto copy active OBRA
44	(2C)	SIGNED	4	optstme	simulate end time
48	(30)	SIGNED	4	optstms	simulate start time
52	(34)	SIGNED	4	optoff_dly	WS offline actions delay
56	(38)	ADDRESS	4	optJdestP	joblog dest parms
60	(3C)	SIGNED	4	optrscont	contentiontime
64	(40)	ADDRESS	4	optawstb	APPC WS table
68	(44)	SIGNED	4	opttcpport	local tcpip port
72	(48)	CHARACTER	8	opttcpid	local tcpip id
80	(50)	ADDRESS	4	optHistp	addr of history block
84	(54)	ADDRESS	4	optabptr	address of no error table
88	(58)	ADDRESS	4	optarptr	address of reset err tbl
92	(5C)	ADDRESS	4	optdtptr	addr of destination tab
96	(60)	ADDRESS	4	optsidp	addr of system id tab CATA
100	(64)	ADDRESS	4	optsrvsptr	to server names
104	(68)	CHARACTER	8	optncfnm	ncf application name
112	(70)	CHARACTER	8	optmirrd	remote standby destination
120	(78)	CHARACTER	8	optjccnm	jcc init member name
128	(80)	CHARACTER	8	optwtrnm	ewtr init member name
136	(88)	CHARACTER	8	optrdrnm (16)	erdr init member names
264	(108)	CHARACTER	8	optxitnm (16)	user exit load module names note 1st entry is for exit0
392	(188)	CHARACTER	8	optarnm	ar init member name
400	(190)	CHARACTER	8	optsscmn	sscm load module name
408	(198)	CHARACTER	8	optgroup	xcf group name
416	(1A0)	CHARACTER	16	optmem	xcf member name
432	(1B0)	CHARACTER	16	optgtab	name of global jcl vartab
448	(1C0)	CHARACTER	16	optcalnm	name of default calendar
464	(1D0)	BITSTRING	1	optaudit (16)	audit specifications
		1...		opttrack	audit this resource
		.1..		optread	audit read accesses
		..1.		optdata	log entire vsam record
480	(1E0)	SIGNED	2	optqlen	max queue length
482	(1E2)	SIGNED	2	optadsmf	ad smoothing factor
484	(1E4)	SIGNED	2	optadlim	ad limit for feedback
486	(1E6)	SIGNED	2	optopitm	number days op inst is new
488	(1E8)	SIGNED	2	optgstask	number of gs tasks

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
490	(1EA)	SIGNED	2	optplan	default dp period start
492	(1EC)	CHARACTER	1	optvsub	var substitution YNS
493	(1ED)	CHARACTER	1	optonode	outputnode F=final A=any
494	(1EE)	SIGNED	2	optsupol	suppresspolicy value
496	(1F0)	SIGNED	2	optshpol	shutdownpolicy value
498	(1F2)	CHARACTER	1	optsuppl	suppressaction char
499	(1F3)	CHARACTER	1	optsfail	subfailaction char
500	(1F4)	SIGNED	2	optjtlog	number of JTlogs LOGA
502	(1F6)	SIGNED	2	optoverc	PS overcomit
504	(1F8)	SIGNED	2	optpulse	Pulse rate
506	(1FA)	SIGNED	2	opttimeout	TCP timeout interval
508	(1FC)	CHARACTER	3	optwsfail	WS failure options
508	(1FC)	CHARACTER	1	opwsfail_sact	restart option
509	(1FD)	CHARACTER	1	opwsfail_ract	reroute option
510	(1FE)	CHARACTER	1	opwsfail_aact	available option
511	(1FF)	CHARACTER	3	optwsoffl	WS offline options
511	(1FF)	CHARACTER	1	opwsoffl_sact	restart option
512	(200)	CHARACTER	1	opwsoffl_ract	reroute option
513	(201)	CHARACTER	1	opwsoffl_aact	available option
514	(202)	BITSTRING	2	optwo	wto opts
		1...		optwote	no task end wto
		.1..		optwoer	ended-in-error status
		..1.		optwole	late operation
		...1		optwodu	too long duration
		.... 1...		optwoqx	queue limit exceeded
		.... .1..		optworc	resource contention
516	(204)	BITSTRING	2	optal	netview generic alert
		1...		optalte	no task end alert
		.1..		optaler	ended-in-error status
		..1.		optalle	late operation
		...1		optaldu	too long duration
		.... 1...		optalqx	queue limit exceeded
518	(206)	BITSTRING	2	optml	mlog write opts
		1...		optmlte	no task end mlog write
		.1..		optmler	ended-in-error status
		..1.		optmlle	late operation
		...1		optmldu	too long duration
		.... 1...		optmlqx	queue limit exceeded
		.... .1..		optmlrc	resource contention
520	(208)	BITSTRING	1	optflags5	Cat Mgt flags deleted D90C
		1...		optOpSumWS	1 Opsum data from WS DAGA
		.1..		optBufPerm	1 Buffer Pool permanent DCOA
		..11 1111	*	free DAGCDCOC	
521	(209)	CHARACTER	1	optbestm	est match A=ALL,F=RF,O=RO
522	(20A)	UNSIGNED	1	optflag3	BOGCLUCABIC
		1...		optgdgns	on GDG_NONST CPRC
		.1..		optftwjsub	1 opc will submit jobs
		..1.		optMhldSf	manual hold at submit fai
		...1		optMonTbsm	1 TBSM external monitor
		.... 1...		optMonTep	1 TEP external monitor
		.... .1..		optCritJobs	1 activate dyn crit path CRRA
		.... ..1.		optRecCPCompl	1 rec. critpaths on comp CRRA
		1		optnoetry	NOERROR Trial cmd msg CTXA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
523	(20B)	CHARACTER	1	optTWSjbnm	TWSJOBNAME val. in JTOPTS BOGA O OCCNAME BOGA E EXTNAME BOGA X EXTNOCC BOGA J JOBNAME BOGA
524	(20C)	SIGNED	4	optgmtof	GMT offset
528	(210)	CHARACTER	8	optrodm_parm	rodm parm member name
536	(218)	CHARACTER	1	optUX001FA	EQQUX001 fail action BRMC
537	(219)	CHARACTER	1	optopins	OPINFOSCOPE I=In-pro ops A=All ops
538	(21A)	SIGNED	2	optRSpol	resource lookahead
540	(21C)	CHARACTER	8	optcodep	codepage
548	(224)	SIGNED	4	optmaxoccn	maximum occ number
552	(228)	SIGNED	4	optstatim	statmsg time
556	(22C)	SIGNED	2	optevelim	evelim number
558	(22E)	CHARACTER	1	optsvfailc (2)	array for savarfail D54CCHWC
560	(230)	CHARACTER	1	optvfailc (3)	array of varfail values (&, ,) ACJA
563	(233)	1...		optxitxres	extended stat X option
		.1..		optsuppENF	SUPPRESSENF CCUA
		..1.		optsamsgMLOG	automation msg logging CHWA
		...1		optsamsgSLOG	automation msg logging CHWA
		.... 1...		optsvfail	1 varsub fail ignore CHWA
		.... .1..		optremjcldir	remove jcl directives
		.... ..1.		optnoecheck	NOERROR consistency test
		1		optnoemsg	NOERROR consistency msgs
564	(234)	ADDRESS	4	optFLparm	pointer to FLOPTS D54A
568	(238)	CHARACTER	10	optWLM	D52a
		1...		optWLMRq	WLM Request D52a
		.111		optWLMMode	WLM Mode D52a
		 1111		optWLMPolicy	WLM Policy in Mode1 D52a
569	(239)	CHARACTER	8	optWLMHPClass	Hi.Perf. WLM1 Class D52a
577	(241)	UNSIGNED	1	optWLMThreshold	WLM Smart Threshold D52a
578	(242)	CHARACTER	1	optsechk	A=all,N=no,O=operonly CCUA
579	(243)	UNSIGNED	1	optCPBPLim	CP Buffer Pool limit CJBC
580	(244)	ADDRESS	4	optRCL	pointer to RCLOPTS D90A
584	(248)	CHARACTER	8	opttplgsrv	topology param member d94a
592	(250)	ADDRESS	4	optjplexp	address of JESPLEX CCUA system name list CCUA (mapped by DCLJPLEX) CCUA
596	(254)	SIGNED	2	optsysplID	SYSPLEXID CCUA (default is 0) CCUA
598	(256)	SIGNED	2	optdlsmo	deadline smoothing factor
600	(258)	SIGNED	2	optdlfdk	deadline limit feedback
602	(25A)	CHARACTER	1	*	free 34ACLEC
603	(25B)	UNSIGNED	1	optCPDTLim	CP dataset limit CLEA
604	(25C)	ADDRESS	4	optMONO	pointer to MONOPTS
608	(260)	BITSTRING	2	optMONP	MONPOL options
		1...		optmper	error job flag
		.1..		optmpcrt	critical job flag
		..1.		optmpcrp	critical path job flag
		...1		optmplat	late job flag
		.... 1...		optmpdur	long duration job flag
		.... .1..		optmpman	manual flag
610	(262)	BITSTRING	2	optmonal	TEP monalert
		1...		optmoer	ended-in-error status
		.1..		optmoeop	tasks that ended in error
		..1.		optmola	late operation
		...1		optmodu	too long duration

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.... 1...		optmoqx	queue limit exceeded
		.... .1..		optmowlm	WLM flag
		.... ..1.		optmospe	special resource time out
		1		optmomsg	messages monitered
611	(263)	1...		optmonop	monitored jobs
		.1..		optdefcpg	default codepage set CQ2A
612	(264)	CHARACTER	8	optmoncode	codepage for Monitor task
620	(26C)	ADDRESS	4	opttcpptr	addr to TCPIP area CMDA
624	(270)	BITSTRING	1	NOERRFLG	reserved CTXA
		1...		optnomemb	reserved CTXA
		.111 1111		*	reserved CTXA
625	(271)	BITSTRING	1	optflag1	
		1...		optCDEPstp	AUDITCP statement
		.1..		optUnexpRC	AUDITCP statement
		..1.		optITOM	on ITOM(YES) jcl tailoring wished
		...1 1111		*	
626	(272)	SIGNED	2	optaleac	CUWA
628	(274)	ADDRESS	4	optParWSTableptr	parallel workstations
632	(278)	ADDRESS	4	opthttpptr	addr to HTTPOPTS area
636	(27C)	SIGNED	4	optmhistr	max DB2 HISTory rows DAIA
640	(280)	SIGNED	2	optalemindur	DCCA
642	(282)	CHARACTER	8	optOCstc	DCCA62a
650	(28A)	CHARACTER	2	*	DJLA
652	(28C)	SIGNED	4	optzchrcdef	highrc default for zCen DJLA
656	(290)	CHARACTER	8	optoslcnm	oslc init member name
664	(298)	ADDRESS	4	optoslcp	pointer to OSLC opts
668	(29C)	CHARACTER	20	*	reserved
688	(2B0)	CHARACTER		optend	end of opt

Cross reference

Name	Hex Offset	Hex Value	Level
NOERRFLG	270		2
opt	0		1
optabnum	1C		2
optabptr	54		2
optacct	9	20	3
optadlim	1E4		2
optadsmf	1E2		2
optal	204		2
optaldu	204	10	3
optaleac	272		2
optalemindur	280		2
optaler	204	40	3
optalle	204	20	3
optalqx	204	08	3
optalte	204	80	3

Name	Hex Offset	Hex Value	Level
optapdst	A	08	3
optappc	8	20	3
optarm	B	08	3
optarnm	188		2
optarnum	20		2
optarptr	58		2
optaudCP	E	C0	3
optaudit	1D0		2
optawstb	40		2
optback	14		2
optbestm	209		2
optBufPerm	208	40	3
optcalnm	1C0		2
optcdep	E	40	4
optCDEPstp	271	80	3
optCNDSub	E	10	3
optcodep	21C		2
optcond	E	80	4
optCPBPLim	243		2
optCPDTLim	25B		2
optcpms	8	40	3
optcpnew	8	80	3
optCritJobs	20A	04	3
optCSABIG	F	80	3
optdally	8	02	3
optdata	1D0	20	3
optdb2	9	08	3
optdefcpg	263	40	3
optdesc	0		2
optdlfdk	258		2
optdlsmo	256		2
optdtptr	5C		2
optdual	8	10	3
optdynev	A	20	3
optDynOnCo	D		2
optdynsu	A	10	3
optend	2B0		2
optettnew	E	20	3
optevelim	22C		2
optevms	7	80	3
optewtr	6	10	3

Name	Hex Offset	Hex Value	Level
optextmon	B	02	3
optflags	6		2
optflags5	208		2
optflag1	271		2
optflag2	A		2
optflag3	20A		2
optflag4	E		2
optflag6	F		2
optFLparm	234		2
optfltask	B	20	3
optftwjsub	20A	40	3
optgdgns	20A	80	3
optgmtof	20C		2
optgroup	198		2
optgsms	7	01	3
optgstask	1E8		2
optgtab	1B0		2
optHistp	50		2
opthost	6	80	3
opthostf	9	40	3
opthrcnr	24		2
opthttpptr	278		2
optially	8	04	3
optITOM	271	20	3
optjcc	6	08	3
optjccnm	78		2
optJdestP	38		2
optjopt	7	08	3
optjplexp	250		2
optjsub	6	02	3
optjtlog	1F4		2
optksrs	7	20	3
optksrx	7	40	3
optmaxjs	28		2
optmaxoccn	224		2
optmem	1A0		2
optmhistr	27C		2
optMhldSf	20A	20	3
optmirrd	70		2
optml	206		2
optmldu	206	10	3

Name	Hex Offset	Hex Value	Level
optmler	206	40	3
optmlle	206	20	3
optmlqx	206	08	3
optmlrc	206	04	3
optmlte	206	80	3
optmodu	262	10	3
optmoeop	262	40	3
optmoer	262	80	3
optmola	262	20	3
optmomsg	262	01	3
optmonal	262		2
optmoncode	264		2
optMONO	25C		2
optmonop	263	80	3
optMONP	260		2
optMonTbsm	20A	10	3
optMonTep	20A	08	3
optmoqx	262	08	3
optmospe	262	02	3
optmowlm	262	04	3
optmpcrp	260	20	3
optmpcrt	260	40	3
optmpdur	260	08	3
optmper	260	80	3
optmplat	260	10	3
optmpman	260	04	3
optncfnm	68		2
optnec	6	20	3
optngens	B	01	3
optnoecheck	233	02	2
optnoemsg	233	01	2
optnoetry	20A	01	3
optnomemb	270	80	3
optnprt	6	01	3
optnrdr	18		2
optOCstc	282		2
optoff_dly	34		2
optOnCompl	C		2
optonode	1ED		2
optopins	219		2
optopitm	1E6		2

Name	Hex Offset	Hex Value	Level
optOpSumWS	208	80	3
optoslcnm	290		2
optoslcp	298		2
optosub	7	10	3
optOutCol	E	01	3
optoverc	1F6		2
optParWSTableptr	274		2
optpgmr	9	10	3
optplan	1EA		2
optPlex	A	04	3
optproc	B	80	3
optpulse	1F8		2
optqlen	1E0		2
optRCL	244		2
optrcvy	6	04	3
optrdrnm	88		2
optread	1D0	40	3
optRecCPCompl	20A	02	3
optremjcldir	233	04	2
optrer	9	04	3
optres	8	08	3
optrodm	A	40	3
optrodm_parm	210		2
optrscont	3C		2
optRSpol	21A		2
optsamsgMLOG	233	20	2
optsamsgSLOG	233	10	2
optsbno	7	04	3
optsbok	7	02	3
optsechk	242		2
optserver	A	02	3
optsfail	1F3		2
optshpol	1F0		2
optsidp	60		2
optsim	8	01	3
optSPIN	B	10	3
optsrvsptr	64		2
optsscmn	190		2
optsscmp	A	80	3
optssxb	9	02	3
optssxbm	9	01	3

Name	Hex Offset	Hex Value	Level
optstatim	228		2
optstby	6	40	3
optstme	2C		2
optstms	30		2
optsupol	1EE		2
optsuppENF	233	40	2
optsuppl	1F2		2
optsvfail	233	08	2
optsvfailc	22E		2
optsysf	9	80	3
optsysplID	254		2
opttaskusr	E	08	3
opttcpid	48		2
opttcpport	44		2
opttcpptr	26C		2
opttimeout	1FA		2
opttplgsrv	248		2
opttplgy	B	04	3
opttrack	1D0	80	3
opttrackW	E	04	3
opttrka3j	E	02	3
optTWSjbnm	20B		2
optUnexpRC	271	40	3
optUX001FA	218		2
optver	4		2
optvfail	B	40	3
optvfailc	230		2
optvsub	1EC		2
optWLM	238		2
optWLMHPClass	239		3
optWLMMode	238	70	3
optWLMPolicy	238	0F	3
optWLMRq	238	80	3
optWLMThreshold	241		3
optwo	202		2
optwodu	202	10	3
optwoer	202	40	3
optwole	202	20	3
optwoqx	202	08	3
optworc	202	04	3
optwote	202	80	3

Name	Hex Offset	Hex Value	Level
optwsfail	1FC		2
optwsms	A	01	3
optwsoffl	1FF		2
optwtrnm	80		2
optxitfl	12		2
optxitnm	108		2
optxitxres	233	80	2
optX01sz	10		2
optzCHRC	F	40	3
optzchrcdef	28C		2
opwsfail_aact	1FE		3
opwsfail_ract	1FD		3
opwsfail_sact	1FC		3
opwsoffl_aact	201		3
opwsoffl_ract	200		3
opwsoffl_sact	1FF		3

OSLC - Integration information

Name : DCLOSLC

Function:
This segment declares options and parameter values used for the integration with OSLC software.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	128	oslcopts	OSLC integration information
0	(0)	CHARACTER	4	oslceye	eye catcher 'OSLC'
4	(4)	CHARACTER	2	oslccver	version
6	(6)	CHARACTER	1	oslcpolicy	policy A | M | C
7	(7)	CHARACTER	1	oslcssl	use SSL for the ticket Y | N
8	(8)	CHARACTER	1	oslcusrf	use user fields for template
9	(9)	CHARACTER	1	*	reserved
10	(A)	SIGNED	2	oslcprio	priority
12	(C)	ADDRESS	4	oslcurip	ptr to URI
16	(10)	SIGNED	4	oslcuril	URI length
20	(14)	ADDRESS	4	oslcusrp	ptr to user
24	(18)	SIGNED	4	oslcusrl	user length
28	(1C)	ADDRESS	4	oslcpwdp	ptr to password
32	(20)	SIGNED	4	oslcpwdl	pwd length
36	(24)	CHARACTER	56	oslchost	hostname from URI
92	(5C)	SIGNED	4	oslcport	port from URI
96	(60)	ADDRESS	4	oslcdescp	ptr to user description
100	(64)	SIGNED	4	oslcdescl	user description length
104	(68)	ADDRESS	4	oslcturip	ptr to DWC URI
108	(6C)	SIGNED	4	oslcturil	DWC URI length
112	(70)	CHARACTER	16	*	reserved

Cross reference

Name	Hex Offset	Hex Value	Level
oslccver	4		2
oslcdescl	64		2
oslcdescp	60		2
oslceye	0		2
oslchost	24		2
oslcopts	0		1
oslcpolicy	6		2
oslcport	5C		2
oslcprio	A		2
oslcpwdl	20		2
oslcpwdp	1C		2
oslcssl	7		2
oslcturil	6C		2
oslcturip	68		2
oslcuril	10		2
oslcurip	C		2
oslcusrf	8		2
oslcusrl	18		2
oslcusrp	14		2

PER - Period layout

Name : DCLPER

Function:
This segment declares a period definition, physically located in the workstation description database (EQQWSDS)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	PERREC	PERIOD LAYOUT
0	(0)	CHARACTER	94	PERFIX	FIXED PART OF RECORD
0	(0)	CHARACTER	2	*	RESERVED BY IO-ROUTINES
2	(2)	CHARACTER	8	PERKEY	
2	(2)	CHARACTER	8	PERIOD	PERIOD ID ========KEY
10	(A)	UNSIGNED	1	PERVERS	VERSION OF RECORD NUM=NEW
11	(B)	1...		PERFROMRG	on from an RG
		.1..		PERMULTI	on RG multi instance
		..11 1111		*	RESERVED
12	(C)	SIGNED	2	PERINTVL	INTERVAL OF CYCLIC ORIGINS
14	(E)	CHARACTER	1	PERTYPE	CYCLICNONCYCLIC TYPE
15	(F)	CHARACTER	30	PERDESC	DESCRIPTION OF PERIOD
45	(2D)	CHARACTER	8	PERUSER	ID OF USER LAST CHANGE
53	(35)	CHARACTER	6	PERDATE	DATE FOR LAST CHANGE
59	(3B)	CHARACTER	4	PERTIME	TIME FOR LAST CHANGE, HHMM

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
63	(3F)	CHARACTER	1	*	free
64	(40)	ADDRESS	4	PERRGTAB	RG table
68	(44)	CHARACTER	8	perluts	last update timestamp
76	(4C)	CHARACTER	16	perjvt	jcl variable table id
92	(5C)	SIGNED	2	PERORGS	ORIGIN DATES IN PERIOD
94	(5E)	CHARACTER	12	PERVAR (*)	VARIABLE PART OF REC
94	(5E)	CHARACTER	6	PERORIG	ORIGIN(=START)DATE INT
100	(64)	CHARACTER	6	PERIVLEND	END DATE OF INTERVAL

Cross reference

Name	Hex Offset	Hex Value	Level
PERDATE	35		3
PERDESC	F		3
PERFIX	0		2
PERFROMRG	B	80	3
PERINTVL	C		3
PERIOD	2		4
PERIVLEND	64		3
perjvt	4C		3
PERKEY	2		3
perluts	44		3
PERMULTI	B	40	3
PERORGS	5C		3
PERORIG	5E		3
PERREC	0		1
PERRGTAB	40		3
PERTIME	3B		3
PERTYPE	E		3
PERUSER	2D		3
PERVAR	5E		2
PERVERS	A		3

RCLOP - Restart and clean up options record

Name : DCLRCLOP

Function:
This segment declares the layout of the restart and clean up options record.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	116	RCLOP	RCLOPTS
0	(0)	CHARACTER	4	RCO_EYE	EYE CATCHER

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
4	(4)	SIGNED	4	RCO_SP	SUBPOOL USE BY STORAGE MACRO
8	(8)	CHARACTER	8	RCO_PROTDSMBR	DSNPROT: MEMBER NAME CONTAINING THE PROTECTED DSN LIST
16	(10)	ADDRESS	4	RCO_PROTDSNP	POINTER TO PROTECTED DSN LIST
20	(14)	CHARACTER	8	RCO_PROTDDMBR	DDPROT: MEMBER NAME CONTAINING THE PROTECTED DD LIST
28	(1C)	ADDRESS	4	RCO_PROTDDNP	POINTER TO PROTECTED DD LIST
32	(20)	ADDRESS	4	RCO_NORESTP	DDNOREST: POINTER TO THE LIST OF DDNAMES THAT MAKE THE STEP NON-RESTARTABLE
36	(24)	ADDRESS	4	RCO_NEVEXP	DDNEVER: POINTER TO THE LIST OF DDNAMES THAT MAKE THE STEP NEVER EXECUTABLE
40	(28)	ADDRESS	4	RCO_ALWAYSP	DDALWAYS: POINTER TO THE LIST OF DDNAMES THAT MAKE THE STEP ALWAYS RE-EXECUTABLE
44	(2C)	CHARACTER	8	RCO_DSDEST	DSTDEST: DESTINATION
52	(34)	CHARACTER	5	RCO_CLJOBNA	CLNJOBNM: PREFIX TO BE USED TO GENERATE THE NAME OF THE CLEAN UP JOB
57	(39)	CHARACTER	1	RCO_USERSYS	USERSYS: HANDLE USER SYSOUT FLAG
58	(3A)	CHARACTER	1	RCO_RMMACTIVE	RMM: Y|N RMM ACTIVE
59	(3B)	CHARACTER	1	RCO_STEPRESChk	STEPRESCHK YES (default) NO
60	(3C)	ADDRESS	4	RCO_SAVPROTDD	POINTER TO DD PROTLIST (MAJOR SHOULD FREE)
64	(40)	ADDRESS	4	RCO_SAVPROTDS	POINTER TO DS PROTLIST (MAJOR SHOULD FREE)
68	(44)	ADDRESS	4	RCO_JCCCLTAB	POINTER TO JES CLASS TAB
72	(48)	SIGNED	4	*	RESERVED
76	(4C)	CHARACTER	40	RCO_JOBINFO	WHAT FOLLOWS JOB IN JOBCARD

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	100	RCO_EXTRA	EXTRA AREA
0	(0)	CHARACTER	44	RCO_STEPLIB	EQQCLEAN STEPLIB
44	(2C)	CHARACTER	8	RCO_SKIPMEM	SKIPINCLULDE MEMBER NAME
52	(34)	ADDRESS	4	RCO_SKIPMEMP	POINTER TO SKIP INCLUDE LIST
56	(38)	CHARACTER	1	RCO_IMMLOGIC	B = BEST, A = AR STEP
57	(39)	CHARACTER	1	RCO_GDGSIMAUTO	Y/N
58	(3A)	CHARACTER	8	RCO_DYMMYLAST	DUMMY LAST STEP
66	(42)	CHARACTER	1	RCO_GDGPROT	GDG PROTECTION LOCK
67	(43)	CHARACTER	1	RCO_JLOG	E = ON ERROR, D = ON DEMAND (DEFAULT)
68	(44)	CHARACTER	1	RCO_JOPI	E = ON ERROR, D = ON DEMAND (DEFAULT)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
69	(45)	CHARACTER	31	*	FREE

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	RCO_SKIPEL	SKIPINCL NAME LIST
0	(0)	CHARACTER	10	RCO_SKIPFIX	FIXED PART
0	(0)	CHARACTER	4	RCO_SKIPEYE	EYE CATCHER = RSKI
4	(4)	SIGNED	4	RCO_SKIPTSZ	TOTAL SIZE OF AREA
8	(8)	SIGNED	2	RCO_SKIPCTR	COUNT OF ARRAY ELEMENTS
10	(A)	CHARACTER	8	RCO_SKIPVAR(*)	VARIABLE PART
10	(A)	CHARACTER	8	RCO_SKIPNAME	PROTECTED DSNAME

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	RCO_PROTDSN	PROTECTED DSN LIST
0	(0)	CHARACTER	10	RCO_PRDSFIX	FIXED PART
0	(0)	CHARACTER	4	RCO_PRDSEYE	EYE CATCHER
4	(4)	SIGNED	4	RCO_PRDSTSZ	TOTAL SIZE OF AREA
8	(8)	SIGNED	2	RCO_PRDSCTR	COUNT OF ARRAY ELEMENTS
10	(A)	CHARACTER	44	RCO_PRDSVAR(*)	VARIABLE PART
10	(A)	CHARACTER	44	RCO_PRDSNAME	PROTECTED DSNAME

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	RCO_PROTDDN	PROTECTED DDN LIST
0	(0)	CHARACTER	10	RCO_PRDDFIX	FIXED PART
0	(0)	CHARACTER	4	RCO_PRDDEYE	EYE CATCHER
4	(4)	SIGNED	4	RCO_PRDDTSZ	TOTAL SIZE OF AREA
8	(8)	SIGNED	2	RCO_PRDDCTR	COUNT OF ARRAY ELEMENTS
10	(A)	CHARACTER	8	RCO_PRDDVAR(*)	VARIABLE PART
10	(A)	CHARACTER	8	RCO_PRDDNAME	PROTECTED DDNAME

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	RCO_NOREST	POINTER TO THE LIST OF DDNAMES THAT MAKE THE STEP NON-RESTARTABLE
0	(0)	CHARACTER	10	RCO_NORSFIX	FIXED PART
0	(0)	CHARACTER	4	RCO_NORSEYE	EYE CATCHER
4	(4)	SIGNED	4	RCO_NORSTSZ	TOTAL SIZE OF AREA
8	(8)	SIGNED	2	RCO_NORSCTR	COUNT OF ARRAY ELEMENTS
10	(A)	CHARACTER	8	RCO_NORSVAR(*)	VARIABLE PART
10	(A)	CHARACTER	8	RCO_NORSDDN	DDNAME

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	RCO_NEVEX	POINTER TO THE LIST OF DDNAMES THAT MAKE THE STEP NEVER EXECUTABLE
0	(0)	CHARACTER	10	RCO_NEVEXFIX	FIXED PART
0	(0)	CHARACTER	4	RCO_NEVEXEYE	EYE CATCHER

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
4	(4)	SIGNED	4	RCO_NEVEXTSZ	TOTAL SIZE OF AREA
8	(8)	SIGNED	2	RCO_NEVEXCTR	COUNT OF ARRAY ELEMENTS
10	(A)	CHARACTER	8	RCO_NEVEXVAR(*)	VARIABLE PART
10	(A)	CHARACTER	8	RCO_NEVEXDDN	DDNAME

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	RCO_ALWAYS	POINTER TO THE LIST OF DDNAMES THAT MAKE THE STEP ALWAYS RE-EXECUTABLE
0	(0)	CHARACTER	10	RCO_ALWYFIX	FIXED PART
0	(0)	CHARACTER	4	RCO_ALWYEYE	EYE CATCHER
4	(4)	SIGNED	4	RCO_ALWYTSZ	TOTAL SIZE OF AREA
8	(8)	SIGNED	2	RCO_ALWYCTR	COUNT OF ARRAY ELEMENTS
10	(A)	CHARACTER	8	RCO_ALWYVAR(*)	VARIABLE PART
10	(A)	CHARACTER	8	RCO_ALWYDDN	DDNAME

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	RCO_JESDST	POINTER TO THE LIST OF DESTINATIONS AND JES CL
0	(0)	CHARACTER	16	RCO_CLTABLE	COMMON PART
0	(0)	SIGNED	4	RCO_LISTCNT	COUNTER OF TABLE ENTRIES
4	(4)	SIGNED	4	RCO_TABLEN	LENGTH OF THE TABLE
8	(8)	CHARACTER	8	*	RESERVED
16	(10)	CHARACTER	16	RCO_CLASTAB(*)	CLASS VAR TABLE
16	(10)	CHARACTER	8	RCO_TRKDEST	TRACKER DESTINATION
24	(18)	CHARACTER	1	RCO_JESCLAS	JES CLASS FOR OUTPUT
25	(19)	CHARACTER	7	*	RESERVED

Cross reference

Name	Hex Offset	Hex Value	Level
RCLOP	0		1
RCO_ALWAYS	0		1
RCO_ALWAYSP	28		2
RCO_ALWYCTR	8		3
RCO_ALWYDDN	A		3
RCO_ALWYEYE	0		3
RCO_ALWYFIX	0		2
RCO_ALWYTSZ	4		3
RCO_ALWYVAR	A		2
RCO_CLASTAB	10		2
RCO_CLJOBNA	34		2
RCO_CLTABLE	0		2
RCO_DSDEST	2C		2
RCO_DUMMYLAST	3A		2

Name	Hex Offset	Hex Value	Level
RCO_EXTRA	0		2
RCO_EYE	0		2
RCO_GDGPROT	42		2
RCO_GDGSIMAUTO	39		2
RCO_IMMLOGIC	38		2
RCO_JCCCLTAB	44		2
RCO_JESCLASS	18		3
RCO_JESDST	0		1
RCO_JLOG	43		2
RCO_JOBINFO	4C		2
RCO_JOPI	44		2
RCO_LISTCNT	0		3
RCO_NEVEX	0		1
RCO_NEVEXCTR	8		3
RCO_NEVEXDDN	A		3
RCO_NEVEXEYE	0		3
RCO_NEVEXFIX	0		2
RCO_NEVEXP	24		2
RCO_NEVEXTSZ	4		3
RCO_NEVEXVAR	A		2
RCO_NOREST	0		1
RCO_NORESTP	20		2
RCO_NORSCTR	8		3
RCO_NORSDDN	A		3
RCO_NORSEYE	0		3
RCO_NORSFIX	0		2
RCO_NORSTSZ	4		3
RCO_NORSVAR	A		2
RCO_PRDDCTR	8		3
RCO_PRDDEYE	0		3
RCO_PRDDFIX	0		2
RCO_PRDDNAME	A		3
RCO_PRDDTSZ	4		3
RCO_PRDDVAR	A		2
RCO_PRDSCTR	8		3
RCO_PRDSEYE	0		3
RCO_PRDSFIX	0		2
RCO_PRDSNAME	A		3
RCO_PRDSTSZ	4		3
RCO_PRDSVAR	A		2
RCO_PROTDDMBR	14		2

Name	Hex Offset	Hex Value	Level
RCO_PROTDDN	0		1
RCO_PROTDDNP	1C		2
RCO_PROTDSMBR	8		2
RCO_PROTDSN	0		1
RCO_PROTDSNP	10		2
RCO_RMMACTIVE	3A		2
RCO_SAVPROTDD	3C		2
RCO_SAVPROTDS	40		2
RCO_SP	4		2
RCO_STEPRESCHK	3B		2
RCO_TABLEN	4		3
RCO_TRKDEST	10		3
RCO_USERSYS	39		2

REL - Job release record

Name : DCLREL

Function:
This segment declares the layout of a release record in a submit release data set

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	REL	RELEASE RECORD
0	(0)	CHARACTER	8	RELJOBNM	JOB NAME
8	(8)	CHARACTER	8	RELJOBID	JOB ID
16	(10)	CHARACTER	16	RELADID	APPLICATION NAME
32	(20)	CHARACTER	10	RELOPIA	OPERATION INPUT ARRIVAL
42	(2A)	SIGNED	2	RELOPNUM	OPERATION NUMBER
44	(2C)	CHARACTER	4	RELWSID	WORK STATION NAME
48	(30)	SIGNED	4	RELRETRY	NUMBER OF RELEASE RETRIES
52	(34)	CHARACTER	28	RELSPARE	SPARE

Cross reference

Name	Hex Offset	Hex Value	Level
REL	0		1
RELADID	10		2
RELJOBID	8		2
RELJOBNM	0		2
RELOPIA	20		2
RELOPNUM	2A		2
RELRETRY	30		2
RELSPARE	34		2

Name	Hex Offset	Hex Value	Level
RELWSID	2C		2

RNGRP - rungroup

Name: DCLRNGRP

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	rungroup	

Fixed part of common data

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	180	rungrcom	
0	(0)	CHARACTER	2	rungrtype	used by IO routines '08'
2	(2)	CHARACTER	8	rungrkey	key of the record
2	(2)	CHARACTER	8	rungrid	run cycle group id
10	(A)	UNSIGNED	1	rungrver	version number
11	(B)	BITSTRING	1	rungrflags	flags
		1...		rungrcaldef	work bit, off in DB
		.111 1111		*	reserved
12	(C)	CHARACTER	4	rungriat	default ia time (opt)
16	(10)	CHARACTER	16	rungrjvtab	default group jv table (opt)
32	(20)	CHARACTER	16	rungrcal	default group calendar (opt)
48	(30)	CHARACTER	50	rungrdesc	run cycle group description
98	(62)	CHARACTER	8	rungrluser	lastupdate user-id
106	(6A)	CHARACTER	6	rungrldate	last update date
112	(70)	CHARACTER	4	rungrltime	last update time
116	(74)	SIGNED	4	rungrrunc	total number of runcycles
120	(78)	SIGNED	4	rungrfrun	offset to first runcycle
124	(7C)	CHARACTER	8	rungrluts	lastupdate timestamp
132	(84)	CHARACTER	16	rungrowner	owner
148	(94)	CHARACTER	6	rungrdead	default deadline
148	(94)	CHARACTER	2	rungrday	deadline day rel. to start
150	(96)	CHARACTER	4	rungrtime	deadline time
154	(9A)	CHARACTER	26	*	reserved
180	(B4)	CHARACTER	*	rungrvar	variable length section

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	156	runcycle	

run cycle definition

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	SIGNED	4	runnrun	offset to next runcycle
4	(4)	SIGNED	2	runcycle	run cycle number
6	(6)	CHARACTER	8	runname	period name, rule name

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
14	(E)	CHARACTER	4	*	reserved
18	(12)	UNSIGNED	1	runrule	free day rule
19	(13)	CHARACTER	1	runtype	run cycle type
20	(14)	CHARACTER	6	runfrom	run cycle valid from
26	(1A)	CHARACTER	6	runout	run cycle not valid from
32	(20)	CHARACTER	50	rundesc	run cycle description
82	(52)	CHARACTER	4	runiatime	input arrival time
86	(56)	CHARACTER	16	runjvtab	jcl var table name
102	(66)	SIGNED	2	runrulelen	rule definition length
104	(68)	CHARACTER	4	runeveryrr	EVERY repeat range
108	(6C)	CHARACTER	4	runeveryet	EVERY end time
112	(70)	CHARACTER	8	runsetid	run cycle set identifier
120	(78)	CHARACTER	16	runcalendar	run cycle secondary calendar
136	(88)	CHARACTER	6	rundead	deadline
136	(88)	CHARACTER	2	rundday	deadline day rel. to start
138	(8A)	CHARACTER	4	rundtime	deadline time
142	(8E)	CHARACTER	14	*	free
156	(9C)	CHARACTER		runend	

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	*	runrdefseg	rule definition segment
0	(0)	CHARACTER	*	runrdef	rule definition value

Cross reference

Name	Hex Offset	Hex Value	Level
runcalendar	78		2
runcycle	0		1
runcycle	4		2
rundday	88		3
rundead	88		2
rundesc	20		2
rundtime	8A		3
runend	9C		2
runeveryet	6C		2
runeveryrr	68		2
runfrom	14		2
rungrcal	20		3
rungrcaldef	B	80	4
rungrcom	0		2
rungrday	94		4
rungrdead	94		3
rungrdesc	30		3
rungrflags	B		3
rungrfrun	78		3

Name	Hex Offset	Hex Value	Level
rungriat	C		3
rungrid	2		4
rungrjvtab	10		3
rungrkey	2		3
rungrldate	6A		3
rungrltime	70		3
rungrluser	62		3
rungrluts	7C		3
rungroup	0		1
rungrowner	84		3
rungrrunc	74		3
rungrtime	96		4
rungrtype	0		3
rungrvar	B4		2
rungrver	A		3
runiatime	52		2
runjvtab	56		2
runname	6		2
runnrun	0		2
runout	1A		2
runrdef	0		2
runrdefseg	0		1
runrule	12		2
runrulelen	66		2
runsetid	70		2
runtype	13		2

RSR - Special resource record

Name : DCLRSR

Function:
Resource Class Data Description Declare. This segment describes the main key of a resource data instance, as well as defining the actual data objects contained for this key. This mapping applies to the RD data set records, the CX data sets, and for resource data in storage.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	44	RSU	
0	(0)	ADDRESS	4	RSUNEXT	ADDRESS TO NEXT ENTRY | 0
4	(4)	CHARACTER	4	RSUEYE	ID
8	(8)	CHARACTER	2	RSUVER	VERSION NUMBER
10	(A)	CHARACTER	2	*	RESERVED
12	(C)	CHARACTER	4	RSUOPER	CP OPERATION ID

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
12	(C)	UNSIGNED	3	RSUOCC	OCCURRENCE ID
15	(F)	UNSIGNED	1	RSUOP	OPERATION ID
16	(10)	CHARACTER	1	RSUALCTYPE	ALLOCATE TYPE (X/S)
17	(11)	CHARACTER	11	*	RESERVED
28	(1C)	CHARACTER	4	RSUWS	WORK STATION
32	(20)	SIGNED	4	RSUALCNUM	AMOUNT ALLOCATED
36	(24)	CHARACTER	8	RSUJOBN	JOBNAME
44	(2C)	CHARACTER		RSUEND	END

OPERATIONS WAITING FOR RESOURCE ALLOCATION ADDRESSED VIA QRSPLNWAIT (SEE DCLQRS)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	44	RSW	
0	(0)	ADDRESS	4	RSWNEXT	ADDRESS TO NEXT ENTRY | 0
4	(4)	CHARACTER	4	RSWEYE	ID
8	(8)	CHARACTER	2	RSWVER	VERSION NUMBER
10	(A)	BITSTRING	1	RSWFLAGS	FLAG BYTE
		1...		RSWCONTTIE	CONTENTION MESSAGE TIE BUILD
		.1..		RSWCHANGED	RESOURCE WAS UPDATED
		..11 1111		*	RESERVED
11	(B)	CHARACTER	1	*	RESERVED
12	(C)	CHARACTER	4	RSWOPER	CP OPERATION KEY
12	(C)	UNSIGNED	3	RSWOCC	OCCURRENCE ID
15	(F)	UNSIGNED	1	RSWOP	OPERATION ID
16	(10)	CHARACTER	1	RSWALCTYPE	ALLOCATE TYPE (X/S)
17	(11)	CHARACTER	11	*	RESERVED
28	(1C)	CHARACTER	4	RSWWS	WORK STATION
32	(20)	SIGNED	4	RSWALCNUM	AMOUNT TO ALLOCATE
36	(24)	CHARACTER	8	RSWALCTIME	TIME PUT ON WAIT Q (TOD)
44	(2C)	CHARACTER		RSWEND	END

DIALOG RESOURCE LIST EXTENSION THE EXTENSION IS APPENDED TO THE RS MAIN BODY DURING A GS LIST REQUEST.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	72	RSL	DIALOG LIST EXTENTION
0	(0)	SIGNED	4	RSLXUSE	CURRENT USED EXCLUSIVE
4	(4)	SIGNED	4	RSLSUSE	CURRENT USED SHARED
8	(8)	SIGNED	4	RSLWAITR	RESERVED
12	(C)	SIGNED	4	*	WAITING REASON (SEE DCLQDRSN)
16	(10)	CHARACTER	1	RSLWQF	ELEMENTS ON WAIT QUEUE (Y/N)
17	(11)	CHARACTER	1	RSLXALL	ANY CURRENT ALL X
18	(12)	CHARACTER	1	RSLSALL	S
19	(13)	CHARACTER	5	*	RESERVED
24	(18)	CHARACTER	48	RSLCURIVL	CURRENT INTERVAL
72	(48)	CHARACTER		RSLEND	END

DIALOG IN USE LIST AND WAITING QUEUE MAPPING. USED FOR DIALOG REQUESTS WITH RESOURCE CODE RESESWQ & RESRSUL.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	72	RSD	
0	(0)	CHARACTER	28	RSDKEY	CP KEY
0	(0)	CHARACTER	16	RSDOCC	OCC ID
16	(10)	CHARACTER	6	RSDOCCIAD	OCC INPUT ARRIVAL DATE
22	(16)	CHARACTER	4	RSDOCCIAT	OCC INPUT ARRIVAL TIME
26	(1A)	SIGNED	2	RSDOPNO	OPERATION NUMBER
28	(1C)	CHARACTER	4	RSDWS	WORK STATION
32	(20)	CHARACTER	8	RSDJOBNAME	JOBNAME
40	(28)	CHARACTER	6	RSDOPSTD	OPERATION START/L.OUT DATE
46	(2E)	CHARACTER	4	RSDOPSTT	OPERATION START/L.OUT TIME
50	(32)	SIGNED	4	RSDOCCNO	OCCURRENCE NUMBER
54	(36)	SIGNED	4	RSDALCNUM	AMOUNT ALLOCATED
58	(3A)	SIGNED	2	RSDPRIO	PRIORITY
60	(3C)	SIGNED	2	RSDDUR	DURATION
62	(3E)	CHARACTER	1	RSDALCTYPE	ALLOCATION TYPE
63	(3F)	CHARACTER	1	RSDSTAT	STATUS
64	(40)	SIGNED	4	RSDWRSN	REASON FOR WAIT
68	(44)	CHARACTER	1	RSDIAFLAG	FLAGS
		1...		RSDINUSEL	IN-USE LIST
69	(45)	CHARACTER	3	*	RESERVED
72	(48)	CHARACTER		RSDEND	END

RESOURCE INSTANCE DESCRIPTION. KEY FIELDS ARE DEFINED HERE. WHEN LOCATED IN THE RS-DATASPACE THE RECORD IS ADDRESSABLE VIA QRSPLNIX (SEE DCLQRS).

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	216	RS	RESOURCE INSTANCE STRUCTURE
0	(0)	CHARACTER	64	RSKEY	KEY STRUCTURE
0	(0)	CHARACTER	2	RSID	USED BY FOR IO METHODS
2	(2)	CHARACTER	44	RSNAME	PLAN INSTANCE KEY
46	(2E)	CHARACTER	16	*	RESERVED FOR ADDITIONAL KEY FIELDS

OBJECT HEADER. CONTAINS THE INDEX ID OF THE RESOURCE AS WELL AS COMMON RESOURCE DATA. NOTE: THE OBJECT HEADER IS NOT POSSIBLE TO UPDATE IN DIALOG CODE.

64	(40)	CHARACTER	12	RSOBJDESC	OBJECT HEADER STRUCTURE
64	(40)	SIGNED	4	*	RESERVED
68	(44)	SIGNED	4	RSOBJIDX	RESOURCE INDEX NUMBER
72	(48)	BITSTRING	1	RSOFLAGS	FLAG BYTE
		1...		RSOUPDATE	RECORD UPDATED (IN DSPC ONLY)
		.1..		RSOCONERR	ON ERROR MODIFIED IN CX
		..1.		RSOCUSEDFOR	USED FOR CHANGED IN CX
		...1		RSOCDEFWSC	DEFAULT WSC CHANGED IN CX
		.... 1...		RSODYNADD	DYNADDED RESOURCE
		.... .111		*	RESERVED
73	(49)	CHARACTER	1	RSORODMA	RODM STATUS, AVAILABILITY
74	(4A)	CHARACTER	1	RSORODMQ	QUANTITY
75	(4B)	CHARACTER	1	RSORODMD	DEVIATION

REST OF MAIN RS BODY.

76	(4C)	CHARACTER	4	RSEYE	RECORD ID
80	(50)	SIGNED	4	RSVER	RECORD VERSION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
84	(54)	CHARACTER	8	RSLUSER	LAST UPDATING USER in RD
92	(5C)	CHARACTER	8	RSGROUP	GROUP ID
100	(64)	CHARACTER	6	RSLUDATE	DATE OF LAST UPDATE in RD
106	(6A)	CHARACTER	4	RSLUTIME	TIME OF LAST UPDATE IN RD
110	(6E)	CHARACTER	2	RSONERROR	ON ERROR OPTION: F=FREE FX=FREE IF EXCLUSIVE FS=FREE IF SHARED K=KEEP <BLANK>=USE DEFAULTS
112	(70)	SIGNED	4	RSDEVIATION	DEVIATION OF ABOVE (MAX)
116	(74)	CHARACTER	46	RSDESC	RESOURCE DESCRIPTION
162	(A2)	CHARACTER	6	RSLUDATECP	DATE OF LAST UPD in CP
168	(A8)	CHARACTER	4	RSLUTIMECP	TIME OF LAST UPD IN CP
			2	CHAR(2) , RESERVED	
172	(AC)	SIGNED	4	RSGQUANT	GLOBAL (OVERRIDE) QUANTITY
176	(B0)	CHARACTER	1	RSGAVAIL	GLOBAL (OVERRIDE) AVAILABILLITY
177	(B1)	CHARACTER	1	RSSELAVAIL	AVAIL FILTER
	NEXT HAD TO BE MOVED DUE TO CONFLICT WITH V2 ENH. WITH FLAG XB				
178	(B2)	CHARACTER	8	RSRLUTS	LAST UPDATE TIMESTAMP
186	(BA)	CHARACTER	8	RSLUSERCP	LAST UPDATING USER IN CP
194	(C2)	CHARACTER	1	RSHIPER	DLF RESOURCE, Y OR N
195	(C3)	CHARACTER	1	RSUSEDFOR	USED FOR (N|P|C|B)
196	(C4)	CHARACTER	1	RSSELWAIT	WAIT FILTER
197	(C5)	CHARACTER	1	RSSELALCS	ALLOCATED FILTER
198	(C6	CHARACTER	1	RSONCOMPLETE	ON COMPLETE OPTION: Y=SET TO YES N=SET TO NO R=RESET <BLANK>=USE DEFAULTS
199	(C7)	CHARACTER	1	RSLASTMODTYPE	LAST UPDATE OF GLOBAL AVAILABILITY DUE TO: E = EVENT APPLIED M = MAX LIMIT L = LIFESPAN TIE C = ON COMPLETE ACTION <BLANK> = OTHER
200	(C8)	SIGNED	4	RSIVLNUM	NUMBER OF INTERVALS
204	(CC)	SIGNED	4	RSIVLOFF	OFFSET
208	(D0)	SIGNED	4	RSIVLCUR	CURRENT INTERVAL
212	(D4)	SIGNED	4	RSIVLDOFF	RESERVED
216	(D8)	SIGNED	4	RSUSAGECOUNTER	USAGE COUNTER: INCREASED BY 1 EACH TIME A JOB ALLOCATES THE SR. RESET TO 0 WHEN MAX USAGE LIMIT IS EXCEEDED.
220	(DC)	SIGNED	4	RSMAXUSAGELIMIT	MAX USAGE LIMIT: 0 = DO NOT USE COUNTERS (THIS IS THE DEFAULT)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
224	(E0)	CHARACTER	1	RSMAXUSAGETYPE	
					MAX USAGE TYPE: Y = SET TO YES N = SET TO NO <BLANK> = RESET TO BLANK
225	(E1)	CHARACTER	1	RSTIELIFEACT	S TIE ACTION
226	(E2)	CHARACTER	10	RSTIELIFEDAT	S TIE DATE AND TIME
236	(EC)	CHARACTER	4	*	FREE
240	(F0)	CHARACTER		RSEND	END OF OBJECT HEADER

INTERVAL DESCRIPTION ARRAY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	48	RSIVL (*)	INTERVAL
0	(0)	SIGNED	4	RSIDAY	DATE
4	(4)	CHARACTER	6	RSIDATE	DATE
10	(A)	CHARACTER	2	*	RESERVED
12	(C)	CHARACTER	4	RSIFTIME	FROM TIME
16	(10)	CHARACTER	4	RSITTIME	TO TIME
20	(14)	SIGNED	4	RSIQUANT	MAX TO ALLOCATE
24	(18)	SIGNED	4	RSIQUANTD	MAX TO ALLOCATE (PLAN)
28	(1C)	SIGNED	4	RSIWSCNUM	NUMBER OF CONNECTED WS
32	(20)	CHARACTER	1	RSIAVAIL	AVAILABLE (Y/N)
33	(21)	CHARACTER	1	RSIAVAILD	AVAILABLE (Y/N) (PLAN)
34	(22)	CHARACTER	1	*	RESERVED
35	(23)	BITSTRING	1	RSIFLAGS	FLAG BYTE
		1...		RSIMODIFY	CHANGED IN DIALOG
		.111 1111		*	RESERVED
36	(24)	SIGNED	4	RSIWSOFF	WS OFFSET
40	(28)	CHARACTER	8	*	RESERVED
48	(30)	CHARACTER		RSIEND	END

INTERVAL DATA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	96	RSIVLDATA	
0	(0)	SIGNED	4	RSIMAXUSED	MAXIMUM CONCURRENT ALLOCATED
4	(4)	SIGNED	4	RSINOWUSES	CURRENTLY USED SHARED
8	(8)	SIGNED	4	RSINOWUSEX	CURRENTLY USED EXCLUSIVE
12	(C)	SIGNED	4	RSINOWALLX	CURRENT ALLOC ALL X
16	(10)	SIGNED	4	RSINOWALLS	CURRENT ALLOC ALL SHARED
20	(14)	SIGNED	4	RSIALCTIME	TIME ALLOCATED
24	(18)	SIGNED	4	RSIWAITTIME	TIME WAITING
28	(1C)	SIGNED	4	RSIUAVATIME	TIME UNAVAILABLE
32	(20)	SIGNED	4	RSIALCFAIL	ALLOCATION FAILIURES
36	(24)	SIGNED	4	RSICONFAIL	WS CONNECT FAILURES (DP ONLY)
40	(28)	SIGNED	4	RSIMADEUA	MADE UNAVAILABLE TIMESTAMP
44	(2C)	SIGNED	4	RSIALCST	ALLOCATION STARTED
48	(30)	SIGNED	4	RSIWAITST	WAIT STARTED

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
52	(34)	SIGNED	4	RSISTTIME	START TIME OF RSID BLOCK
56	(38)	SIGNED	4	RSIENDTIME	END TIME OF THIS IVLD BLOCK
60	(3C)	CHARACTER	10	RSISTTIMEC	START DATE & TIME IN CHAR
60	(3C)	CHARACTER	6	RSISTTIMEC_D	DATE
66	(42)	CHARACTER	4	RSISTTIMEC_T	TIME
70	(46)	CHARACTER	10	*	RESERVED
80	(50)	SIGNED	4	* (4)	RESERVED
96	(60)	CHARACTER		RSIDEND	END

WORKSTATION CONNECT ARRAY (ONE FOR EACH IVL).

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
0	(0)	STRUCTURE	8	RSCONWS (*)	
0	(0)	CHARACTER	4	RSCWSDNAME	WORK STATION NAME
4	(4)	CHARACTER	4	*	RESERVED
8	(8)	CHARACTER		RSCEND	END

Cross reference

Name	Hex Offset	Hex Value	Level
RS	0		1
RSCEND	8		2
RSCONWS	0		1
RSCWSDNAME	0		2
RSDESC	74		2
RSDEVIATION	70		2
RSEND	F0		2
RSEYE	4C		2
RSGAVAIL	B0		2
RSGQUANT	AC		2
RSGROUP	5C		2
RSHIPER	C2		2
RSIALCFAIL	20		2
RSIALCST	2C		2
RSIALCTIME	14		2
RSIAVAIL	20		2
RSIAVAILD	21		2
RSICONFAIL	24		2
RSID	0		3
RSIDATE	4		2
RSIDAY	0		2
RSIDEND	60		2
RSIEND	30		2
RSIENDTIME	38		2

Name	Hex Offset	Hex Value	Level
RSIFLAGS	23		2
RSIFTIME	C		2
RSIMADEUA	28		2
RSIMAXUSED	0		2
RSIMODIFY	23	80	3
RSINOWALLS	10		2
RSINOWALLX	C		2
RSINOWUSES	4		2
RSINOWUSEX	8		2
RSIQUANT	14		2
RSIQUANTD	18		2
RSISTTIME	34		2
RSISTTIMEC	3C		2
RSISTTIMEC_D	3C		3
RSISTTIMEC_T	42		3
RSITTIME	10		2
RSIUAVATIME	1C		2
RSIVL	0		1
RSIVLCUR	D0		2
RSIVLDATA	0		1
RSIVLDOFF	D4		2
RSIVLNUM	C8		2
RSIVLOFF	CC		2
RSIWAITST	30		2
RSIWAITTIME	18		2
RSIWSCNUM	1C		2
RSIWSOFF	24		2
RSKEY	0		2
RSLASTMODETYPE	C7		2
RSLUDATE	64		2
RSLUDATECP	A2		2
RSLUSER	54		2
RSLUSERCP	BA		2
RSLUTIME	6A		2
RSLUTIMECP	A8		2
RSMAXUSAGELIMIT	DC		2
RSMAXUSAGETYPE	E0		2
RSNAME	2		3
RSOBJDESC	40		2
RSOBJIDX	44		3
RSOCDEFWSC	48	10	4

Name	Hex Offset	Hex Value	Level
RSOCONERR	48	40	4
RSOCUSEDFOR	48	20	4
RSODYNADD	48	08	4
RSOFLAGS	48		3
RSONCOMPLETE	C6		2
RSONERROR	6E		2
RSORODMA	49		3
RSORODMD	4B		3
RSORODMQ	4A		3
RSOUPDATE	48	80	4
RSRLUTS	B2		2
RSSELALCS	C5		2
RSSELAVAIL	B1		2
RSSELWAIT	C4		2
RSTIELIFEACT	E1		2
RSTIELIFEDAT	E2		2
RSUSAGECOUNTER	D8		2
RSUSEDFOR	C3		2
RSVER	50		2
RSW	0		1
RSWALCNUM	20		2
RSWALCTIME	24		2
RSWALCTYPE	10		2
RSWCHANGED	A	40	3
RSWCONTTIE	A	80	3
RSWEND	2C		2
RSWEYE	4		2
RSWFLAGS	A		2
RSWNEXT	0		2
RSWOCC	C		3
RSWOP	F		3
RSWOPER	C		2
RSWVER	8		2
RSWWS	1C		2

RSRR - Special resource report record

Name : DCLRSRR

Function:
This segment maps the special resource report records in the EQQCXDS.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	136	RSR	RESOURCE INSTANCE STRUCTURE
0	(0)	CHARACTER	64	RSRKEY	KEY STRUCTURE
0	(0)	CHARACTER	2	RSRID	USED BY FOR IO METHODS
2	(2)	CHARACTER	44	RSRNAME	PLAN INSTANCE KEY
46	(2E)	CHARACTER	10	RSRFROM	REPORT INTERVAL START
46	(2E)	CHARACTER	6	RSRDATE	DATE
52	(34)	CHARACTER	4	RSRTIME	TIME
56	(38)	CHARACTER	6	*	RESERVED FOR ADDITIONAL KEY ...FIELDS
64	(40)	BITSTRING	1	RSRFLAGS	
		1...		RSRDYNADD	ON - DYNADDED
		.1..		RSRUPDCON	ON - UPDATED BY CONTROLLER
		..11 1111		*	SPARE
65	(41)	CHARACTER	3	*	SPARE
68	(44)	CHARACTER	4	RSREYEC	EYE CATCHER
72	(48)	SIGNED	4	RSRVERSON	VERSION NUMBER
76	(4C)	CHARACTER	10	RSRTO	IVL END
76	(4C)	CHARACTER	6	RSREDATE	DATE
82	(52)	CHARACTER	4	RSRETIME	TIME
86	(56)	CHARACTER	2	*	SPARE
88	(58)	CHARACTER	48	RSRPARTS	STATS SECTION
88	(58)	SIGNED	4	RSRMAXPLAN	MAXIMUM CONCURRENT PLAN
92	(5C)	SIGNED	4	RSRMAXUSED	MAXIMUM CONCURRENT ALLOCATED
96	(60)	SIGNED	4	RSRIDLETIME	SECONDS IDLE
100	(64)	SIGNED	4	RSRWAITTIME	SECONDS WAITING
104	(68)	SIGNED	4	RSRAVAILTIME	SECONDS AVAILABLE
108	(6C)	SIGNED	4	RSRALCFAIL	ALLOCATION FAILURES
112	(70)	SIGNED	4	RSRCONFAIL	WS CONNECT FAILURES
116	(74)	SIGNED	4	* (4)	RESERVED
132	(84)	CHARACTER	1	RSRAVAIL	AVAILABLE (Y/N)
133	(85)	CHARACTER	3	*	RESERVED
136	(88)	CHARACTER		RSREND	END OF OBJECT HEADER

Cross reference

Name	Hex Offset	Hex Value	Level
RSR	0		1
RSRALCFAIL	6C		3
RSRAVAIL	84		3
RSRAVAILTIME	68		3
RSRCONFAIL	70		3

Name	Hex Offset	Hex Value	Level
RSRDATE	2E		4
RSRDYNADD	40	80	3
RSREDATE	4C		3
RSREND	88		2
RSRETIME	52		3
RSREYEC	44		2
RSRFLAGS	40		2
RSRFROM	2E		3
RSRID	0		3
RSRIDLETIME	60		3
RSRKEY	0		2
RSRMAXPLAN	58		3
RSRMAXUSED	5C		3
RSRNAME	2		3
RSRPARTS	58		2
RSRTIME	34		4
RSRTO	4C		2
RSRUPDCON	40	40	3
RSRVERSON	48		2
RSRWAITTIME	64		3

SSX - Subsystem communication vector table extension

Name : DCLSSX

Function:
This segment maps the Subsystem Extension block (SSX) for the subsystem. The SSX IEFJSCVT control block is addressed via the SSCTSUSE pointer of the SSCT block mapped by the macro. In some cases there are two SSX blocks, the current, active SSX addressed via SSCTSUSE, and the old, inactive SSX addressed by SSCTSUS2. The SSX block is allocated inCSA (subpool 241) in key 0 storage. The SSX block is built by the EQQINIT module during Master Scheduler initialization, or by EQQMAJOR during subsystem initialization when theBUILDSSX keyword has been specified. The active SSX block is freed and the inactive SSX is made active during EQQMAJOR termination when there is an inactive SSX.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	1664	ssx	ssvt extension block

SSX block identifier and code level

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	4	ssxdesc	block descriptor = 'SSX '
4	(4)	CHARACTER	10	ssxlevel	SSX block code level
4	(4)	CHARACTER	2	ssxver	block mapping version

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
6	(6)	CHARACTER	8	ssxfmid	OPCESA tracker fmid
14	(E)	SIGNED	2	ssxsize	size of the SSX block

Status and option flags

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
16	(10)	BITSTRING	2	ssxflags	subsystem status flags
		1...		ssxjhold	1 holdjob(yes) requested
		.1..		ssxsteps	1 generate all step events
		..1.		ssxjsub	1 job submission active
		...1		ssxjchld	1 jcc wants held data
		.... 1...		ssxett	1 ett function active
		.... .1..		ssxhghcc	1 use highest cc for a job
		.... ..1.		ssxlstcc	1 use last exec step's cc
		1		ssxuhold	1 holdjob(user) requested
17	(11)	1...		ssxnzero	1 gen only nzero cc step events
		.1..		ssxar	1 auto recovery task is active
		..1.		ssxgdgns	1 GDG_NONST ABIC CPRC
		...1		ssxwrn	1 JES sp-lvl warn msg issued
		.... 1...		ssxdiae1	1 diagnose event wanted
		.... .1..		ssxsrsrc	1 SR.SRNAME is a resource
		.... ..1.		ssxftwjsub	1 opc will submit jobs
		1		ssxrpass	1 RCLPASS = YES BNIA
18	(12)	BITSTRING	4	ssxflag2	more status flags
		1...		ssxwrn1	1 JES exit51 warn msg iss. CCRA
		.1..		ssxrlwst	rl.wsstat subresource
		..1.		ssxrlwsn	rl.wsname subresource
		...1		ssxfipl	1 no dslst loaded for IPL
		.... 1...		ssxrladn	rl.adname subresource
		.... .1..		ssxEmDeb	Debug in ZEVSX 41c
		.... ..1.		ssxJCC	1 JCC started D90C
		1		ssxSSCbig	1 SSC big buffer
22	(16)	CHARACTER	1	ssxjes	JES type A=JES2, B=JES3
23	(17)	CHARACTER	1	ssxcjes	JES command character
24	(18)	CHARACTER	8	ssxnode	own nje node name

EQQSSCM load module management

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
32	(20)	CHARACTER	16	ssxsscm_cur	current eqqsscm load module
32	(20)	CHARACTER	8	ssxcmname	eqqsscm load module name
40	(28)	ADDRESS	4	ssxcmcatp	eqqsscm entry point address
44	(2C)	ADDRESS	4	ssxcmssrx	address of eqqzssrx
48	(30)	CHARACTER	16	ssxsscm_ipl	previously loaded eqqsscm
48	(30)	CHARACTER	8	ssxcmname	
56	(38)	ADDRESS	4	ssxcmcatp	
60	(3C)	ADDRESS	4	ssxcmssrx	

Allocated resources

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
64	(40)	ADDRESS	4	ssxareas	chain of allocated csa areas
68	(44)	ADDRESS	4	ssxevtq	event wtr queue address
72	(48)	ADDRESS	4	ssxsscma	perm loaded eqqsscm module addr
76	(4C)	SIGNED	4	ssxsscms	perm loaded eqqsscm module size
80	(50)	ADDRESS	4	ssxnf57A	EQQZNF57 address CCUA
84	(54)	SIGNED	4	ssxnf57S	EQQZNF57 size CCUA

SSX label offsets above this point are fixed The only change allowed is defining a bit in an existing flag byte or modifing one of the SSX level constants

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
88	(58)	CHARACTER		ssx_end_fixed	end of fixed ssx mapping

Control Block addresses

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
88	(58)	ADDRESS	4	ssxmcap	addr of mca in opc addr space
92	(5C)	ADDRESS	4	ssxascb	opc ascb address
96	(60)	ADDRESS	4	ssxssct	opc scvt address
100	(64)	ADDRESS	4	ssxSDEPp	ptr to EW SDEP parms
104	(68)	ADDRESS	4	ssxdsnew	data set trigger table
108	(6C)	ADDRESS	4	ssxnf41A	EQQZNF41 address CCUA
112	(70)	SIGNED	4	ssxnf41S	EQQZNF41 size CCUA
116	(74)	ADDRESS	4	ssxnf53A	EQQZNF53 address CHMA
120	(78)	SIGNED	4	ssxnf53S	EQQZNF53 size CHMA
124	(7C)	SIGNED	4	* (3)	reserved CHMC

SSC storage management addresses

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
136	(88)	ADDRESS	4	ssxsscc	addr of first SSC block
140	(8C)	ADDRESS	4	ssxsscfr	addr of first free SSC

D94A TWS storage management addresses D94A
D94A

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
144	(90)	CHARACTER	8	ssxtwsServerName	TWS Server Name 01c

bin31 reserved 01d D94C
DBCS options

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
152	(98)	CHARACTER	8	ssxdbcs	dbcs options

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
152	(98)	CHARACTER	2	ssxsorto	dbcs sort algorithm
154	(9A)	CHARACTER	1	ssxadidf	dbcs adid option D or E
155	(9B)	CHARACTER	1	ssxowidf	dbcs owner option D or E
156	(9C)	CHARACTER	4	*	reserved

Miscellaneous fields

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
160	(A0)	CHARACTER	8	ssxsjid	started task job number
168	(A8)	CHARACTER	8	ssxsjnm	started task job name
176	(B0)	CHARACTER	4	ssxsid	MVS system id
180	(B4)	CHARACTER	4	ssxssnm	submitting subsystem name
184	(B8)	CHARACTER	8	ssxclas	RACF resource class name
192	(C0)	CHARACTER	16	ssxjccls	JCC sysout classes
208	(D0)	CHARACTER	1	ssxprtev	prtev N=no, ' '=end, A=all
209	(D1)	CHARACTER	1	ssxjes3	JES3 type G=Global, L=Local
210	(D2)	CHARACTER	1	*	free D90C
211	(D3)	CHARACTER	1	*	free D54A
212	(D4)	CHARACTER	8	ssxjfmid	JES fmid
220	(DC)	CHARACTER	4	ssxdiajn	diagnose JOBSTC name
224	(E0)	BITSTRING	4	ssxiecb	JES input service ECB
228	(E4)	BITSTRING	4	ssxeecb	JES execution ECB
232	(E8)	UNSIGNED	4	ssxptkn	current MGCR ptoken
236	(EC)	SIGNED	4	ssxsdtr	used as indicator only celc
240	(F0)	SIGNED	4	ssxutime	time of last uerror event
244	(F4)	SIGNED	4	ssxatime	time of last aerror event
248	(F8)	SIGNED	4	ssxsznew	size of DS trigger table
252	(FC)	SIGNED	4	*	reserved
256	(100)	SIGNED	4	ssxrtoka	addr space res mgr token
260	(104)	SIGNED	4	ssxrtokt	task level res mgr token
264	(108)	SIGNED	4	ssx_pif_cw	pif cw base year
268	(10C)	CHARACTER	6	ssx_pif_hd	pif high date format
274	(112)	CHARACTER	5	ssxclnjob	stand alone jobname prefix D90C
279	(117)	CHARACTER	1	*	reserved D90A

OPC Queue Addresses

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
280	(118)	CHARACTER	104	ssxqtable	OPC queue addresses
280	(118)	ADDRESS	4	* (26)	array of queue addresses

OPC Queue Server blocks

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
384	(180)	CHARACTER	48	ssxques (26)	queue server blocks mapped by dclque D45C
1632	(660)	CHARACTER	32	*	reserved slack space
1664	(680)	CHARACTER		ssxend	end of ssx

OPC Queue array mapping
When adding a new queue pointer do not reuse a used array entry
The maximum array index that can be used is defined by ssxnumq
Increased ssxnumq for future development

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
280	(118)	STRUCTURE	104	ssxqptrs	array of queue addresses
280	(118)	ADDRESS	4	*	reserved by event writer 1
284	(11C)	ADDRESS	4	ssxgenq	gen service queue 2
288	(120)	ADDRESS	4	ssxjccq	jcc service queue 3
292	(124)	ADDRESS	4	ssxmgrq	event mgr queue 4
296	(128)	ADDRESS	4	ssxmsgq	major msg queue 5
300	(12C)	ADDRESS	4	ssxappq	appc queue 6
304	(130)	ADDRESS	4	ssxrtrq	router queue 7
308	(134)	ADDRESS	4	ssxtipq	tcpip queue (ex CatMgt) 8CMDa
312	(138)	ADDRESS	4	ssxsubq	job submit queue 9
316	(13C)	ADDRESS	4	ssxncfq	ncf queue 10
320	(140)	ADDRESS	4	ssxexq	exa queue 11
324	(144)	ADDRESS	4	*	free 1261c
328	(148)	ADDRESS	4	ssxrodq	rodm queue 13L3A
332	(14C)	ADDRESS	4	ssxatrq	appc tracker queue 14lza
336	(150)	ADDRESS	4	ssxsrvq	Server queue D54C 15x6a
340	(154)	ADDRESS	4	ssxarcq	FL task queue D45C 16D54A
344	(158)	ADDRESS	4	ssxpsuq	PSU task queue 17D90A
348	(15C)	ADDRESS	4	ssxtwsq	TWS task queue D94A 18
352	(160)	ADDRESS	4	ssxmonq	MON task queue 1932a
356	(164)	ADDRESS	4	ssxcphq	CPH task queue 20CRFA
360	(168)	ADDRESS	4	ssxmajq	major queue 2141a
364	(16C)	ADDRESS	4	ssxhttq	HTTP task queue 2251a
368	(170)	ADDRESS	4	* (4)	free for future 23-2651c

Some SSX constants

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	24	ssxsdepf	SDEPFILTER options
0	(0)	SIGNED	2	ssxsdstart	- start point (offset)
2	(2)	SIGNED	2	ssxsdlen	- value length
4	(4)	CHARACTER	20	ssxsdval	- value

Cross reference

Name	Hex Offset	Hex Value	Level
ssx	0		1
ssx_end_fixed	58		2
ssx_pif_cw	108		2
ssx_pif_hd	10C		2
ssxadidf	9A		3
ssxappq	12C		2
ssxar	11	40	3

Name	Hex Offset	Hex Value	Level
ssxarcq	154		2
ssxareas	40		2
ssxascb	5C		2
ssxatime	F4		2
ssxatrq	14C		2
ssxcjes	17		2
ssxclas	B8		2
ssxclnjob	112		2
ssxcmcatp	28		3
ssxcmcatp	38		3
ssxcmname	30		3
ssxcmname	20		3
ssxcmssrx	2C		3
ssxcmssrx	3C		3
ssxcphq	164		2
ssxdbcs	98		2
ssxdesc	0		2
ssxdiae1	11	08	3
ssxdiajn	DC		2
ssxdsnew	68		2
ssxeecb	E4		2
ssxEmDeb	12	04	3
ssxend	680		2
ssxett	10	08	3
ssxevtq	44		2
ssxexq	140		2
ssxfipl	12	10	3
ssxflags	10		2
ssxflag2	12		2
ssxfmid	6		3
ssxftwjsub	11	02	3
ssxgdgns	11	20	3
ssxgenq	11C		2
ssxhghcc	10	04	3
ssxhttq	16C		2
ssxiecb	E0		2
ssxJCC	12	02	3
ssxjccls	C0		2
ssxjccq	120		2
ssxjchld	10	10	3
ssxjes	16		2

Name	Hex Offset	Hex Value	Level
ssxjes3	D1		2
ssxjfmid	D4		2
ssxjhold	10	80	3
ssxjsub	10	20	3
ssxlevel	4		2
ssxlstcc	10	02	3
ssxmajq	168		2
ssxmcap	58		2
ssxmgrq	124		2
ssxmonq	160		2
ssxmsgq	128		2
ssxncfq	13C		2
ssxnf41A	6C		2
ssxnf41S	70		2
ssxnf53A	74		2
ssxnf53S	78		2
ssxnf57A	50		2
ssxnf57S	54		2
ssxnode	18		2
ssxnzero	11	80	3
ssxowidf	9B		3
ssxprtev	D0		2
ssxpsuq	158		2
ssxptkn	E8		2
ssxqptrs	118		1
ssxqtable	118		2
ssxques	180		2
ssxrladn	12	08	3
ssxrlwsn	12	20	3
ssxrlwst	12	40	3
ssxrodq	148		2
ssxrpass	11	01	3
ssxrtoka	100		2
ssxrtokt	104		2
ssxrtrq	130		2
ssxsdepf	0		1
ssxSDEPp	64		2
ssxsdlen	2		2
ssxsdstart	0		2
ssxsdtr	EC		2
ssxsdval	4		2

Name	Hex Offset	Hex Value	Level
ssxsid	B0		2
ssxsize	E		2
ssxsjid	A0		2
ssxsjnm	A8		2
ssxsorto	98		3
ssxsrsrc	11	04	3
ssxsrvq	150		2
ssxSSCbig	12	01	3
ssxsscc	88		2
ssxsscfr	8C		2
ssxsscm_cur	20		2
ssxsscm_ipl	30		2
ssxsscma	48		2
ssxsscms	4C		2
ssxssct	60		2
ssxssnm	B4		2
ssxsteps	10	40	3
ssxsubq	138		2
ssxsznew	F8		2
ssxtipq	134		2
ssxtwsq	15C		2
ssxtwsServerName	90		2
ssxuhold	10	01	3
ssxutime	F0		2
ssxver	4		3
ssxwrn	11	10	3
ssxwrn1	12	80	3

TOC - Table of contents for DP reports

Name : DCLTOC

Function:
This segment declares the table of contents record in a daily plan report.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	88	TABCONT	TABLE OF CONTENTS (DP)
0	(0)	CHARACTER	6	TABGIM	GENERAL INFORMATION DJAC
6	(6)	CHARACTER	6	TABDOP	DAILY OPERATION PLAN DJAC
12	(C)	ADDRESS	4	TABWS	WORKSTATION(S) PLANS
16	(10)	CHARACTER	6	TABSUM	SUMMARY OF COMPLETED APPDJAC
22	(16)	CHARACTER	6	TABCMP	COMPLETED APPLICATIONS DJAC
28	(1C)	CHARACTER	6	TABOPE	OPERATION IN ERROR DJAC

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
34	(22)	CHARACTER	6	TABWSU	WORKSTATION UTILIZATION DJAC
40	(28)	CHARACTER	6	TABERS	ERROR STATISTICS DJAC
46	(2E)	CHARACTER	6	TABFBR	MISSED FEEDBACK DJAC
52	(34)	CHARACTER	6	tabhpr	planned res util hisL5ADJAC
58	(3A)	CHARACTER	6	tabpru	planned res util repL5ADJAC
64	(40)	CHARACTER	6	tabaru	actual res util repL5ADJAC
70	(46)	CHARACTER	6	TABMST	MESSAGES DJAC
76	(4C)	CHARACTER	6	TABCri	Critical Path 31A41CDJAC
82	(52)	CHARACTER	6	TABxst	X status 41ADJAC

Cross reference

Name	Hex Offset	Hex Value	Level
tabaru	40		2
TABCMP	16		2
TABCONT	0		1
TABCri	4C		2
TABDOP	6		2
TABERS	28		2
TABFBR	2E		2
TABGIM	0		2
tabhpr	34		2
TABMST	46		2
TABOPE	1C		2
tabpru	3A		2
TABSUM	10		2
TABWS	C		2
TABWSU	22		2
TABxst	52		2

TRG - Data set trigger table layout

Name : DCLTRG

Function:
This segment maps EQQEVLIB members, that are input data for the controller, and the EQQEVLST member of EQQJCLIB, that is input data for the tracker. The mapping varies depending on the type of the specific itemin the triggering selection table, that is one of the following:
Header
Trigger condition
Filter criteria
Filter value
Action
Special resource related information
Data loaded in Common Service Area (CSA), mapped by TRGTABLEHEADER and
following structures.

```
Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     8  TRGITABLEHEADER
    0    (0) CHARACTER     1  TRGITABLECHECK   =0 ALWAYS
    1    (1) CHARACTER     1  TRGITABLEVERSION
    2    (2) CHARACTER     1  TRGITABLERELEASE
    3    (3) CHARACTER     1  *                BLANK
    4    (4) CHARACTER     4  TRGITABLECRC     CRC IDENTIFYING THE TABLE C
                                               MUST BE WORD ALLIGNED

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     9  TRGICONDITION
    0    (0) CHARACTER     1  TRGICONDMONITOR  S FOR SMF
    1    (1) CHARACTER     1  TRGICONDDRAFT    D OR BLANK
    2    (2) CHARACTER     3  TRGICONDTYPE     CONDITION TO BE TESTED
    5    (5) CHARACTER     1  *                BLANK
    6    (6) CHARACTER     3  TRGICONDFILTERSNUMBER
    9    (9) CHARACTER        TRGICONDDATA

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE    11  TRGIFILTER
    0    (0) CHARACTER     1  *                  ;
    1    (1) CHARACTER     3  TRGIFILTERFIELD
    4    (4) CHARACTER     1  *                  BLANK
    5    (5) CHARACTER     1  TRGIFILTERVALUESNUMBERIDENT
    6    (6) CHARACTER     2  TRGIFILTERVALUESNUMBER

    8    (8) CHARACTER     2  TRGIFILTEROPERATOR ==, <>, >=, <= ><(RANGE)

   10    (A) CHARACTER     1  TRGIFILTEROREAND

   11    (B) CHARACTER        TRGIFILTERDATA

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     *  TRGIFILTERVALUE
    0    (0) CHARACTER     1  *                BLANK
    1    (1) CHARACTER     3  TRGIFILTERVALUELEN

    4    (4) CHARACTER     1  *                :
    5    (5) CHARACTER     3  TRGIFILTERVALUEOFFSET

    8    (8) CHARACTER     1  *                :
    9    (9) CHARACTER     *  TRGIFILTERVALUESTR

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     4  TRGIACTION
    0    (0) CHARACTER     1  *              ;
    1    (1) CHARACTER     2  TRGIACTIONTYPE ACTION INDICATOR
    3    (3) CHARACTER     1  *              BLANK
    4    (4) CHARACTER        TRGIACTIONINFO

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     9  TRGISRSTATACTIONINFO
    0    (0) CHARACTER     1  TRGISRAVAILABILITY

    1    (1) CHARACTER     1  TRGISRLIFACTION
    2    (2) CHARACTER     5  TRGISRLIFTIME
    7    (7) CHARACTER     2  *              ;; THIS FIELD IDENTIFY THE
                                             END OF CONDITION.......

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     4  TRGTABLEHEADER
    0    (0) CHARACTER     4  TRGTABLECRC
    4    (4) CHARACTER        TRGTABLEDATA

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     5  TRGCONDITION
    0    (0) UNSIGNED      1  TRGCONDMONITOR
    1    (1) UNSIGNED      1  TRGCONDTYPE
    2    (2) UNSIGNED      2  TRGCONDNEXTOFFSET
    4    (4) UNSIGNED      1  TRGCONDFILTERSNUMBER
    5    (5) CHARACTER        TRGCONDDATA

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     2  TRGFILTER      GENERIC EVENT FILTER
    0    (0) UNSIGNED      1  TRGFILTERFIELD FIELD IN SMF RECORD
    1    (1) 1111  ....       TRGFILTERVALUESNUMBER
             ....  111.       TRGFILTEROPERATOR
             ....  ...1       TRGFILTEROREAND
    2    (2) CHARACTER        TRGFILTERDATA  START OF VALUES

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     *  TRGFILTERVALUE
    0    (0) UNSIGNED      1  TRGFILTERVALUELEN
    1    (1) UNSIGNED      1  TRGFILTERVALUEOFFSET
    2    (2) CHARACTER     *  TRGFILTERVALUESTR

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     1  TRGACTION
             1111  ....       TRGACTIONTYPE

Offsets      Type     Length  Name             Description
    0    (0) STRUCTURE     6  TRGSRSTATACTION  NOT USED
             1111  ....       *
```

Cross reference

```
HEX          HEX
 NAME                                         OFFSET      VALUE     LEVEL
 ____                                         ______     ________   _____
 TRGACTION                                         0                    1
 TRGACTIONTYPE                                     0           F0       2
 TRGCONDDATA                                       5                    2
 TRGCONDFILTERSNUMBER                              4                    2
 TRGCONDITION                                      0                    1
 TRGCONDMONITOR                                    0                    2
 TRGCONDNEXTOFFSET                                 2                    2
 TRGCONDTYPE                                       1                    2
 TRGFILTER                                         0                    1
 TRGFILTERDATA                                     2                    2
 TRGFILTERFIELD                                    0                    2
 TRGFILTEROPERATOR                                 1           0E       2
 TRGFILTEROREAND                                   1           01       2
 TRGFILTERVALUE                                    0                    1
 TRGFILTERVALUELEN                                 0                    2
 TRGFILTERVALUEOFFSET                              1                    2
 TRGFILTERVALUESNUMBER                             1           F0       2
 TRGFILTERVALUESTR                                 2                    2
 TRGIACTION                                        0                    1
 TRGIACTIONINFO                                    4                    2
 TRGIACTIONTYPE                                    1                    2
 TRGICONDDATA                                      9                    2
 TRGICONDDRAFT                                     1                    2
 TRGICONDFILTERSNUMBER                             6                    2
 TRGICONDITION                                     0                    1
 TRGICONDMONITOR                                   0                    2
 TRGICONDTYPE                                      2                    2
 TRGIFILTER                                        0                    1
 TRGIFILTERDATA                                    B                    2
 TRGIFILTERFIELD                                   1                    2
 TRGIFILTEROPERATOR                                8                    2
 TRGIFILTEROREAND                                  A                    2
 TRGIFILTERVALUE                                   0                    1
 TRGIFILTERVALUELEN                                1                    2
 TRGIFILTERVALUEOFFSET                             5                    2
```

TRGIFILTERVALUESNUMBER	6		2
TRGIFILTERVALUESNUMBERIDENT	5		2
TRGIFILTERVALUESTR	9		2
TRGISRAVAILABILITY	0		2
TRGISRLIFACTION	1		2
TRGISRLIFTIME	2		2
TRGISRSTATACTIONINFO	0		1
TRGITABLECHECK	0		2
TRGITABLECRC	4		2
TRGITABLEHEADER	0		1
TRGITABLERELEASE	2		2
TRGITABLEVERSION	1		2
TRGSRAVAILABILITY	0	0C	2
TRGSRLIFACTION	0	03	2
TRGSRLIFTIME	1		2
TRGSRSTATACTION	0		1
TRGTABLECRC	0		2
TRGTABLEDATA	4		2
TRGTABLEHEADER	0		1

TRL - Job-tracking and audit record

Name : DCLTRL

Function:
This segment maps tracklog records in:
1. The JT log data sets. There are at least two such data sets.
2. The JT log backup data sets. These data sets are optional.
3. The JT log archive data set. This data set is required.
4. The tracklog data set. This data set is optional.

JT log data sets are created by an active controller system. The tracklog data set is created by
a Daily Plan batch job. The Daily Plan batch job (plan next period and replan current period)
will copy the contents of the archive JT log to the tracklog data set.
The job also copies all new current plan records type 01, 02 and 03 onto the tracklog data set.
It can optionally contain record types 01, 02, 03, and 04 from the old current plan data set.
The tracklog data set is normally allocated disp=mod by the Daily Plan batch job.

Note: In the JT log, 6-char dates appear in the internal date format, which uses 1972 as
the century window base year. This means that the year 1972 is represented as 00.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	*	trlrec	
0	(0)	CHARACTER	52	trlcom	common section
0	(0)	SIGNED	2	trlsize	size of this record
2	(2)	UNSIGNED	2	trlbdw	always zeroes
4	(4)	CHARACTER	3	trlopc	should always contain 'OPC'
7	(7)	CHARACTER	2	trllogid	curr id of this opc log.
9	(9)	CHARACTER	2	trlrctyp	record type
11	(B)	UNSIGNED	1	trlrcver	record mapping version.
12	(C)	CHARACTER	4	trlevdat	event date, format 00yydddf
16	(10)	UNSIGNED	4	trlevtim	event time, format hhmmssth
20	(14)	CHARACTER	19	trlcpkey	contains current plan key for type 01, 02, 03, 22 COBC
20	(14)	CHARACTER	6	*	operation record key present for 23,26,29,33 rec
20	(14)	SIGNED	4	trlcpeut	elapsed time for mcp present for type 24 rec

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
26	(1A)	CHARACTER	8	trluser	dialog user name
34	(22)	UNSIGNED	2	trlsseq	submit seq number present for type 25 rec
39	(27)	CHARACTER	1	trlfill	flags
		1...		trlfirst	first of a series of recs
		.1..		trllast	last of a series of recs
		..1.		trlstc	log rec is for STC oper
		...1		trlwto	log rec is for WTO oper
		.... 1...		trlresub	re-submit type 25
		.... .1..		trljrpfl	ETT JNR=Y matched ev't
		.... ..1.		trloldcp	from old current plan
		1		trljrnfl	ETT JNR=N matched ev't
40	(28)	SIGNED	4	trllengt	total size of logged evt
44	(2C)	SIGNED	4	trlgmtof	GMT offset
48	(30)	CHARACTER	1	trlfill2	flags AGZA
		1...		trlckptr	performing CKPT refresh
		.1..		trlsa	SA submission CHWA
		..11 1111		*	Reserved CHWC
49	(31)	CHARACTER	3	*	Reserved AGZCLVAQVC
52	(34)	CHARACTER	*	trlrcbdy	end of common section

TRL 20 JT STARTED EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	1	trlbdy20	body of jt started event
52	(34)	CHARACTER	1	trlmod20	mode jt was started in t (track) or u (update)

TRL 23 OPERATION EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	80	trlbdy23	body of operation event
52	(34)	CHARACTER	16	trladi23	ad id for occurrence
68	(44)	CHARACTER	6	trliad23	mod input arr date
74	(4A)	CHARACTER	4	trliat23	mod input arr time
78	(4E)	CHARACTER	6	trloid23	operation id
78	(4E)	CHARACTER	4	trlwsn23	workstation name
82	(52)	SIGNED	2	trlopn23	operation number
84	(54)	CHARACTER	1	trlevt23	new operation status or 'X' = reset status or X status
85	(55)	CHARACTER	6	trldur23	reported duration
91	(5B)	CHARACTER	4	trlerc23	reported error code
95	(5F)	CHARACTER	1	*	various flags
		1...		trlfdb23	on = occ. is completed
		.1..		trlbyp23	on = bypass during restart
		..1.		trlman23	on = status set manually (gs)
		...1		trlX23	on = status X set
		 1111		*	reserved

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
96	(60)	CHARACTER	16	trlowi23	owner id
112	(70)	CHARACTER	8	trljbn23	jobname
120	(78)	UNSIGNED	4	trldurs23	duration in 100th*sec d88a
124	(7C)	CHARACTER	8	trlready23	start time wait opr d88a

TRL 24 MCP EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	*	trlbdy24	body of mcp-event
52	(34)	CHARACTER	*	trlmt024	mapped by dclmt0

TRL 25 SUBMIT EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	80	trlbdy25	body of submit event
52	(34)	CHARACTER	16	trlaid25	ad id for op
68	(44)	CHARACTER	10	trlia25	occ input arrival
68	(44)	CHARACTER	6	trliad25	mod input arr date
74	(4A)	CHARACTER	4	trliat25	mod input arr time
78	(4E)	CHARACTER	8	trljid25	blank for sub or 'RELEASE'
86	(56)	CHARACTER	8	trlddn25	sure ddname or zero
94	(5E)	CHARACTER	8	trljnm25	job name
102	(66)	CHARACTER	8	trlusr25	user id or 'OPC'
110	(6E)	BITSTRING	1	trlFlag	CCUA
		1...		trlWASUJ	on = from WASUJ CCUA
		.1..		trlrer25	on = it was rerouted
		..1.		trlexit13rc	on=rc 12 from exit13 DGWA
		...1 1111		*	CCUADGWC
111	(6F)	CHARACTER	1	trlTaskType25	plugin task type
112	(70)	CHARACTER	8	trltok_full25	operation token
112	(70)	CHARACTER	4	trltok_pre25	token prefix
116	(74)	UNSIGNED	4	trltok25	token value
120	(78)	CHARACTER	8	trlsubtoken25	submission token
128	(80)	CHARACTER	4	trlaltws25	alternate ws for rerouted

TRL 26 AUTO RECOVERY EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	27	trlbdy26	body of auto recovery
52	(34)	CHARACTER	16	trlaid26	ad id for op
68	(44)	CHARACTER	10	trlia26	occ input arrival
78	(4E)	CHARACTER	1	trlxst26	new extended status

TRL 27 MISSED FEEDBACK EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	62	trlbdy27	body of missed feedback rec
52	(34)	CHARACTER	16	trlaid27	ad id for op
68	(44)	CHARACTER	6	trliad27	mod input arr date
74	(4A)	CHARACTER	4	trliat27	mod input arr time
78	(4E)	CHARACTER	6	trloid27	operation id
84	(54)	CHARACTER	4	trledu27	estimated duration
88	(58)	CHARACTER	6	trladu27	actual duration
94	(5E)	CHARACTER	1	trlres27	reason for missed feedback '1' = limit exceeded '2' = ad record in use '3' = not enough space '4' = io problems '5' = DL before IA '6' = no run cycle found
95	(5F)	CHARACTER	1	trlorig27	origin for missed feedback '1' = operation duration '2' = operation deadline '3' = occurrence deadline
96	(60)	SIGNED	2	trlrid27	run cycle id
98	(62)	CHARACTER	6	trledl27	estimated deadline(oprun)
98	(62)	CHARACTER	2	trledld27	day offset DD
100	(64)	CHARACTER	4	trledlt27	time HHMM
104	(68)	CHARACTER	10	trladl27	actual deadline (operrun)
104	(68)	CHARACTER	6	trladld27	date YYMMDD
110	(6E)	CHARACTER	4	trladlt27	time HHMM

TRL 28 MISSED FEEDBACK EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	*	trlbdy28	body of feedback record
52	(34)	CHARACTER	16	trlaid28	ad id for occ
68	(44)	CHARACTER	6	trliad28	mod input arr date
74	(4A)	CHARACTER	4	trliat28	mod input arr time
78	(4E)	CHARACTER	6	trlodl28	old run cycle deadline
78	(4E)	CHARACTER	2	trlodld28	day offset DD
80	(50)	CHARACTER	4	trlodlt28	time HHMM
84	(54)	CHARACTER	6	trladl28	actual run cycle deadline
84	(54)	CHARACTER	2	trladld28	day offset DD
86	(56)	CHARACTER	4	trladlt28	time HHMM
90	(5A)	SIGNED	2	trlrid28	run cycle id
92	(5C)	SIGNED	2	trlopdl28	nr ops updtd for deadline
94	(5E)	SIGNED	2	trlopdu28	nr ops updtd for duration
96	(60)	SIGNED	2	trlop28	number of ops updated
98	(62)	CHARACTER	26	trlops28 (*)	updated operations
98	(62)	CHARACTER	6	trlopi28	operation id
104	(68)	CHARACTER	4	trlodu28	old duration
108	(6C)	CHARACTER	4	trladu28	actual duration
112	(70)	CHARACTER	6	trlopodl28	old operation deadline
112	(70)	CHARACTER	2	trlopodld28	day offset DD
114	(72)	CHARACTER	4	trlopodlt28	time HHMM
118	(76)	CHARACTER	6	trlopadl28	actual operation deadline
118	(76)	CHARACTER	2	trlopadld28	day offset DD
120	(78)	CHARACTER	4	trlopadlt28	time HHMM

TRL 29 AUTO TRACKING EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	*	trlbdy29	auto tracked event
52	(34)	CHARACTER	1	trltyp29	event logging reason 1 event was processed 2 event was suspended
53	(35)	CHARACTER	*	trlexr29	event record (see dcleqe)

code changed below

TRL 30 SPECIAL RESOURCE EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	*	trlbdy30	special resource event
52	(34)	CHARACTER	1	trltyp30	type of record R resource record E EXS event record A Actual report record (RSRR)
53	(35)	CHARACTER	1	trlreq30	request type U update a resource A add a resource
54	(36)	CHARACTER	1	trlfla30	flags
		1...		trlsrs30	ON= SRSTAT change
		.1..		trlchg30	COZA
		..11 1111		*	reserved COZCIJA
55	(37)	CHARACTER	1	*	reserved
56	(38)	CHARACTER	*	trlrsr30	data part

TRL 30 EXS SR EXS EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
56	(38)	STRUCTURE	*	trlexs30	event mapping
56	(38)	SIGNED	4	trlnode30	OPC node number
60	(3C)	SIGNED	4	trlrdrn30	event reader num in node
64	(40)	CHARACTER	8	trlpos30	event ds position
64	(40)	SIGNED	4	trlcyc30	write cycle number
68	(44)	SIGNED	4	trlrec30	record number in cycle
72	(48)	CHARACTER	*	trlexsr30	event record see dclexs

code changed above

TRL 31 ETT EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	*	trlbdy31	ett tab file maint ev
52	(34)	CHARACTER	1	trltyp31	event logged by 2 general service
53	(35)	CHARACTER	1	trlreq31	request type U update a criteria I add a criteria D delete a criteria
54	(36)	CHARACTER	*	trletc31	ett rec (see dcletc)

TRL 32 AUDIT TRAIL EVENT

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	*	trlbdy32	audit trail log record
52	(34)	SIGNED	2	trlkln32	size of key
54	(36)	SIGNED	2	trlrec32	amount of data this record
56	(38)	SIGNED	4	trltot32	total data size
60	(3C)	SIGNED	4	trlrem32	remaining data size
64	(40)	CHARACTER	4	trlfil32	logical file name
68	(44)	CHARACTER	1	trlacc32	access type R read access U update access I add a record D record deleted
69	(45)	BITSTRING	3	trlrsv32	not used
72	(48)	CHARACTER	64	trlkey32	record key blanks
72	(48)	CHARACTER	28	*	reserved JCLA
100	(64)	CHARACTER	8	trllib32	library read from JCLA
108	(6C)	CHARACTER	8	trlmem32	jcl member read JCLA
116	(74)	CHARACTER	1	trlchg32	record changed COZA
117	(75)	CHARACTER	19	*	reserved JCLACOZC
136	(88)	CHARACTER	*	trldat32	record data

TRL 33 WSA INFO EVENT

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	71	trlbdy33	wsa info msg log rec
52	(34)	CHARACTER	16	trlaid33	ad id for op
68	(44)	CHARACTER	10	trlia33	occ input arrival
68	(44)	CHARACTER	6	trliad33	mod input arr date
74	(4A)	CHARACTER	4	trliat33	mod input arr time
78	(4E)	CHARACTER	1	trlmsg33	type of message issued L late operation WALTE D long duration WALTE R resource cont. QQALR W WTO WAWTO U Urgent queue promotion N Urgent queue depromot P WLM promotion F WLM prom failure msg
79	(4F)	CHARACTER	44	trlnam33	resource name

TRL 34 CATALOG MGT EVENT

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	155	trlbdy34	Cat mgt event information
52	(34)	CHARACTER	8	trljnm34	Job name CATA
60	(3C)	CHARACTER	8	trljid34	Job id CATA
68	(44)	CHARACTER	8	trlnod34	node id CATA
76	(4C)	CHARACTER	16	trladi34	application id CATA
92	(5C)	CHARACTER	10	trliat34	input arrival CATA
102	(66)	CHARACTER	1	trlact34	Cat Mgt action CATA R = dataset recataloged CATA U = dataset uncataloged CATA D = dataset deleted CATA M = ds waiting for migrat CATA E = catalog return error CATA

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
103	(67)	CHARACTER	8	trlstp34	proc stepname or blank
111	(6F)	CHARACTER	8	trlddn34	dd name CATA
119	(77)	CHARACTER	44	trldsn34	dataset name CATA
163	(A3)	CHARACTER	44	trlcat34	catalog name, blank if master cat CATA

TRL 35 BACKUP EVENT

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
52	(34)	STRUCTURE	40	trlbdy35	backup event OBRA
52	(34)	CHARACTER	2	trlres35	resource dsn name OBRA
54	(36)	BITSTRING	1	trlfl135	flag byte 1
		1...		trldpa35	DP active, CP bkup ignored
		.111 1111		*	reserved
55	(37)	CHARACTER	1	*	reserved
56	(38)	CHARACTER	8	trlusr35	requesting user id OBRA
64	(40)	CHARACTER	8	trlmjs35	miscellaneous js values
64	(40)	SIGNED	4	trlopt35	max js size in k bytes
68	(44)	SIGNED	4	trlcnt35	num of inserts on js OBRA
72	(48)	CHARACTER	4	trlmcp35	miscellaneous cp values
72	(48)	SIGNED	2	trllim35	backup trk records limit
74	(4A)	SIGNED	2	trltot35	trk records since backup
76	(4C)	SIGNED	4	trlnode35	OPC node number OBRA
80	(50)	SIGNED	4	trlrdrn35	event reader num in node
84	(54)	CHARACTER	8	trlpos35	event ds position OBRA
84	(54)	SIGNED	4	trlcyc35	write cycle number OBRA
88	(58)	SIGNED	4	trlrec35	record number in cycle

TRL 36 CP BACKUP EVENT

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
52	(34)	STRUCTURE	30	trlbdy36	CP backup log record LOGA
52	(34)	CHARACTER	8	trlcpdd36	DDname of backed up ds
60	(3C)	CHARACTER	8	trljtdd36	DD of this JT dataset LOGA
68	(44)	CHARACTER	8	trljtddOCP36	DDname oldcp in CKPT at backup time
76	(4C)	CHARACTER	1	trl36filler	free
77	(4D)	CHARACTER	5	trljtwho36	who generate the trl

TRL 37 LOG DATA EVENT

Offsets Dec	 Hex	 Type	 Len	 Name (Dim)	 Description
52	(34)	STRUCTURE	*	trlbdy37	log data
52	(34)	CHARACTER	4	trltyp37	type of data logged
56	(38)	SIGNED	4	trlsize37	size of data logged
60	(3C)	CHARACTER	*	trldata37	data part

TRL 38 AR RESTEP INFO EVENT

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
52	(34)	STRUCTURE	48	trlbdy38	body of AR RESSTEP info BVZA
52	(34)	CHARACTER	16	trlaid38	ad id for op BVZA
68	(44)	CHARACTER	10	trlia38	occ input arrival BVZA
78	(4E)	CHARACTER	2	*	BVZA
80	(50)	CHARACTER	20	trlARinfo	BVZA
80	(50)	SIGNED	4	trlARstn	AR restart step number BVZA
84	(54)	CHARACTER	8	trlARstep	AR restart step name BVZA
92	(5C)	CHARACTER	8	trlARpStep	AR restart ProcStep name BVZA

TRL 39 BULKDISK EVENT

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
52	(34)	STRUCTURE	32	trlbdy39	bulkdisc event
52	(34)	BITSTRING	1	trlfl139	flag byte 1
		1...		trldpa39	DP active, CP bkup ignore
		.111 1111		*	reserved
53	(35)	CHARACTER	3	*	reserved
56	(38)	CHARACTER	8	trlusr39	requesting user id
64	(40)	CHARACTER	4	trlmcp39	miscellaneous cp values
64	(40)	SIGNED	2	trllim39	backup trk records limit
66	(42)	SIGNED	2	trltot39	trk records since backup
68	(44)	SIGNED	4	trlnode39	OPC node number
72	(48)	SIGNED	4	trlrdrn39	event reader num in node
76	(4C)	CHARACTER	8	trlpos39	event ds position
76	(4C)	SIGNED	4	trlcyc39	write cycle number
80	(50)	SIGNED	4	trlrec39	record number in cycle

TRL 40 CP66 EVENT

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
52	(34)	STRUCTURE	56	trlbdy40	body of CP66 event inf CRFA
52	(34)	CHARACTER	16	trladi40	aoccurrence number CRFA
68	(44)	CHARACTER	10	trlia40	CRFA
68	(44)	CHARACTER	6	trliad40	CRFA
74	(4A)	CHARACTER	4	trliat40	CRFA
78	(4E)	CHARACTER	6	trloid40	CRFA
78	(4E)	CHARACTER	4	trlwsn40	CRFA
82	(52)	SIGNED	2	trlopn40	CRFA
84	(54)	SIGNED	4	trlseqn40	CP66 sequence number CRFA
88	(58)	SIGNED	4	trlrnrec40	nbr of path recalc CRFA
92	(5C)	CHARACTER	1	trltrg40	path recalc trigger CRFA
93	(5D)	BITSTRING	1	trlflag40	flags CRFA
		1...		trlCPchgd40	critical path changed CRRA
		.111 1111		*	reserved CRFA
94	(5E)	CHARACTER	10	trlrec40	last recalculation CRFA
94	(5E)	CHARACTER	6	trlrec40D	date CRFA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
100	(64)	CHARACTER	4	trlrec40T	time CRFA
104	(68)	CHARACTER	4	trltype40	req type DELADDUPD CRFA

TRL 41 TZONE EXIT EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	40	trlbdy41	body of TZONE event CRPA
52	(34)	CHARACTER	16	trlaid41	ad id for op CRPA
68	(44)	CHARACTER	10	trlia41	occ input arrival CRPA
78	(4E)	SIGNED	2	trlTZOFF41	TZone offset CRPA
80	(50)	CHARACTER	12	*	free CRPA

TRL 42 ADDDEL OPERATION JTB EVENT

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	40	trlbdy42	body of event info CRFA
52	(34)	CHARACTER	16	trladi42	aoccurrence number CRFA
68	(44)	CHARACTER	10	trlia42	
68	(44)	CHARACTER	6	trliad42	CRFA
74	(4A)	CHARACTER	4	trliat42	CRFA
78	(4E)	CHARACTER	6	trloid42	CRFA
78	(4E)	CHARACTER	4	trlwsn42	CRFA
82	(52)	SIGNED	2	trlopn42	CRFA
84	(54)	BITSTRING	1	trlflags42	JTB flag CRFA
		1...		trlinJTB42	in JTBL CUTC
		.111 1111		*	free CRFA
85	(55)	CHARACTER	7	*	free CRFA

TRL 43 reserved for trl409 count

TRL 44 CHANGE STATUS OF SIMPLE COND MCP

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	98	trlbdy44	
52	(34)	CHARACTER	16	trladi44	Adid
68	(44)	CHARACTER	10	trlia44	IA
68	(44)	CHARACTER	6	trliad44	
74	(4A)	CHARACTER	4	trliat44	
78	(4E)	SIGNED	2	trlOpno44	opnum
80	(50)	SIGNED	2	trlCid44	Cond Id
82	(52)	SIGNED	2	trl44PredOp	Pred opno
84	(54)	CHARACTER	16	trl44PredAd	Pred adid
100	(64)	CHARACTER	10	trl44PredIA	Pred IA
110	(6E)	CHARACTER	1	trl44PredDep	Int Ext
111	(6F)	CHARACTER	2	trl44PredTyp	RC ST

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
113	(71)	CHARACTER	2	trl44PredLog	LT,LE,GE,GT...
115	(73)	CHARACTER	4	trl44PredValrc	rc value
119	(77)	CHARACTER	4	trl44PredValrc2	rc2 value
123	(7B)	CHARACTER	1	trl44PredValst	Status value
124	(7C)	CHARACTER	1	trlOldSt44	Simple old val
125	(7D)	CHARACTER	1	trlNewSt44	Simple new val
126	(7E)	CHARACTER	2	trlCaller44	Caller MCEM
128	(80)	CHARACTER	8	trlStepN44	Step name
136	(88)	CHARACTER	8	trlProcS44	Proc step
144	(90)	CHARACTER	6	*	free

TRL 45 CHANGE STATUS OF COMPLEX COND MCP

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	42	trlbdy45	
52	(34)	CHARACTER	16	trladi45	Adid
68	(44)	CHARACTER	10	trlia45	IA
68	(44)	CHARACTER	6	trliad45	
74	(4A)	CHARACTER	4	trliat45	
78	(4E)	SIGNED	2	trlOpno45	opnum
80	(50)	SIGNED	2	trlCid45	Cond Id
82	(52)	CHARACTER	1	trlOldSt45	old status
83	(53)	CHARACTER	1	trlNewSt45	new status
84	(54)	CHARACTER	2	trlCaller45	Caller MC
86	(56)	CHARACTER	8	*	free

TRL 46 AD cond dep table auto succ resol event

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	*	trlbdy46	body of ad cond table event
52	(34)	CHARACTER	*	trlcnd46	mapped by dclconrc

TRL 47 missing A3S event in cond evaluation event

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	102	trlbdy47	missing event
52	(34)	CHARACTER	16	trladi47	Adid
68	(44)	CHARACTER	10	trlia47	IA
68	(44)	CHARACTER	6	trliad47	date
74	(4A)	CHARACTER	4	trliat47	time
78	(4E)	SIGNED	2	trlOpno47	opnum
80	(50)	SIGNED	2	trlCid47	Cond Id
82	(52)	SIGNED	2	trl47PredOp	Pred opno
84	(54)	CHARACTER	16	trl47PredAd	Pred adid
100	(64)	CHARACTER	10	trl47PredIA	Pred IA
110	(6E)	CHARACTER	1	trl47PredDep	Int Ext

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
111	(6F)	CHARACTER	2	trl47PredTyp	RC ST
113	(71)	CHARACTER	2	trl47PredLog	LT,LE,GE,GT...
115	(73)	CHARACTER	4	trl47PredValrc	rc value
119	(77)	CHARACTER	4	trl47PredValrc2	rc2 value
123	(7B)	CHARACTER	1	trl47PredValst	Status value
124	(7C)	CHARACTER	2	trlCaller47	Caller MCEM
126	(7E)	CHARACTER	8	trlStepN47	Step name
134	(86)	CHARACTER	8	trlProcS47	Proc Step
142	(8E)	CHARACTER	1	trlMiss47	Missing event
143	(8F)	CHARACTER	8	trlJobNm47	jobname
151	(97)	CHARACTER	3	*	free

TRL 48 recovered by cond event only for monitoring

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	40	trlbdy48	
52	(34)	CHARACTER	16	trladi48	
68	(44)	CHARACTER	10	trlia48	
68	(44)	CHARACTER	6	trliad48	
74	(4A)	CHARACTER	4	trliat48	
78	(4E)	CHARACTER	6	trloid48	
78	(4E)	CHARACTER	4	trlwsn48	
82	(52)	SIGNED	2	trlopn48	
84	(54)	CHARACTER	3	trlcaller48	
87	(57)	CHARACTER	5	*	

TRL 49 A3S match in cond event missno

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	102	trlbdy49	missing event
52	(34)	CHARACTER	16	trladi49	Adid
68	(44)	CHARACTER	10	trlia49	IA
68	(44)	CHARACTER	6	trliad49	date
74	(4A)	CHARACTER	4	trliat49	time
78	(4E)	SIGNED	2	trlOpno49	opnum
80	(50)	SIGNED	2	trlCid49	Cond Id
82	(52)	SIGNED	2	trl49PredOp	Pred opno
84	(54)	CHARACTER	16	trl49PredAd	Pred adid
100	(64)	CHARACTER	10	trl49PredIA	Pred IA
110	(6E)	CHARACTER	1	trl49PredDep	Int Ext
111	(6F)	CHARACTER	2	trl49PredTyp	RC ST
113	(71)	CHARACTER	2	trl49PredLog	LT,LE,GE,GT...
115	(73)	CHARACTER	4	trl49PredValrc	rc value
119	(77)	CHARACTER	4	trl49PredValrc2	rc2 value
123	(7B)	CHARACTER	1	trl49PredValst	Status value
124	(7C)	CHARACTER	2	trlCaller49	Caller MCEM
126	(7E)	CHARACTER	8	trlStepN49	Step name
134	(86)	CHARACTER	8	trlProcS49	Proc Step

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
142	(8E)	CHARACTER	1	trlMissNO49	MissNO event
143	(8F)	CHARACTER	8	trlJobNm49	jobname
151	(97)	CHARACTER	3	*	free

TRL 50 unexpected RC

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	73	trlbdy50	
52	(34)	CHARACTER	16	trladi50	
68	(44)	CHARACTER	10	trlia50	
68	(44)	CHARACTER	6	trliad50	
74	(4A)	CHARACTER	4	trliat50	
78	(4E)	CHARACTER	6	trloid50	
78	(4E)	CHARACTER	4	trlwsn50	
82	(52)	SIGNED	2	trlopn50	
84	(54)	CHARACTER	3	trlcaller50	
87	(57)	CHARACTER	1	trlType50	Y N
88	(58)	CHARACTER	8	trlJobname50	jobname
96	(60)	CHARACTER	8	trlJobid50	jobid
104	(68)	CHARACTER	8	trlSTEP50	step name
112	(70)	CHARACTER	8	trlPSTEP50	proc step name
120	(78)	CHARACTER	1	trlStepEnd50	step end event Y N
121	(79)	CHARACTER	1	trlStatus50	oper status
122	(7A)	CHARACTER	3	*	

TRL 51 Audit trail record access EQQJBLIB member

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	*	TRLBDY51A	joblib jcl log record
52	(34)	CHARACTER	76	TRLBDY51	record data
52	(34)	UNSIGNED	4	TRLKLN51	size of key
56	(38)	UNSIGNED	4	TRLREC51	amount of data this re
60	(3C)	CHARACTER	4	TRLFIL51	logical file name
64	(40)	CHARACTER	1	TRLACC51	access type(r,u,i,d)
65	(41)	BITSTRING	3	TRLRSV51	filler
68	(44)	CHARACTER	52	TRLKEY51	job library
68	(44)	CHARACTER	44	TRLLIB51	job library
112	(70)	CHARACTER	8	TRLMEM51	jcl member
120	(78)	CHARACTER	8	*	reserved
128	(80)	CHARACTER	*	TRLDAT51	record data

TRL 52 LOG XD02 VSAM RECORD CHANGES INVOLVING LTP OR NMM

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	*	trlbdy52	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	SIGNED	4	trlsize52	xd02len
56	(38)	CHARACTER	*	trldata52	xd02 record

TRL 53 NOERROR match occurred for operation

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	92	trlbdy53	DEWA
52	(34)	CHARACTER	16	trladi53	DEWA
68	(44)	CHARACTER	10	trlia53	DEWA
68	(44)	CHARACTER	6	trliad53	DEWA
74	(4A)	CHARACTER	4	trliat53	DEWA
78	(4E)	SIGNED	2	trlopnum53	DEWA
80	(50)	CHARACTER	4	trlws53	DEWA
84	(54)	CHARACTER	8	trljob53	DEWA
92	(5C)	CHARACTER	8	trljid53	DEWA
100	(64)	CHARACTER	41	trlstmt53	DEWA
141	(8D)	CHARACTER	3	*	DEWA

TRL 54 NOERROR proc skip for operation $DJRA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
52	(34)	STRUCTURE	34	trlbdy54	DJRA
52	(34)	CHARACTER	16	trladi54	DJRA
68	(44)	CHARACTER	10	trlia54	DJRA
68	(44)	CHARACTER	6	trliad54	DJRA
74	(4A)	CHARACTER	4	trliat54	DJRA
78	(4E)	SIGNED	2	trlopnum54	DJRA
80	(50)	CHARACTER	4	trlws54	DJRA
84	(54)	1...		trlNESkip54	DJRA
		.111 1111		*	DJRA
85	(55)	CHARACTER	1	*	DJRA

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	12	trlWRK	
0	(0)	ADDRESS	4	trlWRKnxt	
4	(4)	CHARACTER	2	trlWRKtyp	
6	(6)	CHARACTER	2	*	
8	(8)	ADDRESS	4	trlWRKelP	

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	12	trlMCP	
0	(0)	ADDRESS	4	trlMCPnxt	
4	(4)	CHARACTER	2	trlMCPtyp	
6	(6)	CHARACTER	2	*	
8	(8)	ADDRESS	4	trlMCPelP	

Cross reference

Name	Hex Offset	Hex Value	Level
trlopdl28	5C		2
trlopdu28	5E		2
trlop28	60		2
trlacc32	44		2
TRLACC51	40		3
trlact34	66		2
trladi23	34		2
trladi34	4C		2
trladi40	34		2
trladi42	34		2
trladi44	34		2
trladi45	34		2
trladi47	34		2
trladi48	34		2
trladi49	34		2
trladi50	34		2
trladi53	34		2
trladi54	34		2
trladld27	68		3
trladld28	54		3
trladlt27	6E		3
trladlt28	56		3
trladl27	68		2
trladl28	54		2
trladu27	58		2
trladu28	6C		3
trlaid25	34		2
trlaid26	34		2
trlaid27	34		2
trlaid28	34		2
trlaid33	34		2
trlaid38	34		2
trlaid41	34		2
trlaltws25	80		2
trlARinfo	50		2
trlARpStep	5C		3
trlARstep	54		3
trlARstn	50		3
trlbdw	2		3

Name	Hex Offset	Hex Value	Level
trlbdy20	34		1
trlbdy23	34		1
trlbdy24	34		1
trlbdy25	34		1
trlbdy26	34		1
trlbdy27	34		1
trlbdy28	34		1
trlbdy29	34		1
trlbdy30	34		1
trlbdy31	34		1
trlbdy32	34		1
trlbdy33	34		1
trlbdy34	34		1
trlbdy35	34		1
trlbdy36	34		1
trlbdy37	34		1
trlbdy38	34		1
trlbdy39	34		1
trlbdy40	34		1
trlbdy41	34		1
trlbdy42	34		1
trlbdy44	34		1
trlbdy45	34		1
trlbdy46	34		1
trlbdy47	34		1
trlbdy48	34		1
trlbdy49	34		1
trlbdy50	34		1
TRLBDY51	34		2
TRLBDY51A	34		1
trlbdy52	34		1
trlbdy53	34		1
trlbdy54	34		1
trlbyp23	5F	40	3
trlCaller44	7E		2
trlCaller45	54		2
trlCaller47	7C		2
trlcaller48	54		2
trlCaller49	7C		2
trlcaller50	54		2
trlcat34	A3		2

Name	Hex Offset	Hex Value	Level
trlchg30	36	40	3
trlchg32	74		3
trlCid44	50		2
trlCid45	50		2
trlCid47	50		2
trlCid49	50		2
trlckptr	30	80	4
trlcnd46	34		2
trlcnt35	44		3
trlcom	0		2
trlCPchgd40	5D	80	3
trlcpdd36	34		2
trlcpeut	14		5
trlcpkey	14		3
trlcyc30	40		3
trlcyc35	54		3
trlcyc39	4C		3
trldata37	3C		2
trldata52	38		2
trldat32	88		2
TRLDAT51	80		2
trlddn25	56		2
trlddn34	6F		2
trldpa35	36	80	3
trldpa39	34	80	3
trldsn34	77		2
trldurs23	78		2
trldur23	55		2
trledld27	62		3
trledlt27	64		3
trledl27	62		2
trledu27	54		2
trlerc23	5B		2
trletc31	36		2
trlevdat	C		3
trlevtim	10		3
trlevt23	54		2
trlexit13rc	6E	20	3
trlexr29	35		2
trlexsr30	48		2
trlexs30	38		1

Name	Hex Offset	Hex Value	Level
trlfdb23	5F	80	3
trlfill	27		3
trlfill2	30		3
trlfil32	40		2
TRLFIL51	3C		3
trlfirst	27	80	4
trlFlag	6E		2
trlflags42	54		2
trlflag40	5D		2
trlfla30	36		2
trlfl135	36		2
trlfl139	34		2
trlgmtof	2C		3
trliad23	44		2
trliad25	44		3
trliad27	44		2
trliad28	44		2
trliad33	44		3
trliad40	44		3
trliad42	44		3
trliad44	44		3
trliad45	44		3
trliad47	44		3
trliad48	44		3
trliad49	44		3
trliad50	44		3
trliad53	44		3
trliad54	44		3
trliat23	4A		2
trliat25	4A		3
trliat27	4A		2
trliat28	4A		2
trliat33	4A		3
trliat34	5C		2
trliat40	4A		3
trliat42	4A		3
trliat44	4A		3
trliat45	4A		3
trliat47	4A		3
trliat48	4A		3
trliat49	4A		3

Name	Hex Offset	Hex Value	Level
trliat50	4A		3
trliat53	4A		3
trliat54	4A		3
trlia25	44		2
trlia26	44		2
trlia33	44		2
trlia38	44		2
trlia40	44		2
trlia41	44		2
trlia42	44		2
trlia44	44		2
trlia45	44		2
trlia47	44		2
trlia48	44		2
trlia49	44		2
trlia50	44		2
trlia53	44		2
trlia54	44		2
trlinJTB42	54	80	3
trljbn23	70		2
trljid25	4E		2
trljid34	3C		2
trljid53	5C		2
trljnm25	5E		2
trljnm34	34		2
trlJobid50	60		2
trlJobname50	58		2
trlJobNm47	8F		2
trlJobNm49	8F		2
trljob53	54		2
trljrnfl	27	01	4
trljrpfl	27	04	4
trljtddOCP36	44		2
trljtdd36	3C		2
trljtwho36	4D		2
trlkey32	48		2
TRLKEY51	44		3
trlkln32	34		2
TRLKLN51	34		3
trllast	27	40	4
trllengt	28		3

Name	Hex Offset	Hex Value	Level
trllib32	64		3
TRLLIB51	44		4
trllim35	48		3
trllim39	40		3
trllogid	7		3
trlman23	5F	20	3
trlMCP	0		1
trlMCPelP	8		2
trlMCPnxt	0		2
trlMCPtyp	4		2
trlmcp35	48		2
trlmcp39	40		2
trlmem32	6C		3
TRLMEM51	70		4
trlMissNO49	8E		2
trlMiss47	8E		2
trlmjs35	40		2
trlmod20	34		2
trlmsg33	4E		2
trlmt024	34		2
trlnam33	4F		2
trlNESkip54	54	80	2
trlNewSt44	7D		2
trlNewSt45	53		2
trlnode30	38		2
trlnode35	4C		2
trlnode39	44		2
trlnod34	44		2
trlodld28	4E		3
trlodlt28	50		3
trlodl28	4E		2
trlodu28	68		3
trloid23	4E		2
trloid27	4E		2
trloid40	4E		2
trloid42	4E		2
trloid48	4E		2
trloid50	4E		2
trloldcp	27	02	4
trlOldSt44	7C		2
trlOldSt45	52		2

Name	Hex Offset	Hex Value	Level
trlopadld28	76		4
trlopadlt28	78		4
trlopadl28	76		3
trlopc	4		3
trlopi28	62		3
trlOpno44	4E		2
trlOpno45	4E		2
trlOpno47	4E		2
trlOpno49	4E		2
trlopnum53	4E		2
trlopnum54	4E		2
trlopn23	52		3
trlopn40	52		3
trlopn42	52		3
trlopn48	52		3
trlopn50	52		3
trlopodld28	70		4
trlopodlt28	72		4
trlopodl28	70		3
trlops28	62		2
trlopt35	40		3
trlorig27	5F		2
trlowi23	60		2
trlpos30	40		2
trlpos35	54		2
trlpos39	4C		2
trlProcS44	88		2
trlProcS47	86		2
trlProcS49	86		2
trlPSTEP50	70		2
trlrcbdy	34		2
trlrctyp	9		3
trlrcver	B		3
trlrdrn30	3C		2
trlrdrn35	50		2
trlrdrn39	48		2
trlready23	7C		2
trlrec	0		1
trlrec30	44		3
trlrec35	58		3
trlrec39	50		3

Name	Hex Offset	Hex Value	Level
trlrec32	36		2
trlrec40	5E		2
trlrec40D	5E		3
trlrec40T	64		3
TRLREC51	38		3
trlrem32	3C		2
trlreq30	35		2
trlreq31	35		2
trlrer25	6E	40	3
trlresub	27	08	4
trlres27	5E		2
trlres35	34		2
trlrid27	60		2
trlrid28	5A		2
trlrnrec40	58		2
trlrsr30	38		2
trlrsv32	45		2
TRLRSV51	41		3
trlsa	30	40	4
trlseqn40	54		2
trlsize	0		3
trlsize37	38		2
trlsize52	34		2
trlsrs30	36	80	3
trlsseq	22		4
trlStatus50	79		2
trlstc	27	20	4
trlStepEnd50	78		2
trlStepN44	80		2
trlStepN47	7E		2
trlStepN49	7E		2
trlSTEP50	68		2
trlstmt53	64		2
trlstp34	67		2
trlsubtoken25	78		2
trlTaskType25	6F		2
trltok_full25	70		2
trltok_pre25	70		3
trltok25	74		3
trltot32	38		2
trltot35	4A		3

Name	Hex Offset	Hex Value	Level
trltot39	42		3
trltrg40	5C		2
trltype40	68		2
trlType50	57		2
trltyp29	34		2
trltyp30	34		2
trltyp31	34		2
trltyp37	34		2
trlTZOFF41	4E		2
trluser	1A		4
trlusr25	66		2
trlusr35	38		2
trlusr39	38		2
trlWASUJ	6E	80	3
trlWRK	0		1
trlWRKelP	8		2
trlWRKnxt	0		2
trlWRKtyp	4		2
trlwsn23	4E		3
trlwsn40	4E		3
trlwsn42	4E		3
trlwsn48	4E		3
trlwsn50	4E		3
trlws53	50		2
trlws54	50		2
trlwto	27	10	4
trlxst26	4E		2
trlX23	5F	10	3
trl36filler	4C		2
trl44PredAd	54		2
trl44PredDep	6E		2
trl44PredIA	64		2
trl44PredLog	71		2
trl44PredOp	52		2
trl44PredTyp	6F		2
trl44PredValrc	73		2
trl44PredValrc2	77		2
trl44PredValst	7B		2
trl47PredAd	54		2
trl47PredDep	6E		2
trl47PredIA	64		2

Name	Hex Offset	Hex Value	Level
trl47PredLog	71		2
trl47PredOp	52		2
trl47PredTyp	6F		2
trl47PredValrc	73		2
trl47PredValrc2	77		2
trl47PredValst	7B		2
trl49PredAd	54		2
trl49PredDep	6E		2
trl49PredIA	64		2
trl49PredLog	71		2
trl49PredOp	52		2
trl49PredTyp	6F		2
trl49PredValrc	73		2
trl49PredValrc2	77		2
trl49PredValst	7B		2

WAP - Workstation analyzer parameters

Name : DCLWAP

Function:
This segment declares a workstation analyzer area. This control block is built, initialized, and freed by the workstation analyzer task.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	344	WAP	WS ANALYZER PARAMETERS
0	(0)	CHARACTER	4	WAPDESC	BLOCK DESCRIPTOR = WAP
4	(4)	CHARACTER	2	WAPVER	BLOCK MAPPING VERSION
6	(6)	BITSTRING	2	WAPFLAGS	FLAGS
		1...		WAPNOWT	NO WAIT THIS TIME
		.1..		WAPSUBX	CALL SUBUS EXIT
		..1.		WAPUX01	UX02 RC NOT 0
		...1		WAPSUBER	ERROR IN SUBMISSION OF A CENTRALIZED SCRIPT
		.... 1...		WAPSAERR	ERROR RETURNED FROM SA EXIT
		.... .1..		WAPFAKESUB	FAKE SUBMISSION FOR PARALL
8	(8)	BITSTRING	4	WAPTMECB	TIMER ECB
12	(C)	ADDRESS	4	WAPMCAP	MCA ADDRESS
16	(10)	ADDRESS	4	WAPDOAP	DOA ADDRESS
20	(14)	ADDRESS	4	WAPJBUF	JOBLIB BUFFER ADDRESS
24	(18)	ADDRESS	4	WAPEMPP	EV MGR PARAMETER AREA
28	(1C)	ADDRESS	4	WAPOPRP	CURRENT OPER RECORD ADDR
32	(20)	ADDRESS	4	WAPJCLB	JCL RECORD BUFFER, < 16M
36	(24)	SIGNED	4	WAPJSIZE	MAXIMUM JS RECORD SIZE
40	(28)	ADDRESS	4	WAPSSRP	SR AREA TO SUBUS
44	(2C)	CHARACTER	8	WAPRUSER	CURRENT TSO USER
52	(34)	CHARACTER	16	WAPJID	JOB IDENTIFICATION

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
52	(34)	CHARACTER	8	WAPJNAM	JOB NAME
60	(3C)	CHARACTER	8	WAPJNUM	JES JOB NUMBER
68	(44)	CHARACTER	10	WAPLATE	NEXT TEST LATE OPER
78	(4E)	CHARACTER	10	WAPDWTO	NEXT DEADLINE WTO TIME
88	(58)	CHARACTER	1	WAPSSTAT	VAR SUBSTITUTION STATUS
89	(59)	CHARACTER	3	*	RESERVED
92	(5C)	CHARACTER	88	WAPDCB	JOBLIB DCB
180	(B4)	CHARACTER	64	WAPEXIT	TIMER EXIT
244	(F4)	CHARACTER	10	WAPSYNC	NEXT TIME FOR NEW SYNC
254	(FE)	SIGNED	2	WAPPLOOP#	LOOP COUNTER
256	(100)	ADDRESS	4	WAPPDOAP	PREVIOUS DOA ADDRESS
260	(104)	CHARACTER	4	WAPPCPPTR	PREVIOUS CP INDEX
260	(104)	UNSIGNED	3	WAPOCCX	OCCURRENCE INDEX
263	(107)	UNSIGNED	1	WAPOPNO	OPERATION NUMBER
264	(108)	CHARACTER	4	WAPUX01CO	USER EXIT 02 RC
268	(10C)	CHARACTER	48	WAPPERFDB	PERF DEBUG INFO
268	(10C)	CHARACTER	8	WAPWASUB	WASUB ELAPSED TIME
276	(114)	CHARACTER	8	WAPJHGET	JHGET ELAPSED TIME
284	(11C)	CHARACTER	8	WAPJHPDS	JHPDS ELAPSED TIME
292	(124)	CHARACTER	8	WAPEXI02	EXI02 ELAPSED TIME
300	(12C)	CHARACTER	8	WAPJHPUT	JHPUT ELAPSED TIME
308	(134)	CHARACTER	8	WAPEXI01	EXI01 ELAPSED TIME
316	(13C)	1...		WAPDEBUG	PERF DEBUG ON
		.111 1111		*	RESERVED
317	(13D)	CHARACTER	3	*	RESERVED
320	(140)	ADDRESS	4	WAPDIAP	ADDRESS OF ZZWSA
324	(144)	CHARACTER	20	*	RESERVED
340	(154)	CHARACTER	4	*	RESERVED
344	(158)	CHARACTER		WAPEND	END OF WAP

Cross reference

Name	Hex Offset	Hex Value	Level
WAP	0		1
WAPDCB	5C		2
WAPDEBUG	13C	80	2
WAPDESC	0		2
WAPDIAP	140		2
WAPDOAP	10		2
WAPDWTO	4E		2
WAPEMPP	18		2
WAPEND	158		2
WAPEXIT	B4		2
WAPEXI01	134		3
WAPEXI02	124		3
WAPFAKESUB	6	04	3
WAPFLAGS	6		2

Name	Hex Offset	Hex Value	Level
WAPJBUF	14		2
WAPJCLB	20		2
WAPJHGET	114		3
WAPJHPDS	11C		3
WAPJHPUT	12C		3
WAPJID	34		2
WAPJNAM	34		3
WAPJNUM	3C		3
WAPJSIZE	24		2
WAPLATE	44		2
WAPMCAP	C		2
WAPNOWT	6	80	3
WAPOCCX	104		3
WAPOPNO	107		3
WAPOPRP	1C		2
WAPPCPPTR	104		2
WAPPDOAP	100		2
WAPPERFDB	10C		2
WAPPLOOP#	FE		2
WAPRUSER	2C		2
WAPSAERR	6	08	3
WAPSCHE	144		2
WAPSSRP	28		2
WAPSSTAT	58		2
WAPSUBER	6	10	3
WAPSUBX	6	40	3
WAPSYNC	F4		2
WAPTMECB	8		2
WAPUX01	6	20	3
WAPUX01CO	108		2
WAPVER	4		2
WAPWASUX	10C		3

WQA - Workstation queue area

Name : DCLWQA

Function:
This segment declares a workstation queue area entry. WQAs are chained, via WQAWQANX, to the anchor in NMMWAQ1. WQAs are built by the NMM when the scheduler is started from information in current plan workstation records.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	248	WQA	WORKSTATION QUEUE AREA
0	(0)	CHARACTER	4	WQADESC	BLOCK DESCRIPTOR = 'WQA '
4	(4)	CHARACTER	2	WQAVER	BLOCK MAPPING VERSION
6	(6)	BITSTRING	1	WQAFLGS	WORKSTATION FLAGS
		1...		WQADEL	WQA IS LOGICALLY DELETED
		.1..		WQAIVLNP	PS NOT USED
		..1.		WQAR1C	CONTROL ON RESOURCE 1
		...1		WQAR2C	CONTROL ON RESOURCE 2
		.... 1...		WQAREROUTE	1: REROUTE MODE
		.... .1..		WQAWS_VARY	1: VARIED STATUS
		.... ..1.		WQARTR_PEND	1: RTR EVENT PENDING
		1		WQAOFF_PEND	1: PENDING OFFLINE
7	(7)	CHARACTER	1	WQAWSTYP	WORKSTATION TYPE
8	(8)	CHARACTER	4	WQAWSNAM	WORKSTATION NAME
12	(C)	CHARACTER	1	WQAWSREP	REPORTING ATTRIBUTE
13	(D)	CHARACTER	1	WQAPREP	JOB PREPARATION WS Y|N
14	(E)	CHARACTER	1	WQASTATUS	WS STATUS
15	(F)	BITSTRING	1	WQAFLAG2	WORKSTATION FLAG 2
		1...		WQASTC	STC OPTION
		.1..		WQAWTO	WTO OPTION
		..1.		WQASSEQ_PEND	WAITING FOR J0 EVENT
		...1		WQAOSI	USER DEFINED DESTINATION
		.... 1...		WQATOKEN	TOKEN REQUIRED
		.... .1..		WQAWSSTAT	STATUS BY WSSTAT
		.... ..1.		WQAUX009	STATUS BY EQQUX009
		1		WQATCP	TCP DESTINATION
16	(10)	UNSIGNED	2	WQACURR1	AMOUNT OF R1 IN USE
18	(12)	UNSIGNED	2	WQACURR2	AMOUNT OF R2 IN USE
20	(14)	SIGNED	4	WQAMAX15	MAX NO OF EVENTS IN 15 MIN
24	(18)	BITSTRING	2	WQAFLAG3	WORKSTATION FLAG 3
		1...		WQAAPPC	APPC DESTINATION
		.1..		WQAWAIT	WAIT WORKSTATION
		..1.		WQAZNOINFO	ZCENTRIC INIT STATUS
		...1		WQASTCHG	WQA STATUS CHANGE
24	(18)	BITSTRING	1	*	FREE
26	(1A)	BITSTRING	2	*	FREE
		1...		WQATCPNEW	TCPIP CLASSIC TRACKER
28	(1C)	ADDRESS	4		
32	(20)	ADDRESS	4	WQADOAP	ADDR OF 1ST ACTIVE OPER
36	(24)	ADDRESS	4	WQASUREP	ADDR OF SU/RE DS IOC
40	(28)	ADDRESS	4	WQAOINTP	CURR INTERVAL IVL
44	(2C)	ADDRESS	4	WQA_ALTWSP	ADDR OF ALTERNATE
48	(30)	ADDRESS	4	WQADOAPR	ADDR OF 1ST READY OPER
52	(34)	CHARACTER	12	WQASUMC	SUM OF COMPLETED OPS

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
52	(34)	SIGNED	4	WQASUMC#	NUMBER
56	(38)	SIGNED	4	WQASUMCE	ESTIMATED DURATION
60	(3C)	SIGNED	4	WQASUMCR	REAL DURATION
64	(40)	CHARACTER	12	WQASUMI	SUM OF INTERRUPTED OPS
64	(40)	SIGNED	4	WQASUMI#	NUMBER
68	(44)	SIGNED	4	WQASUMIE	ESTIMATED DURATION
72	(48)	SIGNED	4	WQASUMIR	REAL DURATION
76	(4C)	CHARACTER	8	WQASUMS	SUM OF STARTED OPS
76	(4C)	SIGNED	4	WQASUMS#	NUMBER
80	(50)	CHARACTER	8	WQASUMSE	ESTIMATED DURATION
84	(54)	CHARACTER	8	WQASUMR	SUM OF READY OPS
84	(54)	SIGNED	4	WQASUMR#	NUMBER
88	(58)	CHARACTER	8	WQASUMRE	ESTIMATED DURATION
92	(5C)	CHARACTER	8	WQASUMW	SUM OF WAITING OPS
92	(5C)	SIGNED	4	WQASUMW#	NUMBER
96	(60)	SIGNED	4	WQASUMWE	ESTIMATED DURATION
100	(64)	SIGNED	4	WQASUMA	SUM OF ARRIVING OPS
104	(68)	SIGNED	4	WQASUMNRR	SUM OF '*' READY
108	(6C)	SIGNED	4	WQASUMU	SUM OF UNDECIDED
112	(70)	SIGNED	4	WQASUME	SUM OF ERROR OPERS
116	(74)	SIGNED	4	WQASUML	SUM OF LATE OPERS
120	(78)	CHARACTER	8	WQACREVT	LAST SYNCHRONOUS EVENT CREATION TIME
120	(78)	SIGNED	4	WQADATE	DATE FORMAT (00YYDDDF)
124	(7C)	SIGNED	4	WQATIME	TIME FORMAT (SECS*100)
128	(80)	ADDRESS	4	WQAVIRTPTR	IF WQAVTYPE: N ==> 0, V ==> 1° VIRT DEST WQA, D ==> OWNING VIRT WQA
132	(84)	SIGNED	4	WQASUMX	SUM OF DUMMY COMPL OPERATIONS
136	(88)	SIGNED	4	WQASUMD	SUM OF DELETED OPERATIONS
140	(8C)	CHARACTER	8	WQAPREEV	TIME OF PREVIOUS EVENT
140	(8C)	CHARACTER	4	WQAPREDT	DATE (FORMAT 00YYDDDF)
144	(90)	CHARACTER	4	WQAPRETM	TIME (FORMAT HHMMSSTH)
148	(94)	CHARACTER	8	WQAOFF	TIME OF OFFLINE EVENT
148	(94)	CHARACTER	4	WQAOFF_DATE	DATE (FORMAT 00YYDDDF)
152	(98)	CHARACTER	4	WQAOFF_TIME	TIME (FORMAT HHMMSSTH)
156	(9C)	CHARACTER	2	WQAWSR1N	NAME OF R1 ON THIS WORKSTATION
158	(9E)	CHARACTER	2	WQAWSR2N	NAME OF R2 ON THIS WORKSTATION
160	(A0)	CHARACTER	8	WQASUDS	SUBMIT DESTINATION
168	(A8)	SIGNED	4	WQAIVL#	NUMBER OF OPEN INTERVALS
172	(AC)	CHARACTER	32	WQASDESC	WORKSTATION DESCRIPTION
204	(CC)	CHARACTER	1	WQAEOPTP	POINTER TO E2E OPTIONS
208	(D0)	CHARACTE	1	WQAVTYPE	
209	(D1)	UNSIGNED	1	WQARSEQ#	CURRENT SUBMIT REQ#
210	(D2)	UNSIGNED	2	WQASSEQ#	CURRENT SUBMIT SEQ#
212	(D4)	CHARACTER	8	WQANETID	APPC ROUTE NETID
220	(DC)	CHARACTER	8	WQANETLU	APPC ROUTE LU NAME
228	(E4)	UNSIGNED	1	WQASUMCES	SECONDS FOR WQASUMCE
229	(E5)	UNSIGNED	1	WQASUMCRS	SECONDS FOR WQASUMCR
230	(E6)	UNSIGNED	1	WQASUMIES	SECONDS FOR WQASUMIE
231	(E7)	UNSIGNED	1	WQASUMIRS	SECONDS FOR WQASUMIR

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
232	(E8)	UNSIGNED	1	WQASUMSES	SECONDS FOR WQASUMSE
233	(E9)	UNSIGNED	1	WQASUMRES	SECONDS FOR WQASUMRE
234	(EA)	UNSIGNED	1	WQASUMWES	SECONDS FOR WQASUMWE
235	(EB)	CHARACTER	1	WQATWS	TIVOLI WORKLOAD SCHEDULER WORKSTATION
236	(EC)	CHARACTER	1	WQALNKSTA	TIVOLI WORKLOAD SCHEDULER WORKSTATION LINK STATUS
237	(ED)	CHARACTER	1	WQAFLGTWS	TIVOLI WORKLOAD SCHEDULER WORKSTATION SPECIFIC FLAGS
		1...		WQALNKVA	TIVOLI WORKLOAD SCHEDULER LINKED MAN VARIED
		.1..		WQASTAVA	TIVOLI WORKLOAD SCHEDULER STATUS MAN VARIED
		..1.		WQALNKDW	TIVOLI WORKLOAD SCHEDULER LINKED DOWN TO CPU
		...1		WQAWRTST	TIVOLI WORKLOAD SCHEDULER WRITER STATUS FOR CPU
		.... 1...		WQACMDLNK	TIVOLI WORKLOAD SCHEDULER command link: on=link off=unlink
		.... .1..		WQACMDSTA	TIVOLI WORKLOAD SCHEDULER command status: on=start off=stop
		.... ..1.		WQAFULLNK	TIVOLI WORKLOAD FULLY LINKED STATUS: ON=FULLY LINKED OFF=NOT FULLY LINKED
238	(EE)	USIGNED	2	WQACPULIMIT	CPU LIMIT VALUE
240	(F0)	CHARACTER	1	WQAWSFLG	WORKSTATION FLAG
		1...		WQASAUTO	WORKSTATION SYSTEM AUTOMATION STATUS: YES=ENABLED NO=DISABLED
		.1..		WQASTARTV	ON: LAST PROPOSED VIRT
		..1.		WQASUBV	ON: LAST SUB FOR VIRT
		...1		WQAZCAGENT	ON: Z CENTRIC DISTRIBUTED AGENT
		.... 1...		WQAPARALWS	ON: PARALLEL WORKSTATION
		.... .1..		WQADYN	ON: DYNAMIC WORKSTATION
		.... ..1.		*	AVAILABLE
241	(F1)	CHARACTER	1	WQAFULLYACT	Y/N
242	(F2)	CHARACTER	1	WQARENGT	REMOTE ENGINE TYPE
243	(F3)	CHARACTER	1	*	FREE
244	(F4)	SIGNED	4	WQASUMH	SUM OF MAN HOLD OPERATIONS
248	(F8)	CHARACTER		WQAEND	END OF WQA

Cross reference

Name	Hex Offset	Hex Value	Level
WQA	0		1
WQA_ALTWSP	2C		2
WQAACCMP	CC		2
WQAAPPC	18	80	3
WQACMDLNK	ED	08	3
WQACMDSTA	ED	04	3
WQACPULIMIT	EE		2
WQACREVT	78		2
WQACURR1	10		2
WQACURR2	12		2
WQADATE	78		3
WQADEL	6	80	3
WQADESC	0		2
WQADOAP	20		2
WQADOAPR	30		2
WQADYN	F0	04	3
WQAEND	F8		2
WQAEOPTP	CC		2
WQAFLAG2	F		2
WQAFLAG3	18		2
WQAFLGS	6		2
WQAFLGTWS	ED		2
WQAFULLNK	ED	02	3
WQAFULLYACT	F1		2
WQAIVL#	A8		2
WQAIVLNP	6	40	3
WQALNKDW	ED	20	3
WQALNKSTA	EC		2
WQALNKVA	ED	80	3
WQAMAX15	14		2
WQANETID	D4		2
WQANETLU	DC		2
WQAOFF	94		2
WQAOFF_DATE	94		3
WQAOFF_PEND	6	01	3
WQAOFF_TIME	98		3
WQAOINTP	28		2
WQAOSI	F	10	3
WQAPARALWS	F0	08	3

Name	Hex Offset	Hex Value	Level
WQAPREDT	8C		3
WQAPREEV	8C		2
WQAPREP	D		2
WQAPRETM	90		3
WQARENGT	F2		2
WQAREROUTE	6	08	3
WQARSEQ#	D1		2
WQARTR_PEND	6	02	3
WQAR1C	6	20	3
WQAR2C	6	10	3
WQASAUTO	F0	80	3
WQASDESC	AC		3
WQASSEQ_PEND	F	20	3
WQASSEQ#	D2		2
WQASTARTV	F0	40	3
WQASTATUS	E		2
WQASTAVA	ED	40	3
WQASTC	F	80	3
WQASTCHG	18	10	3
WQASUBV	F0	20	3
WQASUDS	A0		2
WQASUMA	64		2
WQASUMC	34		2
WQASUMC#	34		3
WQASUMCE	38		3
WQASUMCES	E4		2
WQASUMCR	3C		3
WQASUMCRS	E5		2
WQASUMD	88		2
WQASUME	70		2
WQASUMH	F4		2
WQASUMI	40		2
WQASUMI#	40		3
WQASUMIE	44		3
WQASUMIES	E6		2
WQASUMIR	48		3
WQASUMIRS	E7		2
WQASUML	74		2
WQASUMNRR	68		2
WQASUMR	54		2
WQASUMR#	54		3

Name	Hex Offset	Hex Value	Level
WQASUMRE	58		3
WQASUMRES	E9		2
WQASUMS	4C		2
WQASUMS#	4C		3
WQASUMSE	50		3
WQASUMSES	E8		2
WQASUMU	6C		2
WQASUMW	5C		2
WQASUMW#	5C		3
WQASUMWE	60		3
WQASUMWES	EA		2
WQASUMX	84		2
WQASUREP	24		2
WQATCP	F	01	3
WQATCPNEW	1A	80	3
WQATIME	7C		3
WQATOKEN	F	08	3
WQATWS	EB		2
WQAUX009	F	02	3
WQAVER	4		2
WQAVIRTPTR	80		2
WQAVTYPE	D0		2
WQAWAIT	18	40	3
WQAWQANX	1C		2
WQAWRTST	ED	10	3
WQAWS_VARY	6	04	3
WQAWSFLG	F0		2
WQAWSNAM	8		2
WQAWSREP	C		2
WQAWSR1N	9C		2
WQAWSR2N	9E		2
WQAWSSTAT	F	04	3
WQAWSTYP	7		2
WQAWTO	F	40	3
WQAZCAGENT	F0	10	3
WQASUBV	18	20	3

WSD - Workstation description record

Name : DCLWSD

Function:
This segment declares a workstation definition, physically located in the workstation description database (EQQWSDS) The WSD structure is as follows:

```
                 1 WSD
      2 common data
      2 a table with one row per day defined describing the day
      2 a table with one row per open time interval describing the interval
```

Each entry in the day table contains an index into the interval table for the first interval defined for that particular day.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	*	WSD	
0	(0)	CHARACTER	140	WSDCOMMON	
0	(0)	CHARACTER	6	*	RESERVED FOR VSAM
6	(6)	CHARACTER	4	WSDKEY	*** KEY FIELD ***********
6	(6)	CHARACTER	4	WSDNAME	WS NAME
10	(A)	CHARACTER	4	WSDEYE	EYECATCHER
14	(E)	UNSIGNED	1	WSDVERS	VERSION NUMBER
15	(F)	CHARACTER	4	WSDTRSPT	TIME FROM PRE WS HHMM
19	(13)	CHARACTER	4	WSDOPDUR	OPER DURATION DEFAULT HHMM
23	(17)	CHARACTER	8	WSDROUT	PRINTOUT ROUTING FOR DP
31	(1F)	CHARACTER	32	WSDDESCR	WS DESCRIPTION
63	(3F)	CHARACTER	1	WSDTYPE	TYPE OF WORKSTATION
64	(40)	BITSTRING	1	WSDFLG	
		1...		WSDPREP	JOBSETUP USED ON THIS WS?
		.1..		WSDPSJT	CONTROL ON SERVERS
		..1.		WSDSPLIT	SPLITTABLE ATTRIBUTE
		...1		WSDSTC	STARTED TASK
		.... 1...		WSDWTO	DEADLINE WTO
		.... .1..		WSDZCAGN	Z-CENTRIC AGENT WORKSTATION
		.... ..1.		WSDWAIT	WAIT WORKSTATION
		1		WSDVIRT	VIRTUAL WORKSTATION
65	(41)	CHARACTER	1	WSDREP	REPORTING ATTRIBUTE
66	(42)	CHARACTER	2	WSDR1NAM	WS RESOURCE NAME
68	(44)	CHARACTER	2	WSDR1FLG	RESOURCE INDICATORS
		1...		WSDR1PLAN	RESOURCE USED AT PLANNING
		.1..		WSDR1CONT	RESOURCE USED AT CONTROL
68	(44)	BITSTRING	1	*	FREE
70	(46)	CHARACTER	2	WSDR2NAM	WS RESOURCE NAME
72	(48)	CHARACTER	2	WSDR2FLG	RESOURCE INDICATORS
		1...		WSDR2PLAN	RESOURCE USED AT PLANNING
		.1..		WSDR2CONT	RESOURCE USED AT CONTROL
72	(48)	BITSTRING	1	*	FREE
74	(4A)	CHARACTER	6	WSDUPDAT	LATEST UPDATE, THIS WSD
80	(50)	CHARACTER	4	WSDUPTI	LATEST UPDATE TIME
84	(54)	CHARACTER	8	WSDUSER	ID OF USER WHO MADE THE.. ..LATEST CHANGE TO THIS WS
92	(5C)	CHARACTER	8	WSDSUDS	SU/RE DATA SET DDNAME
100	(64)	SIGNED	2	WSDVDES#	NUMBER OF VIRTUAL DEST
102	(66)	BITSTRING	1	WSDFLG1	
		1...		WSDINVOPT	INVALID OPTION SPEC

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		.1..		WSDDYN	DYNAMIC SCHEDULING WORKSTATION
		..1.		WSDDYNIO	DYNAMIC INVALID OPT
		...1 1111		*	FREE
103	(67)	CHARACTER	1	WSDPSERV	SERVER USEAGE N/B/P/C
104	(68)	CHARACTER	1	WSDRENG	REMOTE ENGINE TYPE Z/D
105	(69)	CHARACTER	2	*	FREE
107	(6B)	CHARACTER	1	WSDTWS	FTA WORKSTATION
108	(6C)	SIGNED	2	WSDDAY#	TOTAL NO. OF DAYS
110	(6E)	SIGNED	2	WSDTOTIVL#	TOTAL NO. OPEN TIME INTVLS
112	(70)	SIGNED	2	WSDACCM#	0=NO ACC. METH. DATA 1=ACC METH DATA PRESENT
114	(72)	CHARACTER	1	WSDAUTO	SYSTEM AUTOMATION
115	(73)	CHARACTER	1	*	FREE
116	(74)	CHARACTER	8	WSDLUTS	LAST UPDATE TIMESTAMP
124	(7C)	SIGNED	2	WSDEOPT#	0 = 'NO E2E OPTIONS DATA'
126	(7E)	CHARACTER	14	*	FREE
140	(8C)	CHARACTER	*	WSDVARSECT	VARIABLE SECTION

DAY RELATED INFORMATION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
140	(8C)	STRUCTURE	52	WSDDAY (*)	
140	(8C))	CHARACTER	6	WSDDATE	SPECIFIC DATE
140	(8C)	SIGNED	2	WSDDAYNO	1-MONDAY 7-SUNDAY 8-STAND
146	(92	CHARACTER	24	WSDDESD	DESCRIPTION OF DAY
170	(AA)	BITSTRING	1	WSDDFLG	
		1...		WSDDSPECIFIC	SPECIFIC DAY INDICATOR
171	(AB)	CHARACTER	13	*	FREE
184	(B8)	SIGNED	4	WSDDIVL#	NO. OF INTERVALS
188	(BC)	SIGNED	4	WSDIVL_IX	INDEX OF FIRST OPEN TIME.. ..INTERVAL FOR THIS DAY

OPEN TIME INTERVAL INFORMATION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	32	WSDIVL (*)	
0	(0)	CHARACTER	4	WSDSTART	START TIME OF INTERVAL
4	(4)	CHARACTER	4	WSDENDT	END TIME OF INTERVAL
8	(8)	SIGNED	2	WSDSERV#	NUMBER OF PARALLEL SERVERS
10	(A)	SIGNED	2	WSDR1CAP	RESOURCE CAPACITY
12	(C)	SIGNED	2	WSDR2CAP	RESOURCE CAPACITY
14	(E)	CHARACTER	4	WSDAWS	ALTERNATE WS NAME
18	(12)	CHARACTER	14	*	FREE

ACCESS METHOD INFORMATION

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	72	WSDACC (*)	
0	(0)	CHARACTER	12	WSDACCMN	ACCESS METHOD NAME
12	(C)	CHARACTER	52	WSDADR	SAP HOST ADDRESS
12	(C)	CHARACTER	26	WSDADR1	SAP HOST ADDRESS

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
38	(26)	CHARACTER	26	WSDADR2	SAP HOST ADDRESS
64	(40)	SIGNED	4	WSDPORT	SAP PORT NUMBER
68	(44)	CHARACTER	4	*	FREE

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	16	WSDVDES (*)	
0	(0)	CHARACTER	8	WSDVDNAM	REAL DESTINATION
8	(8)	SIGNED	4	WSDVDID	WSV ID
12	(C)	CHARACTER	4	*	FREE

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	173	WSDEOPT (*)	
0	(0)	CHARACTER	47	WSDEOJU	DEFAULT USER NAME
47	(2F)	CHARACTER	1	WSDEOJP	PASSWORD SPECIFIED
48	(30)	CHARACTER	40	WSDEOJT	DEFAULT ACCESS METHOD
88	(58)	CHARACTER	1	WSDEOBR	IS A BROKER (Y/N)
89	(59)	CHARACTER	40	WSDEOPL	BROKER AGENT POOL NAME
129	(81)	CHARACTER	40	WSDEODP	BROKER DYNAMIC POOL
169	(A9)	CHARACTER	4	*	FREE

Cross reference

Name	Hex Offset	Hex Value	Level
WSD	0		1
WSDACC	0		1
WSDACCM#	70		3
WSDACCMN	0		2
WSDADR	C		2
WSDADR1	C		3
WSDADR2	26		3
WSDAUTO	72		3
WSDAWS	E		2
WSDCOMMON	0		2
WSDDATE	8C		2
WSDDAY	8C		1
WSDDAY#	6C		3
WSDDAYNO	8C		3
WSDDESCR	1F		3
WSDDESD	92		2
WSDDFLG	AA		2
WSDDIVL#	B8		2
WSDDSPECIFIC	AA	80	3
WSDDYN	66	40	4

Name	Hex Offset	Hex Value	Level
WEDDYNIO	66	20	4
WSDENDT	4		2
WSDEOBR	58		2
WSDEODP	81		2
WSDEOJP	2F		2
WSDEOJT	30		2
WSDEOJU	0		2
WSDEOPL	59		2
WSDEOPT	0		1
WSDEOPT#	7C		3
WSDEYE	A		3
WSDFLG	40		3
WSDFLG1	66		3
WSDINVOPT	66	80	4
WSDIVL	0		1
WSDIVL_IX	BC		2
WSDKEY	6		3
WSDLUTS	74		3
WSDNAME	6		4
WSDOPDUR	13		3
WSDPORT	40		2
WSDPREP	40	80	4
wsdpserv	67		3
WSDPSJT	40	40	4
WSDRENG	68		3
WSDREP	41		3
WSDROUT	17		3
WSDRSEQ#	66		3
WSDR1CAP	A		2
WSDR1CONT	44	40	4
WSDR1FLG	44		3
WSDR1NAM	42		3
WSDR1PLAN	44	80	4
WSDR2CAP	C		2
WSDR2CONT	48	40	4
WSDR2FLG	48		3
WSDR2NAM	46		3
WSDR2PLAN	48	80	4
WSDSERV#	8		2
WSDSPLIT	40	20	4
WSDSSEQ#	64		3

Name	Hex Offset	Hex Value	Level
WSDSTART	0		2
WSDSTC	40	10	4
WSDSUDS	5C		3
WSDTOTIVL#	6E		3
WSDTRSPT	F		3
WSDTWS	6B		3
WSDTYPE	3F		3
WSDUPDAT	4A		3
WSDUPTI	50		3
WSDUSER	54		3
WSDVARSECT	8C		2
WSDVERS	E		3
WSDVIRT	40	01	4
WSDWAIT	40	02	4
WSDWTO	40	08	4
WSDZCAGN	40	04	4

WSN - Workstation resource report record in DP

Name : DCLWSN

Function:
Describes layout of parallel servers, R1 and R2 indicating the planned workstation load measured in parallel servers, R1 and R2 units. There is one record for each workstation and 'day' in the plan. A 'day' starts|ends on the time given in planhour. For each 15 minutes the amount is recorded of R1/R2 resources needed (according to the operation descriptions) multiplied by 15 planned server minutes for all operations with start-end interval covering the interval (if 1/3 of the interval covered - 1/3*amount). An example: An operation planned between 14.30 and 14.48 using 1 server, 50 R1, and 2 R2 will record 15 server minutes, 15*50 R1 units and 15*2 R2 units in the 14.30-14.45 slot and will record 3 server minutes, 3*50 R1 units and 3*2 R2 units in the 14.45-15.00 slot. The records have the standard recsize and keysize of DPIN records.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	273	WSNAREA	PARALLEL SERV|R1|R2 REC
0	(0)	CHARACTER	74	WSNKEY	KEY
0	(0)	SIGNED	2	WSNTYPE	ALWAYS TYPE 3
2	(2)	CHARACTER	8	WSNROUT	REPORT ROUTING
10	(A)	CHARACTER	4	WSNWSID	WORKSTATION ID
14	(E)	SIGNED	2	WSNSTYPE	SUBTYPE SERVER MINUTES= 3 R1 = 4 R2 = 5
16	(10)	CHARACTER	6	WSNDATE	REPORT 'DAY' (YYMMDD)
22	(16)	CHARACTER	2	WSNSTRNG	R1|R2 STRING IN REPORTS BLANK FOR PARALLEL SERV
24	(18)	CHARACTER	1	WSNRPLN	RESOURCE PLANNING Y|N
25	(19)	CHARACTER	34	*	ADJUST TO KEYLENGTH
59	(3B)	CHARACTER	15	*	TO ADD UP TO 74 BYTES
74	(4A)	CHARACTER	199	WSNDATA	DATA PART OF WSN RECORD

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
74	(4A)	SIGNED	2	WSNFACT	FACTOR TO MULTIPLY AMOUNT
76	(4C)	UNSIGNED	1	WSNAMT (96)	OF RESOURCES
172	(AC)	UNSIGNED	1	WSNLIM (96)	LIMIT OF RESOURCES
268	(10C)	CHARACTER	5	*	TO ADD UP TO STD RECSIZE

Cross reference

Name	Hex Offset	Hex Value	Level
WSNAMT	4C		3
WSNAREA	0		1
WSNDATA	4A		2
WSNDATE	10		3
WSNFACT	4A		3
WSNKEY	0		2
WSNLIM	AC		3
WSNROUT	2		3
WSNRPLN	18		3
WSNSTRNG	16		3
WSNSTYPE	E		3
WSNTYPE	0		3
WSNWSID	A		3

WSP - Workstation plan report record in DP

Name : DCLWSP

Function:
describes layout of wsp records representing report data on operations on certain workstations in the new current plan. New first operations may appear both as an ordinary all operations record (subtype 16) and a first operation record (subtype 32) depending on report options selected. Note that there is additional record for all predecessors after the 2nd (see DCLDAIOP). An additional subsubtype of record is used to represent opentime interval (changes) with a key that sorts the opentime record before the operations planned in the opentime. The records have the standard recsize and keysize of DPIN records.

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	250	WSPAREA	WSP RECORD
0	(0)	CHARACTER	132	WSPKEY	KEY
0	(0)	SIGNED	2	WSPTYPE	TYPE IS ALWAYS 3
2	(2)	CHARACTER	8	WSPROUT	REPORT ROUTING
10	(A)	CHARACTER	4	WSPWSID	WSID OF OPERATION
14	(E)	SIGNED	2	WSPSTYPE	SUBTYPE 1 = (ALL) OPER. SUBTYPE 2 = FIRST OPER.
16	(10)	CHARACTER	1	WSPNONR	NONREPORTING WS (Y|N)
17	(11)	CHARACTER	10	WSPPLNS	OPERATION PLANNED START (OR WSO OPENTIME START)

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
17	(11)	CHARACTER	6	WSPPLNSD	DAY (YYMMDD)
23	(17)	CHARACTER	4	WSPPLNST	TIME (HHMM)
23	(17)	CHARACTER	2	WSPPLNSH	TIME (HH)
25	(19)	CHARACTER	2	WSPPLNSM	TIME (MM)
27	(1B)	UNSIGNED	1	WSPSSTYP	1 = OPENTIME RCD 2 = ORDINARY OPERATION RCD
		THE ABOVE FIELDS FILLED IN FOR ALL RECORD TYPES			
28	(1C)	CHARACTER	10	WSPOCIA	OCC IA AFTER MCP
28	(1C)	CHARACTER	6	WSPOCIAD	DAY (YYMMDD) |BLANK
34	(22)	CHARACTER	4	WSPOCIAT	TIME (HHMM) |BLANK
34	(22)	CHARACTER	2	WSPOCIAH	TIME (HH)
36	(24)	CHARACTER	2	WSPOCIAM	TIME (MM)
38	(26)	CHARACTER	19	WSPID	
38	(26)	CHARACTER	16	WSPADID	APPLICATION ID OF OPER.
54	(36)	CHARACTER	3	WSPOPNO	OPERATION NUMBER
57	(39)	CHARACTER	41	WSPPRE	OPERATION PRED |BLANK
57	(39)	CHARACTER	16	WSPPREOC	PREDECESSOR ADID
73	(49)	CHARACTER	3	WSPPREOP	PREDECESSOR OP NO |BLANK
76	(4C)	CHARACTER	4	WSPPREWS	PREDECESSOR WSID |BLANK
80	(50)	CHARACTER	8	WSPPREJN	PREDECESSOR JOBNAME|BLANK
88	(58)	CHARACTER	10	WSPPREIA	YYMMDDHHMM PRED IA |BLANK
88	(58)	CHARACTER	6	WSPPREID	YYMMDD
94	(5E)	CHARACTER	2	WSPPREIH	HH
96	(60)	CHARACTER	2	WSPPREIM	MM
98	(62)	CHARACTER	34	*	ADJUST HEYLENGTJH
132	(84)	CHARACTER	118	WSPDATA	DATA PART OF WSP RECORD
132	(84)	CHARACTER	1	WSPOPPRI	OPERATION PRIORITY
133	(85)	CHARACTER	1	WSPWSTYP	WORKSTATION TYPE
134	(86)	CHARACTER	8	WSPOPJOB	OPERATION JOBNAME
142	(8E)	CHARACTER	24	WSPOPTXT	OPERATION TEXT
166	(A6)	CHARACTER	2	WSPOPPS	PARALLEL SERVERS
168	(A8)	CHARACTER	2	WSPOPR1	OPERATION R1 RESOURCE
170	(AA)	CHARACTER	2	WSPOPR2	OPERATION R2 RESOURCE
172	(AC)	CHARACTER	8	WSPOPRES	ID OF CRITICAL RESOURCE
180	(B4)	CHARACTER	1	WSPOPEXC	Y = EXCLUSIVE USE
181	(B5)	CHARACTER	1	WSPOPST	OPERATION STATUS
182	(B6)	CHARACTER	1	WSPOPRER	Y = PLANNED RERUN, ELSE N
183	(B7)	CHARACTER	4	WSPOPDUR	OPERATION DURATION HHMM REMAINING
187	(BB)	CHARACTER	8	WSPOPFRM	OPERATION FORMNUMBER
195	(C3)	CHARACTER	1	WSPOPCLS	OPERATION JOB|SYSOUT CLASS
196	(C4)	CHARACTER	10	WSPPLNE	OPERATION PLANNED END
196	(C4)	CHARACTER	6	WSPPLNED	DAY (YYMMDD)
202	(CA)	CHARACTER	4	WSPPLNET	TIME (HHMM)
202	(CA)	CHARACTER	2	WSPPLNEH	TIME (HH)
204	(CC)	CHARACTER	2	WSPPLNEM	TIME (MM)
206	(CE)	CHARACTER	10	WSPOPLU	OPERATION LATEST OUT
206	(CE)	CHARACTER	6	WSPOPLUD	DAY (YYMMDD)
212	(D4)	CHARACTER	4	WSPOPLUT	TIME (HHMM)
212	(D4)	CHARACTER	2	WSPOPLUH	TIME (HH)
214	(D6)	CHARACTER	2	WSPOPLUM	TIME (MM)
216	(D8)	CHARACTER	1	WSPOPDL	Y IF DL = LATESTOUT ELSE N
217	(D9)	CHARACTER	16	WSPOWNER	APPLICATION OWNER

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
THE SECTION ABOVE NOT FILLED IN FOR OPENTIME RECORDS THE SECTION BELOW FILLED IN FOR OPENTIME RECORDS ONLY					
233	(E9)	CHARACTER	4	WSPWSOET	WS OPENTIME END HHMM
233	(E9)	CHARACTER	2	WSPWSOEH	TIME (HH)
235	(EB)	CHARACTER	2	WSPWSOEM	TIME (MM)
237	(ED)	CHARACTER	2	WSPWSO#S	WS OPENTIME NO OF SERVERS
THE FIELD BELOW NOT FILLED IN FOR OPENTIME RECORDS					
239	(EF)	CHARACTER	1	WSPNEW	NEW OPR FROM LTP Y | N
THE FIELD FILLED IN FOR 1ST-OPER-SUBTYPE ONLY					
240	(F0)	CHARACTER	10	WSPOPIA	OPR IA (OR = OCC IA)
240	(F0)	CHARACTER	6	WSPOPIAD	DAY (YYMMDD)
246	(F6)	CHARACTER	4	WSPOPIAT	TIME (HHMM)
246	(F6)	CHARACTER	2	WSPOPIAH	TIME (HH)
248	(F8)	CHARACTER	2	WSPOPIAM	TIME (MM)

Cross reference

Name	Hex Offset	Hex Value	Level
WSPADID	26		4
WSPAREA	0		1
WSPDATA	84		2
WSPID	26		3
WSPKEY	0		2
WSPNEW	EF		3
WSPNONR	10		3
WSPOCIA	1C		3
WSPOCIAD	1C		4
WSPOCIAH	22		5
WSPOCIAM	24		5
WSPOCIAT	22		4
WSPOPCLS	C3		3
WSPOPDL	D8		3
WSPOPDUR	B7		3
WSPOPEXC	B4		3
WSPOPFRM	BB		3
WSPOPIA	F0		3
WSPOPIAD	F0		4
WSPOPIAH	F6		5
WSPOPIAM	F8		5
WSPOPIAT	F6		4
WSPOPJOB	86		3
WSPOPLU	CE		3
WSPOPLUD	CE		4
WSPOPLUH	D4		5

Name	Hex Offset	Hex Value	Level
WSPOPLUM	D6		5
WSPOPLUT	D4		4
WSPOPNO	36		4
WSPOPPRI	84		3
WSPOPPS	A6		3
WSPOPRER	B6		3
WSPOPRES	AC		3
WSPOPR1	A8		3
WSPOPR2	AA		3
WSPOPST	B5		3
WSPOPTXT	8E		3
WSPOWNER	D9		3
WSPPLNE	C4		3
WSPPLNED	C4		4
WSPPLNEH	CA		5
WSPPLNEM	CC		5
WSPPLNET	CA		4
WSPPLNS	11		3
WSPPLNSD	11		4
WSPPLNSH	17		5
WSPPLNSM	19		5
WSPPLNST	17		4
WSPPRE	39		3
WSPPREIA	58		4
WSPPREID	58		5
WSPPREIH	5E		5
WSPPREIM	60		5
WSPPREJN	50		4
WSPPREOC	39		4
WSPPREOP	49		4
WSPPREWS	4C		4
WSPROUT	2		3
WSPSSTYP	1B		3
WSPSTYPE	E		3
WSPTYPE	0		3
WSPWSID	A		3
WSPWSO#S	ED		3
WSPWSOEH	E9		4
WSPWSOEM	EB		4
WSPWSOET	E9		3
WSPWSTYP	85		3

WSV - Virtual workstation description record

Name : DCLWSV

Function:
This segment declares a virtual workstation definition, physically located in the workstation description database (EQQWSDS). For each actual destination associated to a virtual workstation,
defines the same kind of information as WSD. The WSV structure is as follows:
1 WSV
2 common data
2 a table with one row per day defined describing the day
2 a table with one row per open time interval describing the interval

Each entry in the day table contains an index into the interval table for the first interval defined for that particular day.

Offsets		Type	Len	Name	Description
0	(0)	CHARACTER	76	WSVCOMMON	
0	(0)	CHARACTER	2	WSVKEYID	ALWAYS '22'
2	(2)	CHARACTER	8	WSVKEY	
2	(2)	CHARACTER	4	WSVWSNAM	WORKSTATION NAME
6	(6)	SIGNED	4	WSVSEQNO	DESTINATION ID
10	(A)	CHARACTER	4	WSVEYE	EYECATCHER 'WSV '
14	(E)	UNSIGNED	1	WSVVERS	VERSION NUMBER
15	(F)	CHARACTER	1	WSVPSERV	SERVER USAGE (N/C)
16	(10)	CHARACTER	8	*	FREE
24	(18)	CHARACTER	8	WSVDESTN	DESTINATION NAME
32	(20)	SIGNED	2	WSVDAY#	NUMBER OF DAYS
34	(22)	SIGNED	2	WSVTOTIVL#	NUMBER OF IVL
36	(24)	CHARACTER	8	WSVLUTS	LAST UPDATE TOD
44	(2C)	CHARACTER	8	WSVUSER	LAST UPDATE USER
52	(34)	CHARACTER	6	WSVUPDAT	LATEST UPDATE, THIS WSD
58	(3A)	CHARACTER	4	WSVUPTI	LATEST UPDATE TIME
62	(3E)	CHARACTER	2	WSVR1NAM	WS RESOURCE NAME
64	(40)	CHARACTER	2	WSVR1FLG	RESOURCE INDICATORS
		1...		WSVR1PLAN	RESOURCE USED AT PLANNING
		.1..		WSVR1CONT	RESOURCE USED AT CONTROL
64	(40)	BITSTRING	1	*	FREE
66	(42)	CHARACTER	2	WSVR2NAM	WS RESOURCE NAME
68	(44)	CHARACTER	2	WSVR2FLG	RESOURCE INDICATORS
		1...		WSVR2PLAN	RESPURCE USED AT PLANNING
		.1..		WSVR2CONT	RESOURCE USED AT CONTROL
68	(44)	BITSTRING	1	*	FREE
70	(46)	CHARACTER	6	*	FREE
76	(4C)	CHARACTER	*	WSVVARSECT	VARIABLE SECTION

DAY RELATED INFORMATION

Offsets		Type	Len	Name	Description
76	(4C)	STRUCTURE	52	WSVDAY(*)	
76	(4C)	CHARACTER	6	WSVDATE	SPECIFIC DATE
76	(4C)	SIGNED	2	WSVDAYNO	1-MONDAY 7-SUNDAY 8-STAND
82	(52)	CHARACTER	24	WSVDESD	DESCRIPTION OF DAY
106	(6A)	BITSTRING	1	WSVDFLG	
		1...		WSVDSPECIFIC	SPECIFIC DAY INDICATOR
107	(6B)	CHARACTER	13	*	FREE
120	(78)	SIGNED	4	WSVDIVL#	NO. OF INTERVALS
124	(7C)	SIGNED	4	WSVIVL_IX	INDEX OF FIRST OPEN TIME.. ..INTERVAL FOR THIS DAY

OPEN TIME INTERVAL INFORMATION

Offsets	Type	Len	Name	Description

```
    0    (0) STRUCTURE    32  WSVIVL(*)
    0    (0) CHARACTER     4  WSVSTART       START TIME OF INTERVAL
    4    (4) CHARACTER     4  WSVENDT        END TIME OF INTERVAL
    8    (8) UNSIGNED      2  WSVSERV#       NUMBER OF PARALLEL SERV
   10    (A) UNSIGNED      2  WSVR1CAP       RESOURCE CAPACITY
   12    (C) UNSIGNED      2  WSVR2CAP       RESOURCE CAPACITY
   14    (E) CHARACTER     4  *
   18   (12) CHARACTER    14  *              FREE
```

Cross reference

```
                                             HEX        HEX
NAME                                         OFFSET     VALUE     LEVEL
____                                         ______    ________   _____
WSV                                               0                   1
WSVCOMMON                                         0                   2
WSVDATE                                          4C                   2
WSVDAY                                           4C                   1
WSVDAY#                                          20                   3
WSVDAYNO                                         4C                   3
WSVDESD                                          52                   2
WSVDESTN                                         18                   3
WSVDFLG                                          6A                   2
WSVDIVL#                                         78                   2
WSVDSPECIFIC                                     6A           80      3
WSVENDT                                           4                   2
WSVEYE                                            A                   3
WSVIVL                                            0                   1
WSVIVL_IX                                        7C                   2
WSVKEY                                            2                   3
WSVKEYID                                          0                   3
WSVLUTS                                          24                   3
WSVPSERV                                          F                   3
WSVR1CAP                                          A                   2
WSVR1CONT                                        40           40      4
WSVR1FLG                                         40                   3
WSVR1NAM                                         3E                   3
WSVR1PLAN                                        40           80      4
WSVR2CAP                                          C                   2
WSVR2CONT                                        44           40      4
WSVR2FLG                                         44                   3
WSVR2NAM                                         42                   3
WSVR2PLAN                                        44           80      4
WSVSEQNO                                          6                   4
WSVSERV#                                          8                   2
WSVSTART                                          0                   2
WSVTOTIVL#                                       22                   3
WSVUPDAT                                         34                   3
WSVUPTI                                          3A                   3
WSVUSER                                          2C                   3
WSVVARSECT                                       4C                   2
WSVVERS                                           E                   3
WSVWSNAM                                          2                   4
```

XD01 - XD Header record

Name : DCLXD01

Function:
This record ....

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	244	XDREC01	

Key:

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	68	XDKEY01	
0	(0)	CHARACTER	2	XDKYP01	TYPE IS ALWAYS 01
2	(2)	CHARACTER	66	XDKF01	SHOULD CONTAIN BLANKS

Body:

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
68	(44)	CHARACTER	176	XDBDY01	HDR RECORD BODY
68	(44)	CHARACTER	4	XDEYE01	EYECATCHER
72	(48)	UNSIGNED	1	XDVERS01	VERSION NUMBER
73	(49)	CHARACTER	1	*	FREE
74	(4A)	CHARACTER	10	XDCR	CURRENT PLAN CREATION
74	(4A)	CHARACTER	6	XDCRDAT	DATE
80	(50)	CHARACTER	4	XDCRTIM	TIME
84	(54)	CHARACTER	10	XDEND	LAST PLAN NEXT PERIOD END
84	(54)	CHARACTER	6	XDENDAT	DATE
90	(5A)	CHARACTER	4	XDENTIM	TIME
94	(5E)	CHARACTER	10	XDBU	LAST BACKUP
94	(5E)	CHARACTER	6	XDBUDAT	DATE
100	(64)	CHARACTER	4	XDBUTIM	TIME
104	(68)	CHARACTER	10	XDLP	TAIL END
104	(68)	CHARACTER	6	XDLPDAT	DATE
110	(6E)	CHARACTER	4	XDLPTIM	TIME
114	(72)	CHARACTER	10	XDREPS	REPORT PERIOD START
114	(72)	CHARACTER	6	XDREPSD	DATE
120	(78)	CHARACTER	4	XDREPST	TIME
124	(7C)	CHARACTER	10	XDLEVEL	XD level
124	(7C)	CHARACTER	2	XDLEV_VER	-- version
126	(7E)	CHARACTER	8	XDLEV_FMID	-- fmid
134	(86)	CHARACTER	10	XDINIT	blank or 'Init Empty'
144	(90)	CHARACTER	1	XD01DPSET	Y: XD99 to be processed for status change
145	(91)	CHARACTER	99	*	free

Constants

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
2		DECIMAL	244	MAXLENXD01	
2		CHARACTER	01	XDTYP01	first 2 bytes of key
4		CHARACTER	XD01	XDEYEC01	EYEcatcher
1		DECIMAL	1	XDVERSC01	version number

Cross reference

Name	Hex Offset	Hex Value	Level
XDBDY01	44		2
XDBU	5E		3
XDBUDAT	5E		4
XDBUTIM	64		4
XDCR	4A		3
XDCRDAT	4A		4
XDCRTIM	50		4
XDEND	54		3
XDENDAT	54		4
XDENTIM	5A		4
XDEYE01	44		3
XDINIT	86		3
XDKEY01	0		2
XDKF01	2		3
XDKYP01	0		3
XDLEV_FMID	7E		4
XDLEV_VER	7C		4
XDLEVEL	7C		3
XDLP	68		3
XDLPDAT	68		4
XDLPTIM	6E		4
XDREC01	0		1
XDREPS	72		3
XDREPSD	72		4
XDREPST	78		4
XDVERS01	48		3
XD01DPSET	90		3

XD02 - XD Header record

Name : DCLXD02

Function:
This record ....

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	*	XDREC02	XD HDR RECORD

Key:

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	CHARACTER	68	XDKEY02	
0	(0)	CHARACTER	2	XDKYP02	TYPE IS ALWAYS 02
2	(2)	CHARACTER	64	XDREQ02	
2	(2)	CHARACTER	8	XDKDEST02	REQUESTER DESTINATION
10	(A)	CHARACTER	56	XDKALIAS02	REQUESTER ALIAS
66	(42)	CHARACTER	2	XDKF02	FREE

Body:

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
68	(44)	CHARACTER	224	XDBDY02	HDR RECORD BODY
68	(44)	CHARACTER	4	XDEYE02	EYECATCHER
72	(48)	UNSIGNED	1	XDVERS02	VERSION NUMBER
73	(49)	CHARACTER	1	XD02STAT	SUBSCRIBE STATUS (see below)
74	(4A)	UNSIGNED	1	XD02FLAGS	MATCHING CRITERIA FLAGS:
		1...		XD02WAITWQA	ON: started on a wait ws
		.1..		XD02FAILNFY	ON: failed to notify
		..1.		XD02EVOP	ON: started executing
		...1		XD02RECOV	ON: recovery in progress
		 1111		*	FREE
75	(4B)	UNSIGNED	1	*	FREE
76	(4C)	CHARACTER	8	XD02TOK	XD02 ID: CREATION TOD
84	(54)	SIGNED	2	XD02ADEST#	Number of alternate Dest
86	(56)	SIGNED	2	XD02OPN	OPERATION NUMBER TO BIND
88	(58)	CHARACTER	16	XD02ADID	APPLICATION NAME TO BIND
104	(68)	CHARACTER	10	XD02BASEIA	Base IA for matching
104	(68)	CHARACTER	6	XD02BASEIAD	- YYMMDD
110	(6E)	CHARACTER	4	XD02BASEIAT	- HHMM
114	(72)	CHARACTER	10	XD02IFROMIA	match Interval start:
114	(72)	CHARACTER	6	XD02IFROMIAD	- YYMMDD
120	(78)	CHARACTER	4	XD02IFROMIAT	- HHMM
124	(7C)	CHARACTER	10	XD02ITOIA	match Interval end:
124	(7C)	CHARACTER	6	XD02ITOIAD	- YYMMDD
130	(82)	CHARACTER	4	XD02ITOIAT	- HHMM
134	(86)	CHARACTER	26	XD02LTPKEY	LTP key (used during pending)
134	(86)	CHARACTER	6	XD02LOCCDATE	- Run dateYYMMDD
140	(8C)	CHARACTER	16	XD02LOCCNAME	- Application Name
156	(9C)	CHARACTER	4	XD02LOCCTIME	- IA time HHMM
160	(A0)	CHARACTER	8	XD02CP3COCCTOK	Occurrence Token of operation
168	(A8)	CHARACTER	10	XD02JSTART	Matched job start
168	(A8)	CHARACTER	6	XD02JSTARTD	... date

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
174	(AE)	SIGNED	4	XD02JSTARTT	... time
178	(B2)	CHARACTER	1	XD02JSTAT	Matched job status
179	(B3)	CHARACTER	1	*	FREE
180	(B4)	CHARACTER	10	XD02JEND	Matched job end
180	(B4)	CHARACTER	6	XD02JENDD	... date
186	(BA)	SIGNED	4	XD02JENDT	... time
190	(BE)	CHARACTER	8	XD02JNAME	Matched job name
198	(C6)	CHARACTER	4	XD02JEC	Matched job error code
202	(CA)	CHARACTER	4	XD02WSNAME	matched ws name
206	(CE)	SIGNED	2	XD02IAGMT	gmt for IA conversion
208	(D0)	SIGNED	2	XD02JSGMT	gmt for start time conv
210	(D2)	SIGNED	2	XD02JEGMT	gmt for end time conv
212	(D4)	SIGNED	4	XD02DIDX	destination index
216	(D8)	SIGNED	4	XD02JADUR	actual duration in secs
220	(DC)	CHARACTER	72	*	FREE

Variable part:

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
292	(124)	CHARACTER	*	XDVAR02	VAR DATA

Alternate dest variable part:

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	16	XDADEST(*)	
0	(0)	CHARACTER	8	XDADESTNAME	DESTINATION NAME
8	(8)	SIGNED	4	XDADIDX	DESTINATION DTB index
12	(C)	SIGNED	4	*	free

Constants

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
2		CHARACTER	02	XDTYP02	first 2 bytes of key
4		CHARACTER	XD02	XDEYEC02	EYEcatcher
1		CHARACTER	1	XDVERSC02	version number
1		CHARACTER	L	WAITLT_XD2	L waiting for LTP
1		CHARACTER	K	LTLOCK_XD2	K waiting for LTP lock
1		CHARACTER	P	WAITCP_XD2	P waiting for CP
1		CHARACTER	S	SENDCH_XD2	S sending status changes
1		CHARACTER	E	NOTBND_XD2	E not bound
1		CHARACTER	C	KILLED_XD2	C killed
1		CHARACTER	D	DELETE_XD2	D deleted

Cross reference

Name	Hex Offset	Hex Value	Level
XDADEST	0		1

Name	Hex Offset	Hex Value	Level
XDADESTNAME	0		2
XDADIDX	8		2
XDBDY02	44		2
XDEYE02	44		3
XDKALIAS02	A		4
XDKDEST02	2		4
XDKEY02	0		2
XDKF02	42		3
XDKYP02	0		3
XDREC02	0		1
XDREQ02	2		3
XDVAR02	124		2
XDVERS02	48		3
XD02ADEST#	54		3
XD02ADID	58		3
XD02BASEIA	68		3
XD02BASEIAD	68		4
XD02BASEIAT	6E		4
XD02CP3COCCTOK	A0		3
XD02DIDX	D4		3
XD02EVOP	4A	20	4
XD02FAILNFY	4A	40	4
XD02FLAGS	4A		3
XD02IAGMT	CE		3
XD02IFROMIA	72		3
XD02IFROMIAD	72		4
XD02IFROMIAT	78		4
XD02ITOIA	7C		3
XD02ITOIAD	7C		4
XD02ITOIAT	82		4
XD02JADUR	D8		3
XD02JEC	C6		3
XD02JEGMT	D2		3
XD02JEND	B4		3
XD02JENDD	B4		4
XD02JENDT	BA		4
XD02JNAME	BE		3
XD02JSGMT	D0		3
XD02JSTART	A8		3
XD02JSTARTD	A8		4
XD02JSTARTT	AE		4

Name	Hex Offset	Hex Value	Level
XD02JSTAT	B2		3
XD02LOCCDATE	86		4
XD02LOCCNAME	8C		4
XD02LOCCTIME	9C		4
XD02LTPKEY	86		3
XD02OPN	56		3
XD02RECOV	4A	10	4
XD02STAT	49		3
XD02TOK	4C		3
XD02WAITWQA	4A	80	4
XD02WSNAME	CA		3

XD03 - XD HDR RECORD

Name : DCLXD03

Function:
This record ....

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	76	XDREC03	XD HDR RECORD

KEY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	68	XDKEY03	
0	(0)	CHARACTER	2	XDTYP03	TYPE IS ALWAYS 03
2	(2)	CHARACTER	64	XDOCC03	
2	(2)	CHARACTER	16	XDOCCADID	
18	(12)	CHARACTER	10	XDOCCIA	
18	(12)	CHARACTER	6	XDOCCIAD	
24	(18)	CHARACTER	4	XDOCCIAT	
28	(1C)	CHARACTER	38	*	blank
66	(42)	CHARACTER	2	*	FREE

BODY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
68	(44)	CHARACTER	8	XDBDY03	HDR RECORD BODY
68	(44)	CHARACTER	4	XDEYE03	EYECATCHER
72	(48)	UNSIGNED	1	XDVERS03	VERSION NUMBER
73	(49)	CHARACTER	1	XD03STAT	STATUS
74	(4A)	CHARACTER	2	*	FREE

Cross reference

Name	Hex Offset	Hex Value	Level
XDBDY03	44		2
XDEYE03	44		3
XDKEY03	0		2
XDOCCADID	2		4
XDOCCIA	12		4
XDOCCIAD	12		5
XDOCCIAT	18		5
XDOCC03	2		3
XDREC03	0		1
XDTYP03	0		3
XDVERS03	48		3
XD03STAT	49		3

XD12 - XD HDR RECORD

Name: DCLXD12

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	80	XDREC12	XD HDR RECORD

KEY

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	68	XDKEY12	
0	(0)	CHARACTER	2	XDTYP12	TYPE IS ALWAYS 12

Mandatory pred id to be solved a dummy occ with status M and IA XDTOIA12 exist

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
2	(2)	CHARACTER	16	XDMPREDAD12	MANDATORY PRED ADID
18	(12)	SIGNED	2	XDMPREDOP12	MANDATORY PRED OPNUM

Successor id in CP real existing occopr

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
20	(14)	CHARACTER	16	XDSUCCAD12	SUCCESSOR ADID
36	(24)	SIGNED	2	XDSUCCOP12	SUCCESSOR OPNUM
38	(26)	CHARACTER	10	XDSUCCIA12	SUCCESSOR IA
38	(26)	CHARACTER	6	XDSUCCIAD12	- date
44	(2C)	CHARACTER	4	XDSUCCIAT12	- time

Time
range where real occ must belong to solve this mandatory pending dependency

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
48	(30)	CHARACTER	10	XDFROMIA12	FROM IA allowed values
58	(3A)	CHARACTER	10	XDTOIA12	TO IA allowed values

BODY

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
68	(44)	CHARACTER	12	XDBDY12	HDR RECORD BODY
68	(44)	CHARACTER	4	XDEYE12	EYECATCHER
72	(48)	UNSIGNED	1	XDVERS12	VERSION NUMBER
73	(49)	CHARACTER	1	XDSTAT12	STATUS - W waiting - C completed - D deleted
74	(4A)	UNSIGNED	3	XD12SuccOCC	VSAM OCC num
77	(4D)	CHARACTER	3	*	FREE

Cross reference

Name	Hex Offset	Hex Value	Level
XDBDY12	44		2
XDEYE12	44		3
XDFROMIA12	30		3
XDKEY12	0		2
XDMPREDAD12	2		3
XDMPREDOP12	12		3
XDREC12	0		1
XDSTAT12	49		3
XDSUCCAD12	14		3
XDSUCCIAD12	26		4
XDSUCCIAT12	2C		4
XDSUCCIA12	26		3
XDSUCCOP12	24		3
XDTOIA12	3A		3
XDTYP12	0		3

Name	Hex Offset	Hex Value	Level
XDVERS12	48		3
XD12SuccOCC	4A		3

XD98 - XD Index record

Name : DCLXD98

Function:
This record ....

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	164	XDREC98	XD INDEX RECORD

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	68	XDKEY98	
0	(0)	CHARACTER	2	XDKYP98	TYPE IS ALWAYS 99
2	(2)	CHARACTER	56	XDKALIAS98	XD02 TOK KEY
58	(3A)	CHARACTER	10	XDKF98	FREE

Body:

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
68	(44)	CHARACTER	96	XDBDY98	HDR RECORD BODY
68	(44)	CHARACTER	4	XDEYE98	EYECATCHER
72	(48)	UNSIGNED	1	XDVERS98	VERSION NUMBER
73	(49)	UNSIGNED	3	*	free
76	(4C)	CHARACTER	8	XDDEST98	DESTINATION
84	(54)	CHARACTER	80	*	FREE

Constants

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
2		CHARACTER	98	XDTYP98	first 2 bytes of key
4		CHARACTER	XD98	XDEYEC98	EYEcatcher
1		DECIMAL	1	XDVERSC98	version number

Cross reference

Name	Hex Offset	Hex Value	Level
XDBDY98	44		2
XDDEST98	4C		3
XDEYE98	44		3
XDKALIAS98	2		3
XDKEY98	0		2

Name	Hex Offset	Hex Value	Level
XDKF98	3A		3
XDKYP98	0		3
XDREC98	0		1
XDVERS98	48		3

XD99 - XD Index record

Name : DCLXD99

Function:
This record ....

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	225	XDREC99	XD INDEX RECORD

Key:

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
0	(0)	CHARACTER	68	XDKEY99	
0	(0)	CHARACTER	2	XDKYP99	TYPE IS ALWAYS 99
2	(2)	CHARACTER	8	XDKTOK99	XD02 TOK KEY
10	(A)	CHARACTER	58	XDKF99	FREE

Body:

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
68	(44)	CHARACTER	157	XDBDY99	HDR RECORD BODY
68	(44)	CHARACTER	4	XDEYE99	EYECATCHER
72	(48)	UNSIGNED	1	XDVERS99	VERSION NUMBER
73	(49)	CHARACTER	64	XD99_XD02	XD02 key
73	(49)	CHARACTER	8	XD99DEST	- DESTINATION
81	(51)	CHARACTER	56	XD99ALIAS	- ALIAS
137	(89)	CHARACTER	8	XD99OCCTOK	- OCC TOKEN
145	(91)	CHARACTER	1	XD99STAT	XD02 status
146	(92)	CHARACTER	1	XD99DPSET	'Y' when DP batch changes job status (OPRDPSET=on) NOTE: it is left 'Y' until NMM/EM process it
147	(93)	CHARACTER	78	*	FREE

Constants

Offsets Dec	Hex	Type	Len	Name (Dim)	Description
2		CHARACTER	99	XDTYP99	first 2 bytes of key
4		CHARACTER	XD99	XDEYEC99	EYEcatcher
1		DECIMAL	1	XDVERSC99	version number

Cross reference

Name	Hex Offset	Hex Value	Level
XDBDY99	44		2
XDEYE99	44		3
XDKEY99	0		2
XDKF99	A		3
XDKTOK99	2		3
XDKYP99	0		3
XDREC99	0		1
XDVERS99	48		3
XD99_XD02	49		3
XD99ALIAS	51		4
XD99DEST	49		4
XD99DPSET	92		3

Chapter 6. DB2 tables

These tables are used by the CP History function. The names are aliases used by Tivoli Workload Scheduler for z/OS. Each table description includes information about the table, a description of each of the *key columns*, and a description of each of the *data columns*:

- Key columns are marked like this: **k**. They are sorted in the sequence they appear in the table.
- Data columns follow the last key column and are sorted in alphabetic order.

EQQHIMN

Column name		Data type	Description
HIMNRELEASE	**k**	CHAR(4)	Tivoli Workload Scheduler for z/OS release
HIMNSSNAME	**k**	CHAR(4)	Tivoli Workload Scheduler for z/OS subsystem name
HIMNCURRVERSION		SMALLINT	Current version number of the history information
HIMNPREVVERSION		SMALLINT	Previous version number of the history information
HIMNMAXVERSION		SMALLINT	Maximum version number of the history information

EQQHI14

Column name		Data type	Description
HI14RELEASE	**k**	CHAR(4)	Tivoli Workload Scheduler for z/OS release
HI14SSNAME	**k**	CHAR(4)	Tivoli Workload Scheduler for z/OS subsystem name
HI14VERSION	**k**	SMALLINT	Row version number
HI14APPLID	**k**	CHAR(16)	Application name
HI14IADATE	**k**	CHAR(6)	Occurrence input arrival date
HI14IATIME	**k**	CHAR(4)	Occurrence input arrival time
HI14OPERNR	**k**	SMALLINT	Operation number
HI14EXPDATE		CHAR(8)	Expiry date
HI14WSNAME		CHAR(4)	Workstation name
HI14JOBNAME		CHAR(8)	Job name
HI14REC14		VARCHAR(32000)	The job log record

EQQHI3C

Column name		Data type	Description
HI3CRELEASE	**k**	CHAR(4)	Tivoli Workload Scheduler for z/OS release
HI3CSSNAME	**k**	CHAR(4)	Tivoli Workload Scheduler for z/OS subsystem name
HI3CVERSION	**k**	SMALLINT	Row version number
HI3CAPPLID	**k**	CHAR(16)	Application name
HI3CIADATE	**k**	CHAR(6)	Occurrence input arrival date

Column name		Data type	Description
HI3CIATIME	**k**	CHAR(4)	Occurrence input arrival time
HI3CEXPDATE		CHAR(8)	Expiry date
HI3CREC3C		VARCHAR(32000)	The occurrences record

EQQHI3P

Column name		Data type	Description
HI3CRELEASE	**k**	CHAR(4)	Tivoli Workload Scheduler for z/OS release
HI3CSSNAME	**k**	CHAR(4)	Tivoli Workload Scheduler for z/OS subsystem name
HI3CVERSION	**k**	SMALLINT	Row version number
HI3PAPPLID	**k**	CHAR(16)	Application name
HI3PIADATE	**k**	CHAR(6)	Occurrence input arrival date
HI3PIATIME	**k**	CHAR(4)	Occurrence input arrival time
HI3POPERNR		SMALLINT	Operation number
HI3PEXPDATE		CHAR(8)	Expiry date
HI3PWSNAME		CHAR(4)	Workstation name
HI3PJOBNAME		CHAR(4)	Job name
HI3PREC3P		VARCHAR(32000)	The operation record

Part 3. Appendixes

Appendix A. Abend codes

1102:

Explanation

Undefined feedback from VTAM, or a serious error after a VTAM request that makes continued processing impossible.

Problem determination

See message EQQV015E, which precedes this abend, for diagnostic information. Message EQQV015E is issued only if request parameter list (RPL) feedback is available.

System programmer response

If an undefined feedback code was returned by VTAM, this is a VTAM problem. Call your IBM representative.

1103:

Explanation

VTAM has detected an error while processing a request, or VTAM has detected an error in an associated session, task, or address space.

Problem determination

None.

System programmer response

Call your IBM representative.

1104:

Explanation

Serious error opening the network communication function (NCF) ACB. The VTAM definition of the Tivoli Workload Scheduler for z/OS NCF is probably in error.

Problem determination

See message EQQV027E, which precedes this abend, for diagnostic information.

System programmer response

If the diagnosis points to a user-definition error, correct the error and restart the subsystem. Otherwise, call your IBM representative.

1110:

Explanation

Unrecoverable error when scheduling a VTAM macro.

Problem determination

See message EQQV015E, which precedes this abend, for diagnostic information. Message EQQV015E is issued only if request parameter list (RPL) feedback is available.

System programmer response

Call your IBM representative.

1111:

Explanation

Invalid SCB block or invalid pointer to it (internal error).

Problem determination

None.

System programmer response

Call your IBM representative.

1112:

Explanation

Invalid NAB block or invalid pointer to it (internal error).

Problem determination

None.

System programmer response

Call your IBM representative.

1113:

Explanation

Invalid BIB block or invalid pointer to it (internal error).

Problem determination

None.

System programmer response

Call your IBM representative.

1121:

Explanation

All entries in the node table have been used (internal error).

Problem determination

None.

System programmer response

Call your IBM representative.

1124:

Explanation

Invalid action code (internal error).

Problem determination

None.

System programmer response

Call your IBM representative.

2001:

Explanation

Problem accessing the special resource data space during daily planning, and the VALEACTION keyword of BATCHOPT specifies ABEND.

Problem determination

None.

System programmer response

Take a copy of the old current-plan file, the EQQCXDS file, and the EQQRDDS file, and call your IBM representative.

3900:

Explanation

Not enough storage available to continue processing.

Problem determination

None.

System programmer response

Make sure that the region size specified for the system meets the Tivoli Workload Scheduler for z/OS requirements. Region size requirements are documented in the *Planning and Installation* and in the *Program Directory*. If your region size is too small, enlarge it accordingly, and restart the system. Otherwise, call your IBM representative.

3999:

Explanation

Tivoli Workload Scheduler for z/OS internal validity checking has discovered an error condition (internal error). A message that contains the reason for the abend, as well as other debugging information, is written to the Tivoli Workload Scheduler for z/OS diagnostic file, EQQDUMP.

Problem determination

None.

System programmer response

Call your IBM representative.

Appendix B. Known problems

The following are known open problems.

The following limitations are outstanding:

- "Redundant message in TCP/IP server MLOG for error in MCP"
- "Redundant message in TCP/IP server MLOG for error in AD"
- "Job log not issued if DATASTORE is down" on page 480

Redundant message in TCP/IP server MLOG for error in MCP

A message is erroneously displayed in the MLOG of the TCP/IP server when an error occurs while modifying the current plan.

If an error occurs while modifying the current plan and the following message is issued:

```
AWSUI4064E The following error occurred: EQQM943E USER FIELD NAME ddddd ALREADY
PRESENT FOR APPLEXECUTOR, 001
```

The message is also displayed in the message log of the server as follows:

```
  EQQZ024I Error Diagnosis:
EQQZ024I     START ADDRESS: 0CC35FD4                TIME: 120316 153221
EQQZ024I     000000  C5D8D8D4 F9F4F340                    *EQQM943
EQQZ024I Failing MCP request block, MT0:
EQQZ024I     START ADDRESS: 0CC35FC0                TIME: 120316 153221
EQQZ024I     000000  D4E3F040 F0F144F3 D700F1F2 F0F3F1F6  *MT0 01.3P.12
EQQZ024I     000010  F2F4F0F0 C5D8D8D4 F9F4F340 0000011B  *2400EQQM943
EQQZ024I     000020  00000000 00000000 00C1D7D7 D3C5E7C5  *.........APP
EQQZ024I     000030  C3E4E3D6 D9404040 40F1F2F0 F3F1F6F1  *CUTOR    120
EQQZ024I     000040  F1F4F9F4 F0F0F3F1 F6F1F1F4 F9F4F0F0  *149400316114
EQQZ024I     000050  F3F1F6F2 F3F0F0F5 40404040 08404040  *31623005
EQQZ024I     000060  40404040 40404040 40404040 40404040  *
EQQZ024I     000070  40404040 40404040 40404040 40000000  *
EQQZ024I     000080  00000000 00000000 00000100 01C4C2D7  *............
EQQZ024I     000090  D3E4C740 40E9C3C5 F1404040 40404040  *LUG  ZCE1
EQQZ024I     0000A0  40404040 40404040 40404040 40404040  *
EQQZ024I     0000B0  40000000 01000000 00000000 00000000  * ...........
EQQZ024I     0000C0  00000000 00000000 000000EA 60001E00  *............
EQQZ024I     0000D0  01848484 84844040 40404040 40404040  *.ddddd
EQQZ024I     0000E0  40404040 40404040 40404040 40404040  *
EQQZ024I     0000F0  40404040 40404040 40404040 40404040  *
EQQZ024I     000100  40404040 40404040 40404040 40404040  *
EQQZ024I     000110  40404040 40404000 000000             *       ....
```

The message is erroneously displayed within the trace and should be ignored.

Redundant message in TCP/IP server MLOG for error in AD

A message is unduly displayed in the MLOG of the TCP/IP server when an error occurs while modifying the Application Description database.

If an error occurs while modifying the Application Description database and the following message is issued:

```
The operation on object object_name was not completed because the following error
occurred: EQQA537E AUTOMATION INFO SELECTED FOR OP 1, WORKSTATION NOT FOR AUTOMATION
```

the message is also displayed in the message log of the server as follows:

```
  EQQZ024I Error in block AD       at offset 0138 (msg EQQA537 )
EQQZ024I     START ADDRESS: 0CBDAFE0               TIME: 120327 132050630
EQQZ024I     000000  0000C1D7 D7D3E2C1 F3404040 40404040  *..APPLSA3       *
EQQZ024I     000010  4040C1F9 F9F1F2F3 F1C1C4D9 4001F4F0  *  A991231ADR .40*
EQQZ024I     000020  F0F3F2F7 40404040 40404040 40404040  *0327            *
EQQZ024I     000030  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000040  40404040 D3C1D7D3 C1C3C140 40404040  *    PALLACA     *
EQQZ024I     000050  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000060  40404040 40404040 40404040 00F4F0F0  *            .400*
EQQZ024I     000070  F3F2F7F1 F3F2F0D9 C1C3C6C9 C4F240F5  *3271320RACFID2 5*
EQQZ024I     000080  40404040 40400001 00000000 00000000  *      ..........*
EQQZ024I     000090  000000D0 10C4C5C6 C1E4D3E3 40404040  *.....DEFAULT    *
EQQZ024I     0000A0  40404040 40404040 40404040 40404040  *                *
EQQZ024I     0000B0  40404040 40000000 00000000 00000000  *     ...........*
EQQZ024I     0000C0  00000000 00000000 00000000 00000000  *................*
EQQZ024I     0000D0  00000000 C3D7E4F1 00014040 40404040  *....CPU1..      *
EQQZ024I     0000E0  40404040 40404040 40404040 40404040  *                *
EQQZ024I     0000F0  40400000 00644040 40404040 E8D1D6C2  *  ....      YJOB*
EQQZ024I     000100  F1404040 40400000 40404040 40404040  *1     ..        *
EQQZ024I     000110  40404040 40404040 40404040 40860000  *             f..*
EQQZ024I     000120  00000000 00000001 00000000 00004040  *..............  *
EQQZ024I     000130  D500D540 0000D5D5 01004040 40404040  *N.N ..NN..      *
EQQZ024I     000140  40400000 00000000 00000000 00000000  *  ..............*
EQQZ024I     000150  8489A297 97894040 40404040 40404040  *disppi          *
EQQZ024I     000160  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000170  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000180  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000190  40404040 40404040 40404040 40404040  *                *
EQQZ024I     0001A0  40404040 40404040 40404040 40404040  *                *
EQQZ024I     0001B0  40404040 40404040 40404040 40404040  *                *
EQQZ024I     0001C0  40404040 40404040 40404040 40404040  *                *
EQQZ024I     0001D0  40404040 40404040 40404040 40404040  *                *
EQQZ024I     0001E0  40404040 40404040 40404040 40404040  *                *
EQQZ024I     0001F0  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000200  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000210  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000220  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000230  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000240  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000250  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000260  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000270  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000280  40404040 40404040 40404040 40404040  *                *
EQQZ024I     000290  40404040 40404040 40404040 40404040  *                *
............
```

The message should not be displayed within the trace and should be ignored.

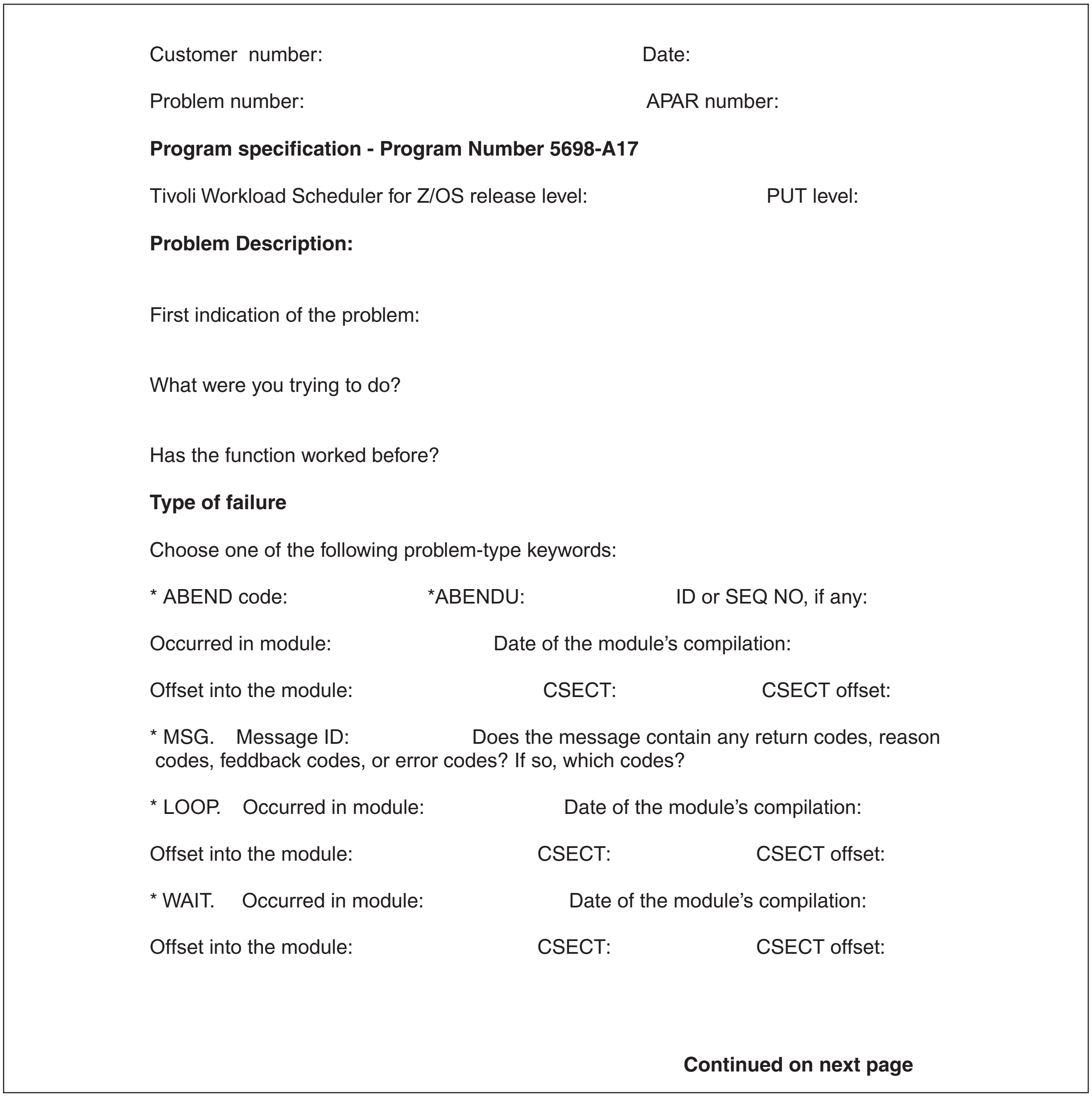
Job log not issued if DATASTORE is down

No job logs are issued while the DATASTORE started task is down.

While the DATASTORE started task is down, no job logs are issued notwithstanding user requests. The following message is recorded in the MLOG of the controller:

```
EQQM938W JOBLOG ABEND    (JOB06207) NOT RETRIEVED. REASON: 0005
```

Appendix C. Problem description sheet



Customer number: Date:

Problem number: APAR number:

Program specification - Program Number 5698-A17

Tivoli Workload Scheduler for Z/OS release level: PUT level:

Problem Description:

First indication of the problem:

What were you trying to do?

Has the function worked before?

Type of failure

Choose one of the following problem-type keywords:

* ABEND code: *ABENDU: ID or SEQ NO, if any:

Occurred in module: Date of the module's compilation:

Offset into the module: CSECT: CSECT offset:

* MSG. Message ID: Does the message contain any return codes, reason codes, feddback codes, or error codes? If so, which codes?

* LOOP. Occurred in module: Date of the module's compilation:

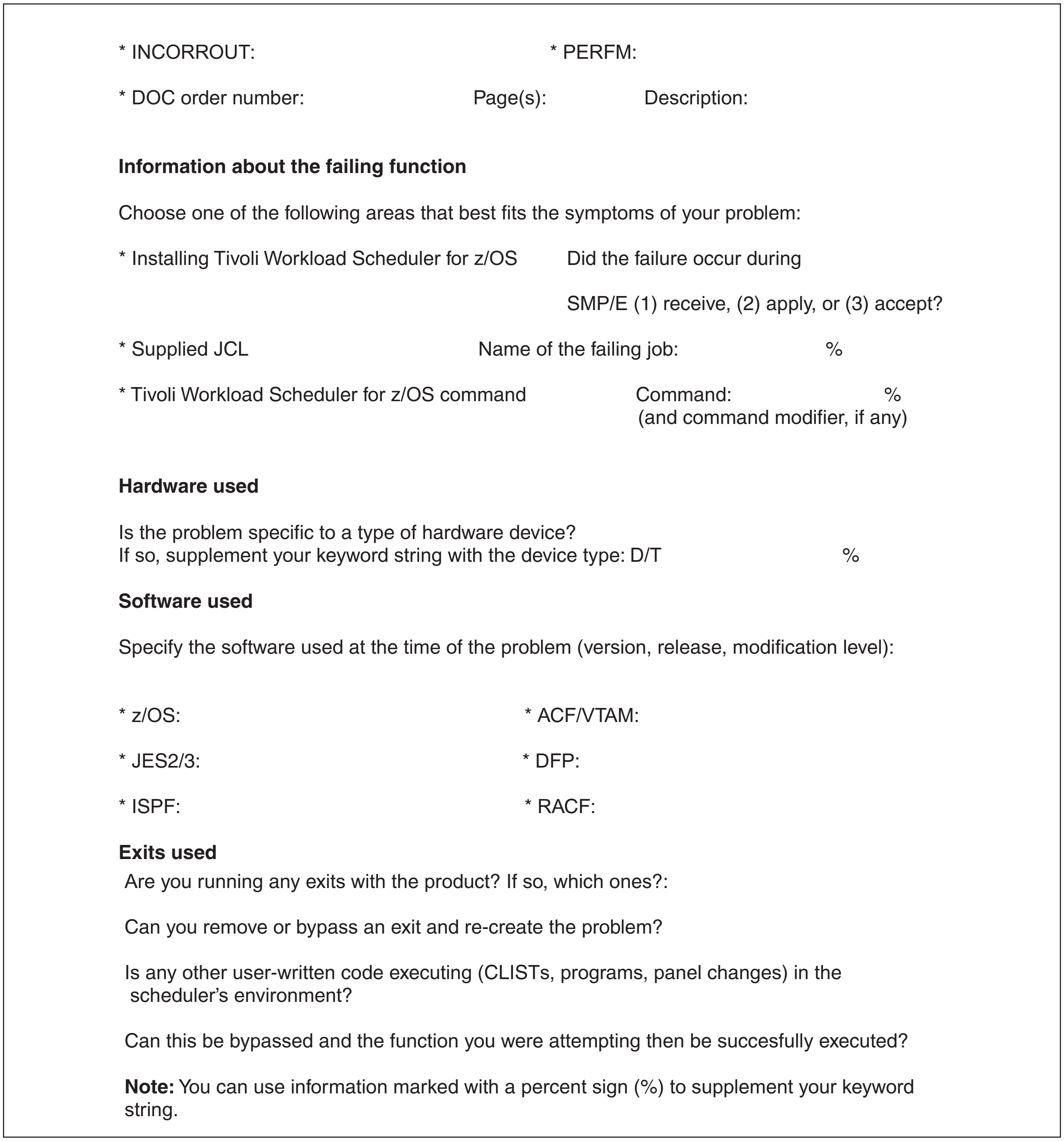
Offset into the module: CSECT: CSECT offset:

* WAIT. Occurred in module: Date of the module's compilation:

Offset into the module: CSECT: CSECT offset:

Continued on next page

Figure 10. Part 1 of 2



* INCORROUT: * PERFM:

* DOC order number: Page(s): Description:

Information about the failing function

Choose one of the following areas that best fits the symptoms of your problem:

* Installing Tivoli Workload Scheduler for z/OS Did the failure occur during

SMP/E (1) receive, (2) apply, or (3) accept?

* Supplied JCL Name of the failing job: %

* Tivoli Workload Scheduler for z/OS command Command: %
(and command modifier, if any)

Hardware used

Is the problem specific to a type of hardware device?
If so, supplement your keyword string with the device type: D/T %

Software used

Specify the software used at the time of the problem (version, release, modification level):

* z/OS: * ACF/VTAM:

* JES2/3: * DFP:

* ISPF: * RACF:

Exits used

Are you running any exits with the product? If so, which ones?:

Can you remove or bypass an exit and re-create the problem?

Is any other user-written code executing (CLISTs, programs, panel changes) in the scheduler's environment?

Can this be bypassed and the function you were attempting then be succesfully executed?

Note: You can use information marked with a percent sign (%) to supplement your keyword string.

Figure 11. Part 2 of 2

Note: You may copy this problem description sheet without payment or royalty provided that each reproduction is done without alteration.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

Product Number: 5698-T08

Printed in USA

SC32-1261-09