CICS Transaction Server for z/OS
5.6

Monitoring Data Reference

Note

Before using this information and the product it supports, read the information in Product Legal Notices.

This edition applies to the IBM® CICS® Transaction Server for z/OS®, Version 5 Release 6 (product number 5655-Y305655-BTA) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this PDF

This PDF is a reference of the monitoring data fields for exception class data, identity class data, transaction resource class data, and system-defined performance class data that are provided to monitor CICS. Before CICS TS V5.4, the information in this PDF was in the *Performance Guide*.

For details of the terms and notation used in this book, see Conventions and terminology used in CICS documentation in IBM Documentation.

Date of this PDF

This PDF was created on 2024-04-22 (Year-Month-Date).

Chapter 1. Monitoring field descriptions

Transaction timing fields

The CMF performance class record provides detailed timing information for each transaction as it is processed by CICS. A transaction can be represented by one or more performance class records, depending on the monitoring options that are selected.

The key transaction timing data fields are as follows:

- The Transaction Start time and Stop time represent the start and end of a transaction measurement interval. This is normally the period between transaction attach and detach, but the performance class record might represent a part of a transaction, depending on the monitoring options selected. To calculate the "Transaction Response Time", subtract the transaction start time from the stop time.
- The Transaction Dispatch time is the time the transaction was dispatched.
- The Transaction Dispatch Wait time is the time the transaction was suspended and waiting for redispatch.
- The Transaction CPU time is the portion of dispatch time when the task is using processor cycles.
- The Transaction Suspend time is the total time the task was suspended. This includes all task suspend (wait) time and includes the following fields:
  - The wait time for redispatch (dispatch wait).
  - The wait time for first dispatch (first dispatch delay). This delay is broken down further into the following fields:

    First dispatch delay caused by TRANCLASS limits
    First dispatch delay caused by MXT limits
  - The total I/O wait and other wait times.

The CMF performance class record also provides a more detailed breakdown of the transaction suspend (wait) time into separate data fields. These comprise the following fields:

Table 1. Performance class wait (suspend) fields

Group Name	Field ID	Field Name	Description
DFHTERM	009	TCIOWTT	Terminal I/O wait time
DFHJOUR	010	JCIOWTT	Journal I/O wait time
DFHTEMP	011	TSIOWTT	Temporary storage I/O wait time
DFHFILE	063	FCIOWTT	File I/O wait time
DFHTERM	100	IRIOWTT	Interregion I/O wait time
DFHDEST	101	TDIOWTT	Transient data I/O wait time
DFHTASK	123	GNQDELAY	Global ENQ delay time
DFHTASK	128	LMDELAY	Lock Manager delay time
DFHTASK	129	ENQDELAY	Local ENQ delay time
DFHTERM	133	LU61WTT	LU 6.1 I/O wait time
DFHTERM	134	LU62WTT	LU 6.2 I/O wait time
DFHFEPI	156	SZWAIT	FEPI suspend time

Table 1. Performance class wait (suspend) fields (continued)			
Group Name	**Field ID**	**Field Name**	**Description**
DFHTASK	171	RMISUSP	Resource manager interface (RMI) suspend time
DFHFILE	174	RLSWAIT	RLS File I/O wait time
DFHFILE	176	CFDTWAIT	Coupling Facility data tables server I/O wait time
DFHSYNC	177	SRVSYWTT	Coupling Facility data tables server sync point and resynchronization wait time
DFHTEMP	178	TSSHWAIT	Shared temporary storage I/O wait time
DFHTASK	181	WTEXWAIT	**EXEC CICS WAIT EXTERNAL** wait time
DFHTASK	182	WTCEWAIT	**EXEC CICS WAITCICS** and **WAIT EVENT** wait time
DFHTASK	183	ICDELAY	Interval Control delay time
DFHTASK	184	GVUPWAIT	Dispatchable Wait wait time
DFHDATA	186	IMSWAIT	IMS DBCTL wait time
DFHDATA	187	DB2RDYQW	Db2® ready queue wait time
DFHDATA	188	DB2CONWT	Db2 connection time
DFHTASK	191	RRMSWAIT	RRMS/MVS™ indoubt wait time
DFHTASK	192	RQRWAIT	Request Receiver wait time
DFHTASK	193	RQPWAIT	Request Processor wait time
DFHTASK	195	RUNTRWTT	CICS BTS run process/activity synchronous wait time
DFHSYNC	196	SYNCDLY	Sync point delay time
DFHSOCK	241	SOIOWTT	Socket I/O wait time
DFHTASK	247	DSCHMDLY	CICS change TCB mode delay time
DFHTASK	250	MAXOTDLY	CICS L8 and L9 mode open TCB delay time
DFHTASK	254	JVMSUSP	JVM suspend time Be aware of the likelihood of double accounting that might be recorded in JVMSUSP. For details, see "JVM elapsed time and suspend time" on page 9.
DFHTASK	268	DSTCBMWT	TCB mismatch wait time
DFHTASK	274	DSMMSCWT	MVS user region or extended user region storage constraint wait time
DFHTASK	279	DSMMSCWT	MVS storage constraint wait time
DFHTASK	281	MAXSTDLY	CICS SSL TCB delay time
DFHTASK	282	MAXXTDLY	CICS XP TCB delay time
DFHTASK	283	MAXTTDLY	CICS JVM server thread TCB delay time
DFHTASK	285	PTPWAIT	3270 bridge partner wait time
DFHSOCK	299	SOOIOWTT	MAXSOCKETS wait time
DFHSOCK	300	ISIOWTT	IS I/O wait time
DFHSOCK	319	ISALWTT	IPIC session allocation wait time

Table 1. Performance class wait (suspend) fields (continued)			
Group Name	**Field ID**	**Field Name**	**Description**
DFHTERM	343	TCALWTT	MRO, LU6.1, and LU6.2 session allocation wait time
DFHDATA	396	WMQGETWT	MQ GETWAIT wait time
DFHTASK	401	JVMTHDWT	JVM server thread wait time. This does not apply to Liberty JVM servers.
DFHDEST	403	TDILWTT	Transient Data intrapartition lock wait time
DFHDEST	404	TDELWTT	Transient Data extrapartition lock wait time
DFHFILE	426	FCXCWTT	File control wait time for exclusive control of a VSAM control interval
DFHFILE	427	FCVSWTT	File control wait time for a VSAM string
DFHTASK	429	DSAPTHWT	Dispatcher allocate pthread wait time
DFHTASK	475	ASFTCHWT	**EXEC CICS FETCH** wait time
DFHTASK	476	ASRNATWT	**EXEC CICS RUN TRANSID** attach wait time

Transaction response time

You can calculate the internal CICS response time by subtracting performance data field 005 (start time) from performance data field 006 (stop time).

Figure 1 on page 3 shows the relationship of dispatch time, suspend time, and CPU time with the response time.

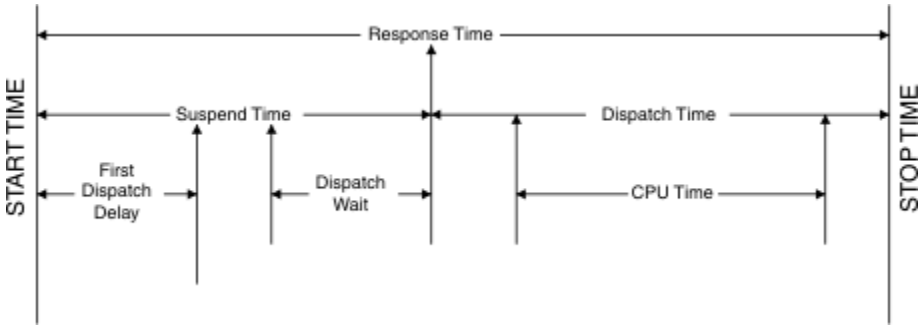

Figure 1. Response time relationships

Transaction dispatch time and CPU time

The transaction total dispatch time field USRDISPT, field 007 in group DFHTASK, is the total elapsed time during which the user task was dispatched by the CICS dispatcher domain on each CICS TCB under which the task executed.

The transaction total CPU time field USRCPUT, field 008 in group DFHTASK, is the total processor time during which the user task was dispatched by the CICS dispatcher domain on each CICS TCB under which the task executed. Where a task executed on a specialty processor, a fraction of this field may be normalized to reflect the relative speed of the general processor compared with the specialty processors. See Relating CICS transactions to hardware resources for a description of the normalization process.

For both these fields, the time recorded in the field can be associated with any of the TCB modes which are managed by the CICS dispatcher in the current CICS release. These include open TCBs, such as L8 mode TCBs, as well as non-open TCBs, such as the QR TCB. Be aware that for each CICS release, new TCB modes might be added or obsolete TCB modes might be removed, particularly in the case of the open TCB modes. You should always check the performance data field descriptions in the current release

documentation to see which TCB modes are applicable. The field descriptions are listed in Performance data in group DFHTASK.

If you want to calculate a transaction's ratio of accumulated CPU time to accumulated dispatch time (CPU/DISP ratio) for the QR TCB, use fields 255 (QRDISPT) and 256 (QRCPUT) in group DFHTASK. These fields show the elapsed time and processor time during which the user task was dispatched on the QR TCB only.

The CPU/DISP ratio for an individual task should always be considered in the context of other activity in the CICS region. The Dispatcher TCB Modes report (see Dispatcher domain: TCB Modes report) which is provided by the sample statistics program DFH0STAT includes a calculation of the CPU/DISP ratio for the QR TCB for the whole CICS region.

Transaction wait (suspend) times

The CMF performance class record provides a breakdown of the transaction suspend (wait) time into separate data fields. You can use these to calculate various wait times.

The performance data fields listed in Table 2 on page 4 record the elapsed time spent waiting for specific types of I/O operation. For example, field 009 records the elapsed time waiting for terminal I/O.

The elapsed time includes the time for the I/O operation, the time for the access method to complete the outstanding event control block, and the subsequent time until the waiting CICS transaction is redispatched.

Table 2. Performance class wait (suspend) fields

Group Name	Field ID	Field Name	Description
DFHTERM	009	TCIOWTT	Terminal I/O wait time
DFHJOUR	010	JCIOWTT	Journal I/O wait time
DFHTEMP	011	TSIOWTT	Temporary storage I/O wait time
DFHFILE	063	FCIOWTT	File I/O wait time
DFHTERM	100	IRIOWTT	Interregion I/O wait time
DFHDEST	101	TDIOWTT	Transient data I/O wait time
DFHTASK	123	GNQDELAY	Global ENQ delay time
DFHTASK	128	LMDELAY	Lock Manager delay time
DFHTASK	129	ENQDELAY	Local ENQ delay time
DFHTERM	133	LU61WTT	LU 6.1 I/O wait time
DFHTERM	134	LU62WTT	LU 6.2 I/O wait time
DFHFEPI	156	SZWAIT	FEPI suspend time
DFHTASK	171	RMISUSP	Resource manager interface (RMI) suspend time
DFHFILE	174	RLSWAIT	RLS File I/O wait time
DFHFILE	176	CFDTWAIT	Coupling Facility data tables server I/O wait time
DFHSYNC	177	SRVSYWTT	Coupling Facility data tables server sync point and resynchronization wait time
DFHTEMP	178	TSSHWAIT	Shared temporary storage I/O wait time
DFHTASK	181	WTEXWAIT	**EXEC CICS WAIT EXTERNAL** wait time
DFHTASK	182	WTCEWAIT	**EXEC CICS WAITCICS** and **WAIT EVENT** wait time

Table 2. Performance class wait (suspend) fields (continued)

Group Name	Field ID	Field Name	Description
DFHTASK	183	ICDELAY	Interval Control delay time
DFHTASK	184	GVUPWAIT	Dispatchable Wait wait time
DFHDATA	186	IMSWAIT	IMS DBCTL wait time
DFHDATA	187	DB2RDYQW	Db2 ready queue wait time
DFHDATA	188	DB2CONWT	Db2 connection time
DFHTASK	191	RRMSWAIT	RRMS/MVS indoubt wait time
DFHTASK	192	RQRWAIT	Request Receiver wait time
DFHTASK	193	RQPWAIT	Request Processor wait time
DFHTASK	195	RUNTRWTT	CICS BTS run process/activity synchronous wait time
DFHSYNC	196	SYNCDLY	Sync point delay time
DFHSOCK	241	SOIOWTT	Socket I/O wait time
DFHTASK	247	DSCHMDLY	CICS change TCB mode delay time
DFHTASK	250	MAXOTDLY	CICS L8 and L9 mode open TCB delay time
DFHTASK	254	JVMSUSP	JVM suspend time Be aware of the likelihood of double accounting that might be recorded in JVMSUSP. For details, see “JVM elapsed time and suspend time” on page 9.
DFHTASK	268	DSTCBMWT	TCB mismatch wait time
DFHTASK	274	DSMMSCWT	MVS user region or extended user region storage constraint wait time
DFHTASK	279	DSMMSCWT	MVS storage constraint wait time
DFHTASK	281	MAXSTDLY	CICS SSL TCB delay time
DFHTASK	282	MAXXTDLY	CICS XP TCB delay time
DFHTASK	283	MAXTTDLY	CICS JVM server thread TCB delay time
DFHTASK	285	PTPWAIT	3270 bridge partner wait time
DFHSOCK	299	SOOIOWTT	MAXSOCKETS wait time
DFHSOCK	300	ISIOWTT	IS I/O wait time
DFHSOCK	319	ISALWTT	IPIC session allocation wait time
DFHTERM	343	TCALWTT	MRO, LU6.1, and LU6.2 session allocation wait time
DFHDATA	396	WMQGETWT	MQ GETWAIT wait time
DFHTASK	401	JVMTHDWT	JVM server thread wait time. This does not apply to Liberty JVM servers.
DFHDEST	403	TDILWTT	Transient Data intrapartition lock wait time
DFHDEST	404	TDELWTT	Transient Data extrapartition lock wait time
DFHFILE	426	FCXCWTT	File control wait time for exclusive control of a VSAM control interval

Table 2. Performance class wait (suspend) fields (continued)			
Group Name	**Field ID**	**Field Name**	**Description**
DFHFILE	427	FCVSWTT	File control wait time for a VSAM string
DFHTASK	429	DSAPTHWT	Dispatcher allocate pthread wait time
DFHTASK	475	ASFTCHWT	**EXEC CICS FETCH** wait time
DFHTASK	476	ASRNATWT	**EXEC CICS RUN TRANSID** attach wait time

Figure 2 on page 6 shows an example of the relationship between a typical transaction wait time field, and the transaction's suspend time, dispatch time, processor, and dispatch wait time fields. The transaction has a period of suspend time, which is the time between two periods of dispatch and processor time. The period of suspend time is equal to the total of all the relevant wait times. The period of suspend time includes the dispatch wait, which ends when the suspend time ends and the dispatch and processor time starts.

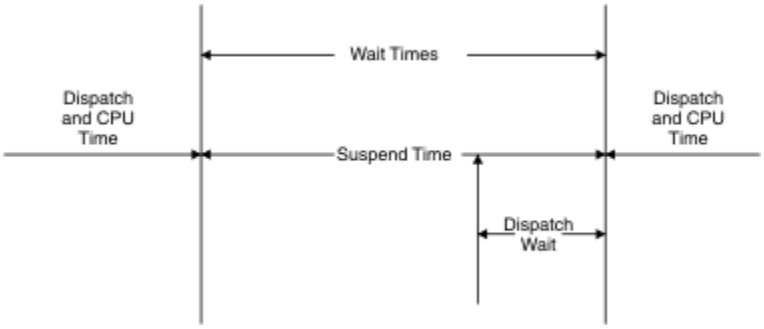

Figure 2. Wait (suspend) time relationships

You can use the CMF suspend time and wait time measurements to perform accurate calculations on the suspend time. For example, to calculate the total I/O wait time, add the values of the fields in the following list:

- Terminal control I/O wait
- Temporary storage I/O wait
- Shared temporary storage I/O wait
- Transient data I/O wait
- Journal (z/OS logger) I/O wait
- File control I/O wait
- RLS file I/O wait
- Coupling Facility data table I/O wait
- Inbound Socket I/O wait
- IS I/O wait time
- Outbound Socket I/O wait
- Interregion (MRO) I/O wait
- LU 6.1 TC I/O wait
- LU 6.2 TC I/O wait
- FEPI I/O wait

To calculate the total other wait time, add the values of the fields in the following list:

- First dispatch delay. This field includes the MXT and TRANCLASS first dispatch delay fields.
- Local ENQ delay

- Global ENQ delay
- Interval control delay
- Lock manager delay
- Wait external wait
- **EXEC CICS WAITCICS** and **EXEC CICS WAIT EVENT** wait
- CICS BTS run synchronous wait
- CFDT server synchronous wait
- Request Receiver wait time
- Request Processor wait time
- Syncpoint delay time
- CICS L8 and L9 mode open TCB delay time
- CICS SSL TCB delay time
- CICS JVM server thread TCB delay time
- CICS XP TCB delay time
- CICS change-TCB mode delay time
- RRMS/MVS wait
- 3270 bridge partner wait
- TCB mismatch wait time
- JVM server thread wait time
- MVS storage constraint wait time
- Intrapartition transient data lock wait time
- Extrapartition transient data lock wait time
- File control wait time for exclusive control of a VSAM control interval
- File control wait time for a VSAM string
- IPIC session allocation wait time
- MRO, LU6.1, and LU6.2 session allocation wait time
- Dispatchable waits wait
- Dispatcher allocate pthread wait time
- **EXEC CICS FETCH** wait time
- **EXEC CICS RUN TRANSID** attach wait time
- MVS user region or extended user region storage constraint wait time

Note: Do not include the redispatch wait time (DISPWTT) in the calculation of total other wait time as the elapsed times listed in Table 2 on page 4 already include the time to complete the outstanding event control block and the subsequent time until the waiting CICS transaction is redispatched.

Very occasionally, the total task suspend time is greater than the sum of the component suspend times. This unaccounted time is identified as *uncaptured wait time*. Uncaptured wait time occurs because CICS does not have a separate monitoring field for every possible type of suspend. To determine the uncaptured wait time, use the following calculation:

```
Uncaptured wait time =  (Suspend - (total I/O wait time + total other wait time))
```

The CMF performance class data also provides the following important transaction timing measurements:

- The Program load time is the program fetch time (dispatch time) for programs invoked by the transaction. See “Program load time” on page 8.
- The Exception wait time is the accumulated time from the exception conditions as measured by the CMF exception class records. For more information, see Exception class data: Listing of data fields .

- The RMI elapsed time is the elapsed time the transaction spent in all Resource Managers invoked by the transaction using the Resource Manager Interface (RMI). See “RMI elapsed and suspend time” on page 8.
- The JVM elapsed time is the elapsed time the transaction spent in the Java Virtual Machine (JVM) for the Java programs invoked by the transaction. See “JVM elapsed time and suspend time” on page 9.
- The JVM initialization elapsed time is the elapsed time the transaction spent initializing the Java Virtual Machine (JVM) environment for all the Java programs invoked by the transaction.
- The Syncpoint elapsed time is the elapsed time the transaction spent processing a sync point. See “Syncpoint elapsed time” on page 9.

Program load time

The program load time is the program fetch time (dispatch time) for programs invoked by a transaction.

Figure 3 on page 8 shows the relationship between the program load time (field id 115 in group DFHPROG) and the dispatch time and the suspend time (field ids 7 and 14 in group DFHTASK).

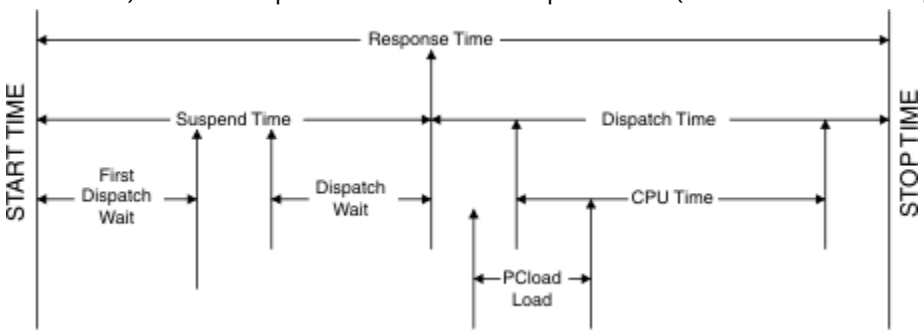

Figure 3. Program load time

The response time of the transaction is the total time from the transaction start time to the transaction stop time. The response time can be subdivided into the following two periods:

- Suspend time.

  The suspend time includes the first dispatch delay, which begins at the transaction start time and ends partway into the suspend time. The suspend time also includes the dispatch wait, which begins further on into the suspend time, and ends when the suspend time ends and the dispatch time begins.
- Dispatch time.

  The dispatch time includes the CPU time, which begins some time after the start of the dispatch time, and ends some time before the dispatch time ends. In this figure, the dispatch time also includes the program load time. The program load time begins after the start of the dispatch time, and overlaps with the first part of the CPU time.

RMI elapsed and suspend time

The RMI elapsed time and suspend time fields provide an insight into the amount of time that a transaction spends in the CICS resource manager interface (RMI).

Figure 4 on page 9 shows the relationship between the RMI elapsed time (field id 170 in group DFHTASK) and the suspend time (field id 171 in group DFHTASK).

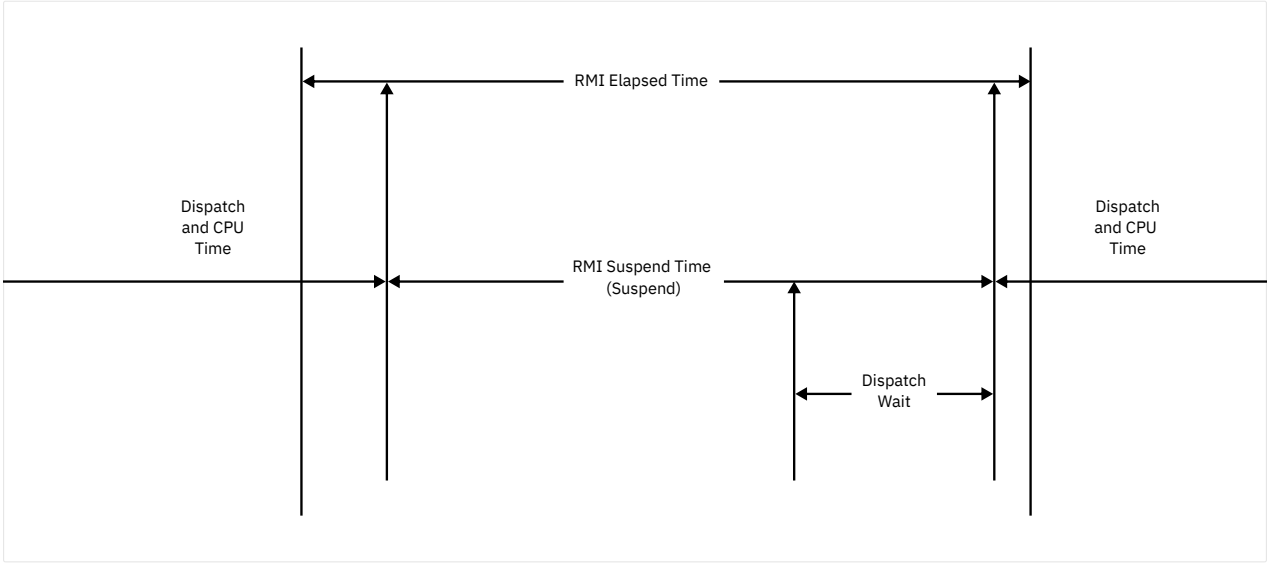

Figure 4. RMI elapsed and suspend time

The RMI elapsed time includes part of a period of dispatch and CPU time at the start. This period of time began before the RMI elapsed time. When the dispatch and CPU time ends, the RMI suspend time begins. The RMI suspend time includes a dispatch wait at the end. When the dispatch wait ends, another period of dispatch and CPU time begins. Shortly afterwards, the RMI elapsed time ends, while the dispatch and CPU time continues. The RMI elapsed time therefore includes parts of two periods of dispatch and CPU time, with an intervening period of RMI suspend time.

The Db2 wait, the Db2 connection wait, and the Db2 readyq wait time fields, as well as the IMS wait and MQ GETWAIT wait time fields are included in the RMI suspend time.

JVM elapsed time and suspend time

The JVM elapsed and suspend time fields provide an insight into the amount of time that a transaction spends in a Java Virtual Machine (JVM).

JVMTIME and JVMSUSP fields

Care must be taken when using the JVM elapsed time field JVMTIME (group name DFHTASK, field id: 253) and JVM suspend time field JVMSUSP (group name DFHTASK, field id: 254) in any calculation with other CMF timing fields. This is because of the likelihood of double accounting other CMF timing fields in the performance class record within the JVM time fields. For example, if a Java application program invoked by a transaction issues a read file (non-RLS) request using the Java API for CICS (JCICS) classes, the file I/O wait time will be included in both the file I/O wait time field (group name DFHFILE, field id: 063), and the transaction suspend time field (group name DFHTASK, field id: 014), as well as the JVM suspend time field.

JCICS requests

The number of Java API for CICS (JCICS) requests issued by the user task is included in the CICS OO foundation class request count field (group name: DFHCICS, field id: 025).

Syncpoint elapsed time

The syncpoint elapsed time is the elapsed time the transaction spent processing a syncpoint.

Figure 5 on page 10 shows the relationship between the syncpoint elapsed time (field 173 in group DFHSYNC) and the suspend time (field 14 in group DFHTASK).

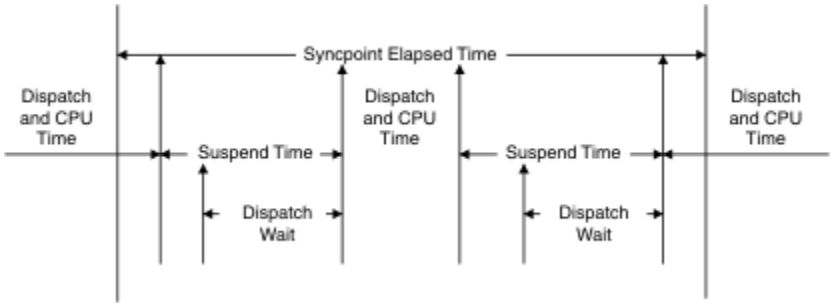

Figure 5. Syncpoint elapsed time

The syncpoint elapsed time begins during a period of dispatch and CPU time. A period of suspend time follows, which includes a dispatch wait at the end. When the dispatch wait and the suspend time end, there is another period of dispatch and CPU time. When this period ends, another period of suspend time begins, which includes another dispatch wait. When the dispatch wait and the suspend time end, another period of dispatch and CPU time begins. Shortly afterward, the syncpoint elapsed time ends, while the period of dispatch and CPU time continues. Therefore, the syncpoint elapsed time in this example includes two complete periods of suspend time.

Storage occupancy counts

An occupancy count measures the area under the curve of user-task storage in use against elapsed time.

The unit of measure is the “byte-unit”, where the “unit” is equal to 1024 microseconds, or 1.024 milliseconds. Where *ms* is milliseconds, a user task occupying, for example, 256 bytes for 125 milliseconds, is measured as follows:

125 / 1.024 *ms* = 122 units * 256 = 31 232 byte-units.

Note: All references to “Start time” and “Stop time” in the following calculations refer to the middle 4 bytes of each 8 byte start/stop time field. Bit 47 of Start time or Stop time represents a unit of 16 microseconds.

To calculate response time and convert into microsecond units:
Response = ((Stop time - Start time) * 16)
To calculate number of 1024 microsecond “units”:
Units = (Response / 1024)
or
Units = ((Stop time - Start time) / 64)
To calculate the average user-task storage used from the storage occupancy count:
Average user-task storage used = (Storage Occupancy / Units)

To calculate units per second:
Units Per Second = (1 000 000 / 1024) = 976.5625
To calculate the response time in seconds:
Response time = (((Stop time - Start time) * 16) / 1 000 000)

During the life of a user task, CICS measures, calculates, and accumulates the storage occupancy at the following points:

- Before GETMAIN increases current user-storage values
- Before FREEMAIN reduces current user-storage values
- Just before the performance record is moved to the buffer.

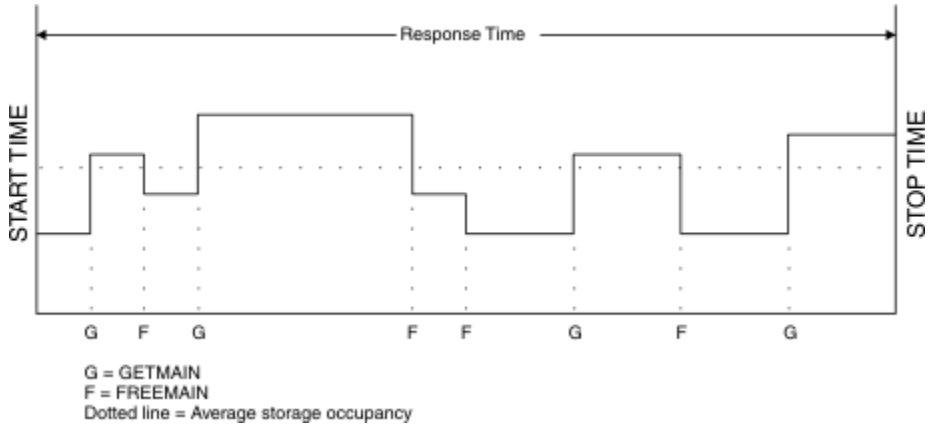

Figure 6. Storage occupancy

Program storage

The level of program storage that is in use is incremented at LOAD, LINK, and XCTL events by the size (in bytes) of the referenced program, and is decremented at RELEASE or RETURN events. On an XCTL event, the program storage currently in use is also decremented by the size of the program issuing the XCTL, because the program is no longer required.

Figure 7 on page 12 shows the relationships between the high-water mark data fields that show the maximum amounts of program storage in use by the user task. The PCSTGHWM field (id 087) shows the maximum amount of program storage in use by the task both above and below 16 MB. The PC31AHWM (139) and PC24BHWM (108) fields are subsets of PCSTGHWM, and show the maximum amounts in use above and below 16M, respectively. Further subset fields show the maximum amounts of storage in use by the task in each of the CICS dynamic storage areas (DSAs).

Note:

1. The total of the values for all the subsets in a superset might not be equal to the value for the superset. For example, the value of PC31AHWM plus the value of PC24BHWM might not be the value of PCSTGHWM. This is because the peaks in the different types of program storage acquired by the user task do not necessarily occur simultaneously.
2. If a task loads the same program several times, the program storage data fields might not reflect the true high-water mark of program storage used by the task. The fields are incremented each time the LOAD command is issued, but if the program has already been loaded by the task, the existing copy of the program is used, meaning that only one copy of the program exists in storage. Because of this, for tasks that repeatedly load the same program, the data in the fields PCSTGHWM, PC24BHWM, PC31RHWM, PC31AHWM, PC31CHWM, PC24CHWM, PC24SHWM, PC31SHWM and PC24RHWM should be used with caution.
3. When you disable an exit program, the program can be released, and the program storage fields might be decremented. If the program was not linked to within the lifetime of the current task, then the program storage fields are decremented unnecessarily, which means you might not get a true value and should proceed with caution.

The high-water mark fields and program storage fields are described in detail in Performance data in group DFHSTOR.

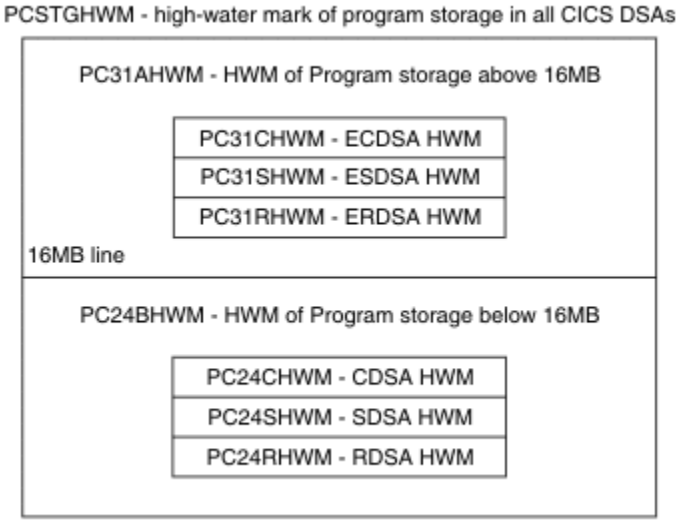

Figure 7. Relationships between the high-water mark program storage data fields

Chapter 2. Performance class data: listing of data fields

The performance class data is listed in this section in order of group name. The group name is always in field CMODNAME of the dictionary entry.

A user task can be represented by one or more performance class monitoring records, depending on whether the MCT event monitoring option DELIVER or the system initialization parameters MNCONV=YES or MNSYNC=YES have been selected. In the descriptions that follow, the term *user task* means that part or whole of a transaction that is represented by a performance class record, unless the description states otherwise.

To understand the format of the performance data section, see Performance data sections.

Performance data in group DFHCBTS

Descriptions of the performance data fields in the DFHCBTS group, including the numeric identifier, type, and size of each field.

200 (TYPE-C, 'PRCSNAME', 36 BYTES)
The name of the CICS business transaction service (BTS) process of which the user task formed part.

201 (TYPE-C, 'PRCSTYPE', 8 BYTES)
The process-type of the CICS BTS process of which the user task formed part.

202 (TYPE-C, 'PRCSID', 52 BYTES)
The CICS-assigned identifier of the CICS BTS root activity that the user task implemented.

203 (TYPE-C, 'ACTVTYID', 52 BYTES)
The CICS-assigned identifier of the CICS BTS activity that the user task implemented.

204 (TYPE-C, 'ACTVTYNM', 16 BYTES)
The name of the CICS BTS activity that the user task implemented.

205 (TYPE-A, 'BARSYNCT', 4 BYTES)
The number of CICS BTS run process, or run activity, requests that the user task made in order to execute a process or activity synchronously.

206 (TYPE-A, 'BARASYCT', 4 BYTES)
The number of CICS BTS run process, or run activity, requests that the user task made in order to execute a process or activity asynchronously.

207 (Type-A, 'BALKPACT', 4 BYTES)
The number of CICS BTS link process, or link activity, requests that the user task issued.

208 (TYPE-A, 'BADPROCT', 4 BYTES)
The number of CICS BTS define process requests issued by the user task.

209 (TYPE-A, 'BADACTCT', 4 BYTES)
The number of CICS BTS define activity requests issued by the user task.

210 (TYPE-A, 'BARSPACT', 4 BYTES)
The number of CICS BTS reset process and reset activity requests issued by the user task.

211 (TYPE-A, 'BASUPACT', 4 BYTES)
The number of CICS BTS suspend process, or suspend activity, requests issued by the user task.

212 (TYPE-A, 'BARMPACT', 4 BYTES)
The number of CICS BTS resume process, or resume activity, requests issued by the user task.

213 (TYPE-A, 'BADCPACT', 4 BYTES)
The number of CICS BTS delete activity, cancel process, or cancel activity, requests issued by the user task.

214 (TYPE-A, 'BAACQPCT', 4 BYTES)
The number of CICS BTS acquire process, or acquire activity, requests issued by the user task.

215 (Type-A, 'BATOTPCT', 4 BYTES)
Total number of CICS BTS process and activity requests issued by the user task.

216 (TYPE-A, 'BAPRDCCT', 4 BYTES)
The number of CICS BTS delete, get, move, or put, container requests for process data containers issued by the user task.

217 (TYPE-A, 'BAACDCCT', 4 BYTES)
The number of CICS BTS delete, get, move, or put, container requests for current activity data containers issued by the user task.

218 (Type-A, 'BATOTCCT', 4 BYTES)
Total number of CICS BTS delete, get, move, or put, process container and activity container requests issued by the user task.

219 (TYPE-A, 'BARATECT', 4 BYTES)
The number of CICS BTS retrieve-reattach event requests issued by the user task.

220 (TYPE-A, 'BADFIECT', 4 BYTES)
The number of CICS BTS define-input event requests issued by the user task.

221 (TYPE-A, 'BATIAECT', 4 BYTES)
The number of CICS BTS DEFINE TIMER EVENT, CHECK TIMER EVENT, DELETE TIMER EVENT, and FORCE TIMER EVENT requests issued by the user task.

222 (TYPE-A, 'BATOTECT', 4 BYTES)
Total number of CICS BTS event-related requests issued by the user task.

Performance data in group DFHCHNL

Descriptions of the performance data fields in the DFHCHNL group, including the numeric identifier, type, and size of each field.

321 (TYPE-A, 'PGTOTCCT', 4 BYTES)
The number of CICS requests for channel containers issued by the user task.

322 (TYPE-A, 'PGBRWCCT', 4 BYTES)
The number of CICS browse requests for channel containers issued by the user task.

323 (TYPE-A, 'PGGETCCT', 4 BYTES)
The number of GET CONTAINER and GET64 CONTAINER requests for channel containers issued by the user task.

324 (TYPE-A, 'PGPUTCCT', 4 BYTES)
The number of PUT CONTAINER and PUT64 CONTAINER requests for channel containers issued by the user task.

325 (TYPE-A, 'PGMOVCCT', 4 BYTES)
The number of MOVE CONTAINER requests for channel containers issued by the user task.

326 (TYPE-A, 'PGGETCDL', 4 BYTES)
The total length, in bytes, of the data in the containers of all the GET CONTAINER CHANNEL and GET64 CONTAINER CHANNEL commands issued by the user task.

327 (TYPE-A, 'PGPUTCDL', 4 BYTES)
The total length, in bytes, of the data in the containers of all the PUT CONTAINER CHANNEL and PUT64 CONTAINER CHANNEL commands issued by the user task.

328 (TYPE-A, 'PGCRECCT', 4 BYTES)
The number of containers created by MOVE, PUT CONTAINER, and PUT64 CONTAINER requests for channel containers issued by the user task.

329 (TYPE-A, 'PGCSTHWM', 4 BYTES)
Maximum amount (high-water mark), in bytes, of container storage allocated to the user task.

Performance data in group DFHCICS

Descriptions of the performance data fields in the DFHCICS group, including the numeric identifier, type, and size of each field.

005 (TYPE-T, 'START', 8 BYTES)
Start time of measurement interval, which is one of the following times:

- The time at which the user task was attached.
- The time at which data recording was most recently reset in support of the MCT user event monitoring point DELIVER option or the monitoring options MNCONV, MNSYNC, or FREQUENCY.

For more information, see Clocks and time stamps.

Note: Response time = STOP - START. For more information, see Transaction response time.

006 (TYPE-T, 'STOP', 8 BYTES)
Finish time of measurement interval, which is one of the following times:

- The time at which the user task was detached.
- the time at which data recording was completed in support of the MCT user event monitoring point DELIVER option or the monitoring options MNCONV, MNSYNC, or FREQUENCY.

For more information, see Clocks and time stamps.

Note: Response time = STOP - START. For more information, see Transaction response time.

025 (TYPE-A, 'CFCAPICT', 4 BYTES)
Number of CICS OO foundation class requests, including the Java API for CICS (JCICS) classes, issued by the user task.

089 (TYPE-C, 'USERID', 8 BYTES)
User identification at task creation. This identification can also be the remote user identifier for a task created as the result of receiving an ATTACH request across an MRO or APPC link with attach-time security enabled.

Note: In Liberty, when using the CICS security feature, the user ID is established at a time later than **`Task attach`**. The user ID value reflects the final user ID value used in secure Liberty transactions.

103 (TYPE-S, 'EXWTTIME', 12 BYTES)
Accumulated data for exception conditions. The timer component of the clock contains the total elapsed time for which the user waited on exception conditions. The period count equals the number of exception conditions that have occurred for this task. For more information on exception conditions, see Exception class data: Listing of data fields. For more information on clocks, see Clocks and time stamps.

Note: The performance class data field 'EXWTTIME' is updated when exception conditions are encountered even when the exception class is inactive.

112 (TYPE-C, 'RTYPE', 4 BYTES)
Performance record type (low-order byte-3):

C
Record output for a terminal converse

D
Record output for a user EMP DELIVER request

F
Record output for a long-running transaction

S
Record output for a sync point

T
Record output for the end of a task.

130 (TYPE-C, 'RSYSID', 4 BYTES)
The name (SYSID) of the remote system to which this transaction was routed either statically or dynamically.

This field also includes the connection name (SYSID) of the remote system to which this transaction was routed when using the CRTE routing transaction. The field is null for those CRTE transactions that establish or cancel the transaction routing session.

Note: If the transaction was not routed or was routed locally, this field is set to null. Also see the program name (field 71).

131 (TYPE-A, 'PERRECNT', 4 BYTES)
The number of performance class records written by the CICS Monitoring Facility (CMF) for the user task.

167 (TYPE-C, 'SRVCLASS', 8 BYTES)
The z/OS Workload Manager (WLM) service class for this transaction. This field is null if no transaction classification rules are defined for CICS subsystems in the active z/OS Workload Manager (WLM) service policy, or if the transaction was WLM-classified in another CICS region.

168 (TYPE-C, 'RPTCLASS', 8 BYTES)
The z/OS Workload Manager (WLM) report class for this transaction. This field is null if no transaction classification rules are defined for CICS subsystems in the active z/OS Workload Manager (WLM) service policy, or if the transaction was WLM-classified in another CICS region.

351 (TYPE-C, 'OADID', 64 BYTES)
The adapter identifier added to the origin data by the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.

352 (TYPE-C, 'OADATA1', 64 BYTES)
The data added to the origin data by the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.

353 (TYPE-C, 'OADATA2', 64 BYTES)
The data added to the origin data by using the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.

354 (TYPE-C, 'OADATA3', 64 BYTES)
The data added to the origin data by the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.

359 (TYPE-C 'ONETWKID', 8 BYTES
The network identifier from which this work request (transaction) originated.

360 (TYPE-C, 'OAPPLID', 8 BYTES)
The APPLID of the CICS region in which this work request (transaction) originated; for example, the region in which the CWXN task ran.

361 (TYPE-T, 'OSTART', 8 BYTES)
The time at which the originating task, for example the CWXN task, was started.

362 (TYPE-P, 'OTRANNUM', 4 BYTES)
The number of the originating task; for example, the CWXN task.

363 (TYPE-C, 'OTRAN', 4 BYTES)
The transaction ID (TRANSID) of the originating task; for example, the CWXN task.

364 (TYPE-C, 'OUSERID', 8 BYTES)
The originating Userid-2 or Userid-1, for example from CWBA, depending on the originating task.

365 (TYPE-C, 'OUSERCOR', 64 BYTES)
The originating user correlator.

366 (TYPE-C, 'OTCPSVCE', 8 BYTES)
The name of the originating TCPIPSERVICE.

367 (TYPE-A, 'OPORTNUM', 4 BYTES)
The port number used by the originating TCPIPSERVICE.

369 (TYPE-A, 'OCLIPORT', 4 BYTES)
The TCP/IP port number of the originating client or Telnet client.

370 (TYPE-A, 'OTRANFLG', 8 BYTES)
Originating transaction flags, a string of 64 bits used for signaling transaction definition and status information:

Byte 0
The facility-type of the originating transaction:

Bit 0
None (X'80')

Bit 1
Terminal (X'40')

Bit 2
Surrogate (X'20')

Bit 3
Destination (X'10')

Bit 4
3270 bridge (X'08')

Bit 5
Reserved

Bit 6
Reserved

Bit 7
Reserved

Byte 1
Transaction identification information:

Bit 0
System transaction (x'80')

Bit 1
Mirror transaction (x'40')

Bit 2
DPL mirror transaction (x'20')

Bit 3
ONC/RPC Alias transaction (x'10')

Bit 4
WEB Alias transaction (x'08')

Bit 5
3270 Bridge transaction (x'04')

Bit 6
Reserved (x'02')

Bit 7
CICS BTS Run transaction

Byte 2
z/OS workload manager request (transaction).

Byte 3
Transaction definition information:

Bit 0
Taskdataloc = below (x'80')

Bit 1
Taskdatakey = cics (x'40')

Bit 2
Isolate = no (x'20')

Bit 3
Dynamic = yes (x'10')

Bits 4–7
Reserved

Byte 4
The type of the originating transaction:

X'01'
None

X'02'
Terminal

X'03'
Transient data

X'04'
START

X'05'
Terminal-related START

X'06'
CICS business transaction services (BTS) scheduler

X'07'
Transaction manager domain (XM)-run transaction

X'08'
3270 bridge

X'09'
Socket domain

X'0A'
CICS web support (CWS)

X'0B'
Internet Inter-ORB Protocol (IIOP)

X'0C'
Resource Recovery Services (RRS)

X'0D'
LU 6.1 session

X'0E'
LU 6.2 (APPC) session

X'0F'
MRO session

X'10'
External Call Interface (ECI) session

X'11'
IIOP domain request receiver

X'12'
Request stream (RZ) instore transport

X'13'
IP interconnectivity session

X'14'
Event

X'15'
JVM server

X'16'
Asynchronous services domain (AS)-run transaction

Byte 5
Transaction status information.

Byte 6
Transaction tracking origin data tag.

Byte 7
Recovery manager information:

Bit 0
Indoubt wait = no

Bit 1
Indoubt action = commit

Bit 2
Recovery manager - UOW resolved with indoubt action

Bit 3
Recovery manager - shunt

Bit 4
Recovery manager - unshunt

Bit 5
Recovery manager - indoubt failure

Bit 6
Recovery manager - resource owner failure

Bit 7
Reserved

371 (TYPE-C, 'OFCTYNME', 8 BYTES)
The facility name of the originating transaction. If the originating transaction is not associated with a facility, this field is null. The transaction facility type, if any, can be identified using byte 0 of the originating transaction flags, OTRANFLG (370), field.

372 (TYPE-C, 'OCLIPADR', 40 BYTES)
The IP address of the originating client or Telnet client.

373 (TYPE-C, 'PHNTWKID', 8 BYTES)
The network identifier of the CICS system of an immediately previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.

374 (TYPE-C, 'PHAPPLID', 8 BYTES)
The APPLID from previous hop data. This is the APPLID of the CICS system of a previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.

375 (TYPE-T, 'PHSTART', 8 BYTES)
The start time of the immediately previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.

376 (TYPE-P, 'PHTRANNO', 4 BYTES)
The task number of the immediately previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.

377 (TYPE-C, 'PHTRAN', 4 BYTES)
The transaction ID (TRANSID) of the immediately previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.

378 (TYPE-A, 'PHCOUNT', 4 BYTES)
The number of times there has been a request from one CICS system to another CICS system to initiate a task with which this task is associated. See Previous hop data characteristics for more information about previous hop data.

402 (TYPE-A, 'EICTOTCT', 4 BYTES)
The total number of **EXEC CICS** commands issued by the user task.

405 (TYPE-A, 'TIASKTCT', 4 BYTES)
The number of **EXEC CICS ASKTIME** commands issued by the user task.

406 (TYPE-A, 'TITOTCT', 4 BYTES)
The total number of **EXEC CICS ASKTIME**, **CONVERTTIME**, and **FORMATTIME** commands issued by the user task.

408 (TYPE-A, 'BFDGSTCT', 4 BYTES)
The total number of **EXEC CICS BIF DIGEST** commands issued by the user task.

409 (TYPE-A, 'BFTOTCT', 4 BYTES)
The total number of **EXEC CICS BIF DEEDIT** and **BIF DIGEST** commands issued by the user task.

415 (TYPE-A, 'ECSIGECT', 4 BYTES)
The number of **EXEC CICS SIGNAL EVENT** commands issued by the user task.

416 (TYPE-A, 'ECEFOPCT', 4 BYTES)
The number of event filter operations performed by the user task.

417 (TYPE-A, 'ECEVNTCT', 4 BYTES)
The number of events captured by the user task.

418 (TYPE-A, 'ECSEVCCT', 4 BYTES)
The number of synchronous emission events captured by the user task.

449 (TYPE-A, 'MPPRTXCD', 4 BYTES)
The number of policy task rule thresholds that this task has exceeded. This field is all nulls (0x00 bytes) if no thresholds have been exceeded or if the task has had no task rules applied to it.

464 (TYPE-A, 'NCGETCT', 4 BYTES)
The total number of requests to a named counter server to satisfy **EXEC CICS GET COUNTER** and **EXEC CICS GET DCOUNTER** commands issued by the user task.

466 (TYPE-A, 'MPSRECT', 4 BYTES)
The number of times that policy system rules have been evaluated for the task.

467 (TYPE-A, 'MPSRACT', 4 BYTES)
The number of times that policy system rules that have been evaluated true and have triggered an action. This field is all nulls (0x00 bytes) if no system rules have been evaluated true.

480 (TYPE-T, 'PTSTART', 8 BYTES)
The start time of the immediately previous or parent task in the same CICS system with which the task is associated. See Previous transaction data characteristics for more information about previous transaction data.

481 (TYPE-P, 'PTTRANNO', 4 BYTES)
The task number of the immediately previous or parent task in the same CICS system with which the task is associated. See Previous transaction data characteristics for more information about previous transaction data.

482 (TYPE-C, 'PTTRAN', 4 BYTES)
The transaction ID (TRANSID) of the immediately previous or parent task in the same CICS system with which the task is associated. See Previous transaction data characteristics for more information about previous transaction data.

483 (TYPE-A, 'PTCOUNT', 4 BYTES)
The number of times there has been a request from one task to initiate another task in the same CICS system with which this task is associated, such as by a **RUN TRANSID** or **START** command. This is effectively the task depth in the local region when using the **RUN TRANSID** command, or the **START** command when a new point of origin is not created. See Previous transaction data characteristics for more information about previous transaction data.

Performance data in group DFHDATA

Descriptions of the performance data fields in the DFHDATA group, including the numeric identifier, type, and size of each field.

For more information about the time measurements used in some fields in this group, see Clocks and time stamps.

For more information about the elapsed time spent waiting for I/O operations and the relationship of that time to other time periods recorded for the transaction, see Transaction wait (suspend) times in Reference.

179 (TYPE-A, 'IMSREQCT', 4 BYTES)
The number of IMS (DBCTL) requests issued by the user task.

180 (TYPE-A, 'DB2REQCT', 4 BYTES)
The total number of Db2 EXEC SQL and Instrumentation Facility Interface (IFI) requests issued by the user task.

186 (TYPE-S, 'IMSWAIT', 12 BYTES)
The elapsed time during which the user task waited for DBCTL to service the IMS requests issued by the user task.

This field value is zero if IMS supports the open transaction environment (OTE).

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

187 (TYPE-S, 'DB2RDYQW', 12 BYTES)
The elapsed time during which the user task waited for a Db2 thread to become available.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

188 (TYPE-S, 'DB2CONWT', 12 BYTES)
The elapsed time during which the user task waited for a Db2 connection to become available for use with the user task's open TCB.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

395 (TYPE-A, 'WMQREQCT', 4 BYTES)
The total number of WebSphere® MQ requests issued by the user task.

396 (TYPE-S, 'WMQGETWT', 12 BYTES)
The elapsed time during which the user task waited for WebSphere MQ to service the user task's GETWAIT request.

397 (TYPE-S, 'WMQASRBT', 12 BYTES)
The WebSphere MQ SRB time this transaction spent processing WebSphere MQ API requests. Add this field to the transaction CPU time field (USRCPUT) when considering the measurement of the total processor time consumed by a transaction. This field is zero for point-to-point messaging activity, but it is nonzero where WebSphere MQ API requests result in publish and subscribe type messaging.

Note: WebSphere MQ only returns this value to CICS when Class 3 accounting information is being collected in WebSphere MQ; if this information is not being collected, the field is always zero. To start collecting Class 3 accounting information, issue the command `START TRACE(ACCTG) DEST(SMF) CLASS(3)` in WebSphere MQ.

Performance data in group DFHDEST

Descriptions of the performance data fields in the DFHDEST group, including the numeric identifier, type, and size of each field.

041 (TYPE-A, 'TDGETCT', 4 BYTES)
Number of transient data GET requests issued by the user task.

042 (TYPE-A, 'TDPUTCT', 4 BYTES)
Number of transient data PUT requests issued by the user task.

043 (TYPE-A, 'TDPURCT', 4 BYTES)
Number of transient data PURGE requests issued by the user task.

091 (TYPE-A, 'TDTOTCT', 4 BYTES)
Total number of transient data requests issued by the user task. This field is the sum of TDGETCT, TDPUTCT, and TDPURCT.

101 (TYPE-S, 'TDIOWTT', 12 BYTES)
Elapsed time in which the user waited for VSAM transient data I/O. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

403 (TYPE-S, 'TDILWTT', 12 BYTES)
The elapsed time for which the user task waited for an intrapartition transient data lock (TDIPLOCK). For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference. For more information about tasks suspended on resource type TDIPLOCK, see Resource type TDIPLOCK: waits for transient data intrapartition requests in Troubleshooting.

This field is a component of the task suspend time, SUSPTIME (014), field.

404 (TYPE-S, 'TDELWTT', 12 BYTES)
The elapsed time for which the user task waited for an extrapartition transient data lock (TDEPLOCK). For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference. For more information about tasks suspended on resource type TDEPLOCK, see Resource type TDEPLOCK: waits for transient data extrapartition requests in Troubleshooting.

This field is a component of the task suspend time, SUSPTIME (014), field.

Performance data in group DFHDOCH

Descriptions of the performance data fields in the DFHDOCH group, including the numeric identifier, type, and size of each field.

223 (TYPE-A, 'DHDELCT', 4 BYTES)
The number of document handler DELETE requests issued by the user task.

226 (TYPE-A, 'DHCRECT', 4 BYTES)
The number of document handler CREATE requests issued by the user task.

227 (TYPE-A, 'DHINSCT', 4 BYTES)
The number of document handler INSERT requests issued by the user task.

228 (TYPE-A, 'DHSETCT', 4 BYTES)
The number of document handler SET requests issued by the user task.

229 (TYPE-A, 'DHRETCT', 4 BYTES)
The number of document handler RETRIEVE requests issued by the user task.

230 (TYPE-A, 'DHTOTCT', 4 BYTES)
The total number of document handler requests issued by the user task.

240 (TYPE-A, 'DHTOTDCL', 4 BYTES)
The total length of all documents created by the user task.

Performance data in group DFHFEPI

Descriptions of the performance data fields in the DFHFEPI group, including the numeric identifier, type, and size of each field.

150 (TYPE-A,'SZALLOCT', 4 BYTES)
Number of conversations allocated by the user task. This number is incremented for each FEPI ALLOCATE POOL or FEPI CONVERSE POOL.

151 (TYPE-A,'SZRCVCT', 4 BYTES)
Number of FEPI RECEIVE requests made by the user task. This number is also incremented for each FEPI CONVERSE request.

152 (TYPE-A,'SZSENDCT', 4 BYTES)
Number of FEPI SEND requests made by the user task. This number is also incremented for each FEPI CONVERSE request.

153 (TYPE-A,'SZSTRTCT', 4 BYTES)
Number of FEPI START requests made by the user task.

154 (TYPE-A,'SZCHROUT', 4 BYTES)
Number of characters sent through FEPI by the user task.

155 (TYPE-A,'SZCHRIN', 4 BYTES)
Number of characters received through FEPI by the user task.

156 (TYPE-S,'SZWAIT', 12 BYTES)
Elapsed time in which the user task waited for all FEPI services. For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

157 (TYPE-A,'SZALLCTO', 4 BYTES)
Number of times the user task timed out while waiting to allocate a conversation.

158 (TYPE-A,'SZRCVTO', 4 BYTES)
Number of times the user task timed out while waiting to receive data.

159 (TYPE-A,'SZTOTCT', 4 BYTES)
Total number of all FEPI API and SPI requests made by the user task.

Performance data in group DFHFILE

Descriptions of the performance data fields in the DFHFILE group, including the numeric identifier, type, and size of each field.

For a breakdown by individual file of some of the information provided in group DFHFILE, you can request transaction resource monitoring. See Chapter 4, “Transaction resource class data: Listing of data fields,” on page 63 for details.

036 (TYPE-A, 'FCGETCT', 4 BYTES)
Number of file GET requests issued by the user task.

037 (TYPE-A, 'FCPUTCT', 4 BYTES)
Number of file PUT requests issued by the user task.

038 (TYPE-A, 'FCBRWCT', 4 BYTES)
Number of file browse requests issued by the user task. This number excludes the START and END browse requests.

039 (TYPE-A, 'FCADDCT', 4 BYTES)
Number of file ADD requests issued by the user task.

040 (TYPE-A, 'FCDELCT', 4 BYTES)
Number of file DELETE requests issued by the user task.

063 (TYPE-S, 'FCIOWTT', 12 BYTES)
Elapsed time in which the user task waited for file I/O. For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

070 (TYPE-A, 'FCAMCT', 4 BYTES)
Number of times the user task invoked file access-method interfaces. This number excludes requests for OPEN and CLOSE.

093 (TYPE-A, 'FCTOTCT', 4 BYTES)
Total number of file control requests issued by the user task. This number excludes any request for OPEN, CLOSE, ENABLE, or DISABLE of a file.

How **EXEC CICS** file commands correspond to file control monitoring fields is shown in Table 3 on page 24.

*Table 3. **EXEC CICS** file commands related to file control monitoring fields*

EXEC CICS command	**Monitoring fields**
READ	FCGETCT and FCTOTCT
READ UPDATE	FCGETCT and FCTOTCT
DELETE (after READ UPDATE)	FCDELCT and FCTOTCT
DELETE (with RIDFLD)	FCDELCT and FCTOTCT
REWRITE	FCPUTCT and FCTOTCT
WRITE	FCADDCT and FCTOTCT
STARTBR	FCTOTCT
READNEXT	FCBRWCT and FCTOTCT
READNEXT UPDATE	FCBRWCT and FCTOTCT
READPREV	FCBRWCT and FCTOTCT
READPREV UPDATE	FCBRWCT and FCTOTCT
ENDBR	FCTOTCT
RESETBR	FCTOTCT
UNLOCK	FCTOTCT

Note: The number of STARTBR, ENDBR, RESETBR, and UNLOCK file control requests can be calculated by subtracting the file request counts, FCGETCT, FCPUTCT, FCBRWCT, FCADDCT, and FCDELCT from the total file request count, FCTOTCT.

174 (TYPE-S, 'RLSWAIT', 12 BYTES)
Elapsed time in which the user task waited for RLS file I/O. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

175 (TYPE-S, 'RLSCPUT', 12 BYTES)
For RLS requests issued from the QR TCB:

The RLS File Request CPU (SRB) time field (RLSCPUT) is the SRB CPU time this transaction spent processing RLS file requests. This field should be added to the transaction CPU time field (USRCPUT) when considering the measurement of the total CPU time consumed by a transaction. Also, this field cannot be considered a subset of any other single CMF field (including RLSWAIT). This is because the RLS field requests execute asynchronously under an MVS SRB which can be running in parallel with the requesting transaction. It is also possible for the SRB to complete its processing before the requesting transaction waits for the RLS file request to complete.

For RLS requests issued from an open TCB:

There is no RLSCPUT field for applications that are running on an open TCB mode because the requests are completed on the same TCB on which the application is running . In this case, the CPU time for the request is already accumulated in the USRCPUT field.

Note that system initialization parameters **FCQRONLY** and **FORCEQR** can both influence the TCB under which the RLS requests are issued. See System initialization parameter descriptions and summary for details.

176 (TYPE-S, 'CFDTWAIT', 12 BYTES)
The elapsed time in which the user task waited for a data table access request to the Coupling Facility Data Table server to complete. For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

426 (TYPE-S, 'FCXCWTT', 12 BYTES)
The elapsed time in which the user task waited for exclusive control of a VSAM control interval. This field counts time spent waiting on resource type FCXCSUSP, FCXDSUSP, FCXCPROT, or FCXDPROT. For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

427 (TYPE-S, 'FCVSWTT', 12 BYTES)
The elapsed time in which the user task waited for a VSAM string. This field counts time spent waiting on resource type FCPSSUSP or FCSRSUSP. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

Performance data in group DFHJOUR

Descriptions of the performance data fields in the DFHJOUR group, including the numeric identifer, type, and size of each field.

010 (TYPE-S, 'JCIOWTT', 12 BYTES)
Elapsed time for which the user task waited for journal (logstream) I/O. For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

058 (TYPE-A, 'JNLWRTCT', 4 BYTES)
Number of journal write requests issued by the user task.

172 (TYPE-A, 'LOGWRTCT', 4 BYTES)
Number of CICS log stream write requests issued by the user task.

Performance data in group DFHMAPP

Descriptions of the performance data fields in the DFHMAPP group, including the numeric identifer, type, and size of each field.

050 (TYPE-A, 'BMSMAPCT', 4 BYTES)
Number of BMS MAP requests issued by the user task. This field corresponds to the number of RECEIVE MAP requests that did not incur a terminal I/O, and the number of RECEIVE MAP FROM requests.

051 (TYPE-A, 'BMSINCT', 4 BYTES)
Number of BMS IN requests issued by the user task. This field corresponds to the number of RECEIVE MAP requests that incurred a terminal I/O.

052 (TYPE-A, 'BMSOUTCT', 4 BYTES)
Number of BMS OUT requests issued by the user task. This field corresponds to the number of SEND MAP requests.

090 (TYPE-A, 'BMSTOTCT', 4 BYTES)
Total number of BMS requests issued by the user task. This field is the sum of BMS RECEIVE MAP, RECEIVE MAP FROM, SEND MAP, SEND TEXT, and SEND CONTROL requests issued by the user task.

Performance data in group DFHPROG

Descriptions of the performance data fields in the DFHPROG group, including the numeric identifer, type, and size of each field.

055 (TYPE-A, 'PCLINKCT', 4 BYTES)
Number of program LINK and INVOKE APPLICATION requests issued by the user task, including the link to the first program of the user task. This field does not include program LINK URM (user-replaceable module) requests.

056 (TYPE-A, 'PCXCTLCT', 4 BYTES)
Number of program XCTL requests issued by the user task.

057 (TYPE-A, 'PCLOADCT', 4 BYTES)
Number of program LOAD requests issued by the user task.

071 (TYPE-C, 'PGMNAME', 8 BYTES)
The name of the first program called at transaction attach-time.

Note these points about remote transactions:

- If the CICS definition of the remote transaction does not specify a program name, this field contains blanks.
- If the CICS definition of the remote transaction specifies a program name, this field contains the name of the specified program. (This program is not necessarily the program that is run on the remote system.)

For a dynamically routed transaction, if the dynamic transaction routing program routes the transaction locally and specifies an alternative program name, this field contains the name of the alternative program.

For a dynamic program link (DPL) mirror transaction, this field contains the initial program name specified in the dynamic program LINK request. DPL mirror transactions can be identified using byte 1 of the transaction flags, TRANFLAG (164), field.

For web service applications, this field contains the target application program name.

For a web alias transaction, this field contains the initial application program name called by the alias transaction. Web alias transactions can be identified using byte 1 of the transaction flags, TRANFLAG (164), field.

For an ONC RPC transaction, this field contains the initial application program name called by the alias transaction. ONC RPC transactions can be identified using byte 1 of the transaction flags, TRANFLAG (164), field.

For an ECI over TCP/IP transaction, this field contains the name of the application program specified in the External Call Interface (ECI) request from the client application.

072 (TYPE-A, 'PCLURMCT', 4 BYTES)
Number of program LINK URM (user-replaceable module) requests issued by, or on behalf of, the user task.

A user-replaceable module (or user-replaceable program) is a CICS-supplied program that is always called at a particular point in CICS processing, as if it were part of the CICS code. You can modify the supplied program by including your own logic, or replace it with a version that you write yourself.

The user-replaceable programs are described in Customizing with user-replaceable programs in Developing system programs.

073 (TYPE-A, 'PCDPLCT', 4 BYTES)
Number of distributed program link (DPL) requests issued by the user task.

For a breakdown by program name and system identifier (sysid) of the individual distributed program link (DPL) requests, you can request transaction resource monitoring. For more details, see Chapter 4, “Transaction resource class data: Listing of data fields,” on page 63.

113 (TYPE-C, 'ABCODEO', 4 BYTES)
Original abend code.

114 (TYPE-C, 'ABCODEC', 4 BYTES)
Current abend code.

115 (TYPE-S, 'PCLOADTM', 12 BYTES)
Elapsed time in which the user task waited for fetches from DFHRPL or dynamic LIBRARY concatenations. Only fetches for programs with installed program definitions or autoinstalled as a result of application requests are included in this figure. However, installed programs in the LPA are not included (because they do not incur a physical fetch from a library). For more information about program load time, see Clocks and time stamps, and Program load time.

286 (TYPE-A, 'PCDLCSDL', 4 BYTES)
The total length, in bytes, of the data in the containers of all the distributed program link (DPL) requests issued with the CHANNEL option by the user task. This total includes the length of any headers to the data.

287 (TYPE-A, 'PCDLCRDL', 4 BYTES)
The total length, in bytes, of the data in the containers of all DPL RETURN CHANNEL commands issued by the user task. This total includes the length of any headers to the data.

306 (TYPE-A, 'PCLNKCCT', 4 BYTES)
Number of local program LINK and INVOKE APPLICATION requests, with the CHANNEL option, issued by the user task.

This field is a subset of the program LINK and INVOKE APPLICATION requests field, PCLINKCT (055).

307 (TYPE-A, 'PCXCLCCT', 4 BYTES)
Number of program XCTL requests issued with the CHANNEL option by the user task.

This field is a subset of the program XCTL requests field, PCXCTLCT (056).

308 (TYPE-A, 'PCDPLCCT', 4 BYTES)
Number of program distributed program link (DPL) requests issued with the CHANNEL option by the user task.

This field is a subset of the distributed program link requests field, PCDPLCT (073).

309 (TYPE-A, 'PCRTNCCT', 4 BYTES)
Number of remote pseudoconversational RETURN requests, with the CHANNEL option, issued by the user task.

310 (TYPE-A, 'PCRTNCDL', 4 BYTES)
The total length, in bytes, of the data in the containers of all the remote pseudoconversational RETURN CHANNEL commands issued by the user task. This total includes the length of any headers to the data.

Abends that might not produce a transaction dump

Due to the circumstances under which transactions are called, a transaction dump might not be taken when the following abends occur:

ASPF
ASPN
ASPO
ASPP
ASPQ
ASPR
ASP1
ASP2
ASP3
ASP7
ASP8

Performance data in group DFHRMI

Descriptions of the performance data fields in the DFHRMI group, including the numeric identifier, type, and size of each field.

Group DFHRMI is present in the performance class record only if RMI=YES is specified on the DFHMCT TYPE=INITIAL macro. For more information, see the RMI parameter on the DFHMCT TYPE=INITIAL macro in Control section: DFHMCT TYPE=INITIAL.

001 (TYPE-S, 'RMITOTAL', 12 BYTES)
The total elapsed time spent in the CICS Resource Manager Interface (RMI).

For more information, see Clocks and time stamps, and RMI elapsed and suspend time.

002 (TYPE-S, 'RMIOTHER', 12 BYTES)
The total elapsed time spent in the CICS RMI for resource manager requests other than Db2, DBCTL, EXEC DLI, IBM MQ, CICSPlex® SM, and CICS TCP/IP socket requests.

003 (TYPE-S, 'RMIDB2', 12 BYTES)
The total elapsed time spent in the CICS RMI for Db2 requests.

004 (TYPE-S, 'RMIDBCTL', 12 BYTES)
The total elapsed time spent in the CICS RMI for DBCTL requests.

005 (TYPE-S, 'RMIEXDLI', 12 BYTES)
The total elapsed time spent in the CICS RMI for EXEC DLI requests.

006 (TYPE-S, 'RMIMQM', 12 BYTES)
The total elapsed time spent in the CICS RMI for IBM MQ requests.

007 (TYPE-S, 'RMICPSM', 12 BYTES)
The total elapsed time spent in the CICS RMI for CICSPlex SM requests.

008 (TYPE-S, 'RMITCPIP', 12 BYTES)
The total elapsed time spent in the CICS RMI for CICS TCP/IP socket requests.

Performance data in group DFHSOCK

Descriptions of the performance data fields in the DFHSOCK group, including the numeric identifier, type, and size of each field.

241 (TYPE-S, 'SOIOWTT', 12 BYTES)
The elapsed time the user task spent waiting for any socket I/O to complete. This time includes the time the task spent on send and receive calls. It applies to inbound and outbound sockets. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

242 (TYPE-A, 'SOBYENCT', 4 BYTES)
The number of bytes encrypted by the secure sockets layer for the user task.

243 (TYPE-A, 'SOBYDECT', 4 BYTES)
The number of bytes decrypted by the secure sockets layer for the user task.

245 (TYPE-C, 'TCPSRVCE', 8 BYTES)
The TCP/IP service name that attached the user task.

246 (TYPE-A, 'PORTNUM', 4 BYTES)
The TCP/IP port number of the TCP/IP service that attached the user task.

288 (TYPE-A, 'ISALLOCT, 4 BYTES)
The number of allocate session requests issued by the user task for sessions using IPIC.

289 (TYPE-A, 'SOEXTRCT', 4 BYTES)
The number of EXTRACT TCPIP and EXTRACT CERTIFICATE requests issued by the user task.

290 (TYPE-A,'SOCNPSCT',4 BYTES)
The total number of requests made by the user task to create a nonpersistent outbound socket.

291 (TYPE-A,'SOCPSCT',4 BYTES)
The total number of requests made by the user task to create a persistent outbound socket.

292 (TYPE-A,'SONPSHWM',4 BYTES)
The peak number of nonpersistent outbound sockets owned by the user task.

293 (TYPE-A,'SOPSHWM',4 BYTES)
The peak number of persistent outbound sockets owned by the user task.

294 (TYPE-A,'SORCVCT',4 BYTES)
The total number of receive requests issued for outbound sockets (persistent and nonpersistent) by the user task.

295 (TYPE-A,'SOCHRIN',4 BYTES)
The total number of bytes received on outbound sockets by the user task

296 (TYPE-A,'SOSENDCT',4 BYTES)
The total number of send requests issued for outbound sockets (persistent and nonpersistent) by the user task.

297 (TYPE-A,'SOCHROUT',4 BYTES)
The total number of bytes sent on outbound sockets by the user task.

298 (TYPE-A, 'SOTOTCT', 4 BYTES)
The total number of socket requests issued by the user task.

299 (TYPE-S, 'SOOIOWTT', 12 BYTES)
The elapsed time that the user task spent waiting for a socket to be created by the socket manager when CICS is at the MAXSOCKETS limit. This field is currently not used because no callers within CICS choose to wait for a socket to become available when CICS is at the MAXSOCKETS limit.

This field is a component of the task suspend time, SUSPTIME (014), field.

300 (TYPE-S, 'ISIOWTT', 12 BYTES)
The elapsed time for which a user task waited for control at this end of an IPIC connection.

301 (TYPE-A, 'SOMSGIN1', 4 BYTES)
The number of inbound socket receive requests issued by the user task.

302 (TYPE-A, 'SOCHRIN1', 4 BYTES)
The number of characters received by inbound socket receive requests issued by the user task.

303 (TYPE-A, 'SOMSGOU1', 4 BYTES)
The number of inbound socket send requests issued by the user task.

304 (TYPE-A, 'SOCHROU1', 4 BYTES)
The number of characters sent by inbound socket send requests issued by the user task.

305 (TYPE-C, 'ISIPICNM', 8 BYTES)
The name of the IPIC connection for the TCP/IP service that attached the user task.

318 (TYPE-C, 'CLIPADDR', 40 BYTES)
The IP address of the client or Telnet client.

319 (TYPE-S, 'ISALWTT', 12 BYTES)
The elapsed time for which a user task waited for an allocate request for an IPIC session. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

320 (TYPE-A, 'SOCIPHER', 4 BYTES)
Identifies the code for the cipher suite that was selected during the SSL handshake for use on the inbound connection, for example X'0000002F'. For a list of the cipher suites that are supported by CICS and z/OS and their codes, see Cipher suites and cipher suite specification files.

330 (TYPE-A, 'CLIPPORT', 4 BYTES)
The port number of the client or Telnet client.

344 (TYPE-C, 'SOCONMSG', 4 BYTES)
Indicates whether the task processed the first message for establishing a new connection for a client. This field helps you measure how often a new socket connection is created.

Y
Indicates that the task processed the first message from the client.

N
Indicates that the task processed a subsequent message from the client.

Performance data in group DFHSTOR

Descriptions of the performance data fields in the DFHSTOR group, including the numeric identifier, type, and size of each field.

User storage fields in group DFHSTOR

033 (TYPE-A, 'SCUSRHWM', 4 BYTES)
Maximum amount (high-water mark) of user storage allocated to the user task below the 16 MB line, in the user dynamic storage area (UDSA).

054 (TYPE-A, 'SCUGETCT', 4 BYTES)
Number of user-storage GETMAIN requests issued by the user task for storage below the 16 MB line, in the UDSA.

095 (TYPE-A, 'SCUSRSTG', 8 BYTES)
Storage occupancy of the user task below the 16 MB line, in the UDSA. This measures the area under the curve of storage in use against elapsed time. For more information about storage occupancy, see Storage occupancy counts.

105 (TYPE-A, 'SCUGETCT', 4 BYTES)
Number of user-storage GETMAIN requests issued by the user task for storage above the 16 MB line, in the extended user dynamic storage area (EUDSA).

106 (TYPE-A, 'SCUSRHWM', 4 BYTES)
Maximum amount (high-water mark) of user storage allocated to the user task above the 16 MB line, in the EUDSA.

107 (TYPE-A, 'SCUSRSTG', 8 BYTES)
Storage occupancy of the user task above the 16 MB line, in the EUDSA. This measures the area under the curve of storage in use against elapsed time. For more information, see Storage occupancy counts.

116 (TYPE-A, 'SC24CHWM', 4 BYTES)
Maximum amount (high-water mark) of user storage allocated to the user task below the 16 MB line, in the CICS dynamic storage area (CDSA).

117 (TYPE-A, 'SCCGETCT', 4 BYTES)
Number of user-storage GETMAIN requests issued by the user task for storage below the 16 MB line, in the CDSA.

118 (TYPE-A, 'SC24COCC', 8 BYTES)
Storage occupancy of the user task below the 16 MB line, in the CDSA. This measures the area under the curve of storage in use against elapsed time. For more information, see Storage occupancy counts.

119 (TYPE-A, 'SC31CHWM', 4 BYTES)
Maximum amount (high-water mark) of user storage allocated to the user task above the 16 MB line, in the extended CICS dynamic storage area (ECDSA).

120 (TYPE-A, 'SCCGETCT', 4 BYTES)
Number of user-storage GETMAIN requests issued by the user task for storage above the 16 MB line, in the ECDSA.

121 (TYPE-A, 'SC31COCC', 8 BYTES)
Storage occupancy of the user task above the 16 MB line, in the ECDSA. This measures the area under the curve of storage in use against elapsed time. For more information, see Storage occupancy counts.

441 (TYPE-A, 'SC64CGCT', 4 BYTES)
Number of user-storage GETMAIN requests issued by the user task for storage above the bar, in the CICS dynamic storage area (GCDSA).

442 (TYPE-A, 'SC64CHWM', 4 BYTES)
Maximum amount (high-water mark) of user storage, rounded up to the next 4K, allocated to the user task above the bar, in the CICS dynamic storage area (GCDSA).

443 (TYPE-A, 'SC64UGCT', 4 BYTES)
Number of user-storage GETMAIN requests issued by the user task for storage above the bar, in the user dynamic storage area (GUDSA).

444 (TYPE-A, 'SC64UHWM', 4 BYTES)
Maximum amount (high-water mark) of user storage, rounded up to the next 4K, allocated to the user task above the bar, in the user dynamic storage area (GUDSA).

Table 4. User storage field ID by DSA (CICS key and user key)

Field	UDSA	EUDSA	CDSA	ECDSA	GCDSA	GUDSA
GETMAIN request count	054	105	117	120	441	443
High-water mark	033	106	116	119	442	444
Occupancy	095	107	118	121	n/a	n/a

Shared storage fields in group DFHSTOR

144 (TYPE-A, 'SC24SGCT', 4 BYTES)
Number of storage GETMAIN requests issued by the user task for shared storage below the 16 MB line, in the CDSA or SDSA.

145 (TYPE-A, 'SC24GSHR', 4 BYTES)
Number of bytes of shared storage obtained by the user task by using a GETMAIN request below the 16 MB line, in the CDSA or SDSA.

146 (TYPE-A, 'SC24FSHR', 4 BYTES)
Number of bytes of shared storage released by the user task by using a FREEMAIN request below the 16 MB line, in the CDSA or SDSA.

147 (TYPE-A, 'SC31SGCT', 4 BYTES)
Number of storage GETMAIN requests issued by the user task for shared storage above the 16 MB line, in the ECDSA or ESDSA.

148 (TYPE-A, 'SC31GSHR', 4 BYTES)
Number of bytes of shared storage obtained by the user task by using a GETMAIN request above the 16 MB line, in the ECDSA or ESDSA.

149 (TYPE-A, 'SC31FSHR', 4 BYTES)
Number of bytes of shared storage released by the user task by using a FREEMAIN request above the 16 MB line, in the ECDSA or ESDSA.

445 (TYPE-A, 'SC64SGCT', 4 BYTES)
Number of storage GETMAIN requests issued by the user task for shared storage above the bar, in the GCDSA or GSDSA.

446 (TYPE-A, 'SC64GSHR', 4 BYTES)
Amount of shared storage obtained by the user task by using a GETMAIN request above the bar, in the GCDSA or GSDSA. The total number of bytes obtained is rounded up to the next 4096 bytes, and the resulting number of 4K pages is displayed.

447 (TYPE-A, 'SC64FSHR', 4 BYTES)
Amount of shared storage released by the user task by using a FREEMAIN request above the bar, in the GCDSA or GSDSA. The total number of bytes obtained is rounded up to the next 4096 bytes, and the resulting number of 4K pages is displayed.

Table 5. Shared storage field ID by DSA

Field	CDSA or SDSA	ECDSA or ESDSA	GCDSA or GSDSA
GETMAIN request count	144	147	445
Shared storage obtained	145	148	446
Shared storage released	146	149	447

Program storage fields in group DFHSTOR

For more information on program storage, see Storage manager statistics.

Note: If a task loads the same program several times, the fields in this group might not reflect the true high-water mark of program storage used by the task. The fields are incremented each time the LOAD command is issued, but if the program has already been loaded by the task, the existing copy of the program is used, meaning that only one copy of the program exists in storage. Because of this, for tasks that repeatedly load the same program, the data in the fields PCSTGHWM, PC24BHWM, PC31RHWM, PC31AHWM, PC31CHWM, PC24CHWM, PC24SHWM, PC31SHWM and PC24RHWM should be used with caution.

087 (TYPE-A, 'PCSTGHWM', 4 BYTES)
Maximum amount (high-water mark) of program storage in use by the user task both above *and* below the 16 MB line.

108 (TYPE-A, 'PC24BHWM', 4 BYTES)
Maximum amount (high-water mark) of program storage in use by the user task below the 16 MB line. This field is a subset of PCSTGHWM (field id 087) that resides below the 16 MB line.

122 (TYPE-A, 'PC31RHWM', 4 BYTES)
Maximum amount (high-water mark) of program storage in use by the user task above the 16 MB line, in the extended read-only dynamic storage area (ERDSA). This field is a subset of PC31AHWM (field id 139) that resides in the ERDSA.

139 (TYPE-A, 'PC31AHWM', 4 BYTES)
Maximum amount (high-water mark) of program storage in use by the user task above the 16 MB line. This field is a subset of PCSTGHWM (field id 087) that resides above the 16 MB line.

142 (TYPE-A, 'PC31CHWM', 4 BYTES)
Maximum amount (high-water mark) of program storage in use by the user task above the 16 MB line, in the extended CICS dynamic storage area (ECDSA). This field is a subset of PC31AHWM (139) that resides in the ECDSA.

143 (TYPE-A, 'PC24CHWM', 4 BYTES)
Maximum amount (high-water mark) of program storage in use by the user task below the 16 MB line, in the CICS dynamic storage area (CDSA). This field is a subset of PC24BHWM (108) that resides in the CDSA.

160 (TYPE-A, 'PC24SHWM', 4 BYTES)
Maximum amount (high-water mark) of program storage in use by the user task below the 16 MB line, in the shared dynamic storage area (SDSA). This field is a subset of PC24BHWM (108) that resides in the SDSA.

161 (TYPE-A, 'PC31SHWM', 4 BYTES)
Maximum amount (high-water mark) of program storage in use by the user task above the 16 MB line, in the extended shared dynamic storage area (ESDSA). This field is a subset of PC31AHWM (139) that resides in the ESDSA.

162 (TYPE-A, 'PC24RHWM', 4 BYTES)
Maximum amount (high-water mark) of program storage in use by the user task below the 16 MB line, in the read-only dynamic storage area (RDSA). This field is a subset of PC24BHWM (108) that resides in the RDSA.

Performance data in group DFHSYNC

Descriptions of the performance data fields in the DFHSYNC group, including the numeric identifer, type, and size of each field.

060 (TYPE-A, 'SPSYNCCT', 4 BYTES)
Number of SYNCPOINT requests issued during the user task.

Note:

1. A SYNCPOINT is implicitly issued as part of the task-detach processing.
2. A SYNCPOINT is issued at PSB termination for DBCTL.

173 (TYPE-S, 'SYNCTIME', 12 BYTES)
Total elapsed time for which the user task was dispatched and was processing syncpoint requests.

177 (TYPE-S, 'SRVSYWTT', 12 BYTES)
Total elapsed time in which the user task waited for syncpoint or resynchronization processing using the Coupling Facility data tables server to complete.

Note: This field is a component of the task suspend time, SUSPTIME (014), field.

196 (TYPE-S, 'SYNCDLY', 12 BYTES)
The elapsed time in which the user task waited for a syncpoint request to be issued by its parent transaction. The user task was executing as a result of the parent task issuing a CICS BTS run-process or run-activity request to execute a process or activity synchronously. For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

199 (TYPE-S, 'OTSINDWT', 12 BYTES)
The elapsed time in which the user task was dispatched or suspended indoubt (or both) while processing a syncpoint for an Object Transaction Service (OTS) syncpoint request. For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014) field.

Performance data in group DFHTASK

Performance data fields in the DFHTASK group are described, including the numeric identifier, type, and size of each field.

001 (TYPE-C, 'TRAN', 4 BYTES)
Transaction identification.

004 (TYPE-C,'TTYPE',4 BYTES)
Transaction start type. The high-order bytes (0 and 1) are set as follows:

'TO'
Attached from terminal input

'S '
Attached by automatic transaction initiation (ATI) without data

'SD'
Attached by automatic transaction initiation (ATI) with data

'QD'
Attached by transient data trigger level

'U '
Attached by user request

'TP'
Attached from terminal TCTTE transaction ID

'SZ'
Attached by front-end programming interface (FEPI)

007 (TYPE-S, 'USRDISPT', 12 BYTES)
Total elapsed time during which the user task was dispatched on each CICS TCB under which the task ran. The TCB modes managed by the CICS dispatcher are: QR, RO, CO, FO, SZ ,RP, SL, SP, SO, EP, L8, L9, S8, TP, T8, X8, X9, and D2. Be aware that, for each CICS release, new TCB modes might be added to this list, or obsolete TCB modes might be removed. For more information about dispatch time and CPU time, see Transaction dispatch time and CPU time.

008 (TYPE-S, 'USRCPUT', 12 BYTES)
Processor time for which the user task was dispatched on each CICS TCB under which the task ran. The TCB modes managed by the CICS dispatcher are: QR, RO, CO, FO, SZ, RP, SL, SP, SO, EP, L8, L9, S8, TP, T8, X8, X9, and D2. Be aware that, for each CICS release, new TCB modes might be added to this list, or obsolete TCB modes might be removed. For more information about dispatch time and CPU time, see Transaction dispatch time and CPU time.

014 (TYPE-S, 'SUSPTIME', 12 BYTES)
Total elapsed wait time for which the user task was suspended by the dispatcher. This wait time includes these values:

- The elapsed time waiting for the first dispatch. This elapsed time also includes any delay incurred because of the limits set for the class of the transaction (if any) or by the system parameter MXT being reached.
- The task suspend (wait) time.
- The elapsed time waiting for redispatch after a suspended task has been resumed.

For more information, see Transaction wait (suspend) times in Reference.

031 (TYPE-P, 'TRANNUM', 4 BYTES)
Transaction identification number. The transaction number field is normally a 4-byte packed decimal number. However, some CICS system tasks are identified by transaction numbers that comprise special characters, as follows:

' III'
System initialization task

' TCP'
Terminal control

These special identifiers are placed in bytes 2 - 4. Byte 1 is a blank (X'40') before the terminal control TCP identifier, and a null value (X'00') before the others.

059 (TYPE-A, 'ICPUINCT', 4 BYTES)
Number of interval control START or INITIATE requests during the user task.

064 (TYPE-A, 'TASKFLAG', 4 BYTES)
Task error flags, a string of 32 bits used for signaling unusual conditions occurring during the user task:

Bit 0
Reserved

Bit 1
Detected an attempt either to start a user clock that was already running or to stop one that was not running

Bits 2–31
Reserved

065 (TYPE-A, 'ICSTACCT', 4 BYTES)
Total number of local interval control START requests, with the CHANNEL option, issued by the user task.

066 (TYPE-A, 'ICTOTCT', 4 BYTES)
Total number of Interval Control Start, Cancel, Delay, and Retrieve requests issued by the user task.

082 (TYPE-C, 'TRNGRPID', 28 BYTES)
The transaction group ID is assigned at transaction attach time, and can be used to correlate the transactions that CICS runs for the same incoming work request; for example, the CWXN and CWBA transactions for Web requests. This transaction group ID relationship is useful when applied to the requests that originate through the CICS Web, ECI over TCP/IP, or 3270 bridge interface, as indicated by the transaction origin in byte 4 of the transaction flags field (group name DFHTASK, field ID 164). For more information on using the transaction group ID, see Transaction tracking in the CICS Intercommunication Guide.

097 (TYPE-C, 'NETUOWPX', 20 BYTES)
Fully qualified name by which the originating system is known to the z/OS Communications Server network. This name is assigned at attach time using either the netname derived from the TCT (when the task is attached to a local terminal) or the netname passed as part of an ISC APPC or IRC attach header. At least three padding bytes (X'00') are present at the right end of the name.

If the originating terminal is z/OS Communications Server across an ISC APPC or IRC link, the NETNAME is the *networkid.LUname*. If the terminal is non-z/OS Communications Server, the NETNAME is *networkid.generic_applid*.

All originating information passed as part of an ISC LUTYPE6.1 attach header has the same format as the non-z/OS Communications Server terminal originators.

When the originator is communicating over an external CICS interface (EXCI) session, the name is a concatenation of:

```
'DFHEXCIU | .       |  MVS Id    |  Address Space Id (ASID)'
 8 bytes  | 1 byte  |  4 bytes   |  4 bytes
```

derived from the originating system. That is, the name is a 17-byte LU name consisting of these fields:

- An 8-byte eye-catcher set to 'DFHEXCIU'.
- A 1-byte field containing a period (.).
- A 4-byte field containing the MVSID, in characters, under which the client program is running.
- A 4-byte field containing the address space ID (ASID) in which the client program is running. This field contains the 4-character EBCDIC representation of the 2-byte hex address space ID.

098 (TYPE-C, 'NETUOWSX', 8 BYTES)
Name by which the network unit of work ID is known in the originating system. This name is assigned at attach time using either an STCK-derived token (when the task is attached to a local terminal), or the network unit of work ID passed as part of an ISC (APPC) or IRC (MRO) attach header.

The first 6 bytes of this field are a binary value derived from the system clock of the originating system and which can wrap round at intervals of several months.

The last 2 bytes of this field are for the period count. These bytes can change during the life of the task as a result of sync point activity.

When using MRO or ISC, the NETUOWSX field must be combined with the NETUOWPX field (097) to uniquely identify a task, because NETUOWSX is unique only to the originating CICS system.

102 (TYPE-S, 'DISPWTT', 12 BYTES)
Elapsed time for which the user task waited for redispatch. This time is the aggregate of the wait times between each event completion and user-task redispatch.

This field does not include the elapsed time spent waiting for first dispatch. This field is a component of the task suspend time, SUSPTIME (014), field.

109 (TYPE-C, 'TRANPRI', 4 BYTES)
Transaction priority when monitoring of the task was initialized (low-order byte-3).

123 (TYPE-S, 'GNQDELAY', 12 BYTES)
The elapsed time waiting for a CICS task control global enqueue. For more information, see Clocks and time stamps.

This field is a component of the task suspend time, SUSPTIME (014), field.

124 (TYPE-C, 'BRDGTRAN', 4 BYTES)
Bridge listener transaction identifier. For CICS 3270 Bridge transactions, this field is the name of the Bridge listener transaction that attached the user task.

125 (TYPE-S, 'DSPDELAY', 12 BYTES)
The elapsed time waiting for first dispatch.

This field is a component of the task suspend time, SUSPTIME (014), field. For more information, see Clocks and time stamps.

126 (TYPE-S, 'TCLDELAY', 12 BYTES)
The elapsed time waiting for first dispatch, which was delayed because of the limits set for the transaction class of this transaction, TCLSNAME (166), being reached. For more information, see Clocks and time stamps. This field is a component of the first dispatch delay, DSPDELAY (125), field.

127 (TYPE-S, 'MXTDELAY', 12 BYTES)
The elapsed time waiting for the first dispatch, which was delayed because of the limits set by the system parameter, MXT, being reached. The field is a component of the first dispatch delay, DSPDELAY (125), field.

128 (TYPE-S, 'LMDELAY', 12 BYTES)
The elapsed time that the user task waited to acquire a lock on a resource. A user task cannot explicitly acquire a lock on a resource, but many CICS modules lock resources on behalf of user tasks using the CICS lock manager (LM) domain.

For more information about CICS lock manager, see Investigating lock manager waits.

For information about times, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

129 (TYPE-S, 'ENQDELAY', 12 BYTES)
The elapsed time waiting for a CICS task control local enqueue. For more information, see Clocks and time stamps. This field is a component of the task suspend time, SUSPTIME (014), field.

132 (TYPE-T, 'RMUOWID', 8 BYTES)
The identifier of the unit of work (unit of recovery) for this task. Unit of recovery values are used to synchronize recovery operations among CICS and other resource managers, such as IMS and Db2.

163 (TYPE-C, 'FCTYNAME', 4 BYTES)
Transaction facility name. This field is null if the transaction is not associated with a facility. The transaction facility type (if any) can be identified using byte 0 of the transaction flags, **TRANFLAG**, (164) field.

164 (TYPE-A, 'TRANFLAG', 8 BYTES)
Transaction flags, a string of 64 bits used for signaling transaction definition and status information:

Byte 0
The facility-type of the originating transaction:

Bit 0
Transaction facility name = none (x'80')

Bit 1
Transaction facility name = terminal (x'40')

If this bit is set, FCTYNAME and TERM contain the same terminal ID.

Bit 2
Transaction facility name = surrogate (x'20')

Bit 3
Transaction facility name = destination (x'10')

Bit 4
Transaction facility name = 3270 bridge (x'08')

Bits 5–7
Reserved

Byte 1
Transaction identification information:

Bit 0
System transaction (x'80')

Bit 1
Mirror transaction (x'40')

Bit 2
DPL mirror transaction (x'20')

Bit 3
ONC/RPC Alias transaction (x'10')

Bit 4
WEB Alias transaction (x'08')

Bit 5
3270 Bridge transaction (x'04')

Bit 6
Reserved (x'02')

Bit 7
CICS BTS Run transaction

Byte 2
z/OS workload manager request (transaction) completion information:

Bit 0
Report the total response time (begin-to-end phase) for completed work request (transaction).

Bit 1
Notify that the entire execution phase of the work request is complete.

Bit 2
Notify that a subset of the execution phase of the work request is complete.

Bit 3
This transaction has been reported to the z/OS workload manager as completing abnormally because it has tried to access Db2 and a "connection unavailable" response has been returned. This abnormal completion occurs when all the following are true:

- Bit 0 is set.
- CICS is not connected to Db2.
- The CICS-Db2 adapter is in standby mode (STANDBYMODE(RECONNECT) or STANDBYMODE(CONNECT)).
- CONNECTERROR(SQLCODE) is specified, causing the application to receive a -923 SQL code.

Bits 4-7
Reserved

Byte 3
Transaction definition information:

Bit 0
Taskdataloc = below (x'80')

Bit 1
Taskdatakey = cics (x'40')

Bit 2
Isolate = no (x'20')

Bit 3
Dynamic = yes (x'10')

Bits 4–7
Reserved

Byte 4
Transaction origin type:

X'01'
None

X'02'
Terminal

X'03'
Transient data

X'04'
START

X'05'
Terminal-related START

X'06'
CICS business transaction services (BTS) scheduler

X'07'
Transaction manager domain (XM)-run transaction

X'08'
3270 bridge

X'09'
Sockets domain

X'0A'
CICS Web support (CWS)

X'0B'
Internet Inter-ORB Protocol (IIOP)

X'0C'
Resource Recovery Services (RRS)

X'0D'
LU 6.1 session

X'0E'
LU 6.2 (APPC) session

X'0F'
MRO session

X'10'
External Call Interface (ECI) session

X'11'
IIOP domain request receiver

X'12'
Request stream (RZ) instore transport

X'13'
IPIC session

X'14'
Event

X'15'
JVMSERVER

X'16'
Asynchronous services domain (AS)-run transaction

X'17
NODEJSAPP

Byte 5
Transaction status information:

Bit 0
The transaction origin

Bit 1
Reserved

Bit 2
Resource class record, or records, for this task

Bit 3
Identity class record, or records, for this task

Bit 4
Reserved

Bit 5
Reserved

Bit 6
Task purge or runaway resulted in the open TCB the task was executing on being terminated.

Note: If bit 6 is set, the transaction's timing clocks were left in an unreliable state. As a result, the clocks are set to zero when the record is written by the CICS Monitoring Facility (CMF).

Bit 7
Task abnormally terminated

Byte 6
Transaction tracking origin data tag. For more information about how to set the transaction tracking origin data tag, see The SET_TRACKING_DATA call.

Note: The transaction tracking origin data tag can be set in the record only when it is the transaction origin. The transaction origin can be determined when bit 0 is set in the transaction status information byte 5 of the **TRANFLAG** field.

Byte 7
Recovery manager information:

Bit 0
Indoubt wait = no

Bit 1
Indoubt action = commit

Bit 2
Recovery manager, UOW resolved with indoubt action

Bit 3
Recovery manager, Shunt

Bit 4
Recovery manager, Unshunt

Bit 5
Recovery manager, Indoubt failure

Bit 6
Recovery manager, Resource owner failure

Bit 7
Reserved

Note: Bits 2 through 6 are reset on a SYNCPOINT request when the MNSYNC=YES option is specified.

166 (TYPE-C, 'TCLSNAME', 8 BYTES)
Transaction class name. This field is null if the transaction is not in a TRANCLASS.

170 (TYPE-S, 'RMITIME', 12 BYTES)
The total elapsed time spent in the CICS Resource Manager Interface (RMI). For more information, see Clocks and time stamps, Transaction wait (suspend) times in Reference, and RMI elapsed and suspend time.

171 (TYPE-S, 'RMISUSP', 12 BYTES)
The total elapsed time that the task was suspended by the CICS dispatcher while in the CICS Resource Manager Interface (RMI). For more information, see Clocks and time stamps, Transaction wait (suspend) times in Reference, and RMI elapsed and suspend time. The field is a component of the task suspend time, SUSPTIME (014), field and also the RMITIME (170) field.

181 (TYPE-S, 'WTEXWAIT', 12 BYTES)
The elapsed time that the user task waited for one or more ECBs, passed to CICS by the user task using the **EXEC CICS WAIT EXTERNAL ECBLIST** command, to be posted by the MVS POST command. The user task can wait on one or more ECBs. If it waits on more than one, it is dispatchable as soon as one of the ECBs is posted. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

182 (TYPE-S, 'WTCEWAIT', 12 BYTES)
The elapsed time that the user task waited for one of these events:

- One or more ECBs, passed to CICS by the user task using the **EXEC CICS WAITCICS ECBLIST** command, to be posted by the MVS POST command. The user task can wait on one or more ECBs. If it waits on more than one, it is dispatchable as soon as one of the ECBs is posted.
- Completion of an event initiated by the same or by another user task. The event is usually be the posting, at the expiration time, of a timer-event control area provided in response to an **EXEC CICS POST** command. The **EXEC CICS WAIT EVENT** command provides a method of directly giving up control to some other task until the event being waited on is completed.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

183 (TYPE-S, 'ICDELAY', 12 BYTES)
The elapsed time that the user task waited as a result of issuing one of the following commands:

- An interval control **EXEC CICS DELAY** command for a specified time interval.
- An interval control **EXEC CICS DELAY** command for a specified time of day to expire.
- An interval control **EXEC CICS RETRIEVE** command with the WAIT option specified.

For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.This field is a component of the task suspend time, SUSPTIME (014), field.

184 (TYPE-S, 'GVUPWAIT', 12 BYTES)
The elapsed time that the user task waited as a result of giving up control to another task. A user task can give up control in many ways. Some examples are application programs that use one or more of the following **EXEC CICS** API or SPI commands:

- The **EXEC CICS SUSPEND** command. This command causes the issuing task to give up control to another task of higher or equal dispatching priority. Control is returned to this task as soon as no other task of a higher or equal priority is ready to be dispatched.
- The **EXEC CICS CHANGE TASK PRIORITY** command. This command immediately changes the priority of the issuing task and causes the task to give up control for it to be dispatched at its new priority. The task is not redispatched until tasks of higher or equal priority, and that are also dispatchable, have been dispatched.
- The **EXEC CICS DELAY** command with INTERVAL (0). This command causes the issuing task to give up control to another task of higher or equal dispatching priority. Control is returned to this task as soon as no other task of a higher or equal priority is ready to be dispatched.
- The **EXEC CICS POST** command requesting notification that a specified time has expired. This command causes the issuing task to give up control so that CICS has the opportunity to post the time-event control area.
- The **EXEC CICS PERFORM RESETTIME** command to synchronize the CICS date and time with the z/OS system date and time of day.
- The **EXEC CICS START TRANSID** command with the ATTACH option.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

190 (TYPE-C, 'RRMSURID', 16 BYTES)
RRMS/MVS unit-of-recovery ID (URID).

191 (TYPE-S, 'RRMSWAIT', 12 BYTES)
The elapsed time in which the user task waited indoubt using resource recovery services for EXCI.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

192 (TYPE-S, 'RQRWAIT', 12 BYTES)
The elapsed time during which the request receiver user task CIRR (or user specified transaction ID) waited for any outstanding replies to be satisfied.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

193 (TYPE-S, 'RQPWAIT', 12 BYTES)
The elapsed time during which the request processor user task CIRP waited for any outstanding replies to be satisfied.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

194 (TYPE-C, 'OTSTID', 128 BYTES)
This field is the first 128 bytes of the Object Transaction Service (OTS) Transaction ID (TID).

195 (TYPE-S, 'RUNTRWTT', 12 BYTES)
The elapsed time in which the user task waited for completion of a transaction that ran as a result of the user task issuing a CICS BTS run process request and a run activity request synchronously.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

247 (TYPE-S, 'DSCHMDLY', 12 BYTES)
The elapsed time in which the user task waited for redispatch after a CICS Dispatcher change-TCB mode request was issued by or on behalf of the user task. For example, a change-TCB mode request from a CICS L8 or S8 mode TCB back to the CICS QR mode TCB might have to wait for the QR TCB because another task is currently dispatched on the QR TCB. This field is a component of the task suspend time, SUSPTIME (014), field.

249 (TYPE-S, 'QRMODDLY', 12 BYTES)
The elapsed time for which the user task waited for redispatch on the CICS QR mode TCB. This time is the aggregate of the wait times between each event completion and user-task redispatch. This field does not include the elapsed time spent waiting for the first dispatch. The QRMODDLY field is a component of the task suspend time, SUSPTIME (014), field, and also the redispatch wait, DISPWTT (102), field.

250 (TYPE-S, 'MAXOTDLY', 12 BYTES)
The elapsed time in which the user task waited to obtain a CICS L8 or L9 mode open TCB, because the region had reached the limit set by CICS for these TCBs. L8 and L9 mode open TCBs are used by OPENAPI application programs or by task-related user exit programs that have been enabled with the OPENAPI option, for example, the CICS-Db2 adapter, when CICS connects to DB2® Version 6 or later and the CICS-MQ adapter, when CICS connects to Websphere MQ Version 6 or later.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

251 (TYPE-A, 'TCBATTCT', 4 BYTES)
The number of CICS TCBs attached by or on behalf of the user task.

252 (TYPE-A, 'DSTCBHWM', 4 BYTES)
The peak number of CICS open TCBs (in TCB modes L8, L9, S8, T8, X8, and X9) that have been concurrently allocated to the user task.

253 (TYPE-S, 'JVMTIME', 12 BYTES)
The total elapsed time spent in the JVM by the user task. For more information, see JVM elapsed time, suspend time, and cleanup time.

254 (TYPE-S, 'JVMSUSP', 12 BYTES)
The elapsed time for which the user task was suspended by the CICS dispatcher while running in the JVM. For more information, see JVM elapsed time, suspend time, and cleanup time. This field is a component of the task suspend time, SUSPTIME (014), field.

255 (TYPE-S, 'QRDISPT', 12 BYTES)
The elapsed time for which the user task was dispatched on the CICS QR TCB. For more information, see Clocks and time stamps.

256 (TYPE-S, 'QRCPUT', 12 BYTES)
The processor time for which the user task was dispatched on the CICS QR TCB. For more information, see Clocks and time stamps.

257 (TYPE-S, 'MSDISPT', 12 BYTES)
Elapsed time for which the user task was dispatched on each CICS TCB. The CICS TCB modes are used as follows:

- RO and FO are always used.
- CO is used if **SUBTSKS=1** is specified as a system initialization parameter.
- SZ is used if FEPI is active.

- RP is used if ONC/RPC is installed and active.
- SL, SO, and SP are used if **TCPIP=YES** is specified as a system initialization parameter. Mode SL is used by the CICS support for TCP/IP (TCP/IP Service) Listener system transaction CSOL. Mode SO is used to process the CICS support for TCP/IP socket requests issued by or on behalf of the user task. Mode SP is the CICS support for TCP/IP sockets IPT task (Initial Pthread TCB) and also owns all the SSL pthreads (S8 TCBs).
- D2 is used to stop Db2 protected threads.
- EP is used for event processing.
- CICS creates a TP mode TCB for every JVMSERVER resource definition that is installed and enabled. The TP TCB owns the IPT task (Initial Process Thread TCB), the Language Environment® enclave, the JVM, the THRD TCB pool, and the T8 TCBs for that JVM server.

For more information, see Clocks and time stamps.

258 (TYPE-S, 'MSCPUT', 12 BYTES)
The processor time for which the user task was dispatched on each CICS TCB. The usage of each CICS TCB is shown in the description for field **MSDISPT** (field ID 257 in group DFHTASK). For more information, see Clocks and time stamps.

259 (TYPE-S, 'L8CPUT', 12 BYTES)
The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS L8 mode TCB. When a transaction starts an OPENAPI application program defined with EXECKEY=CICS, or a task-related user exit program that has been enabled with the OPENAPI option, CICS allocates a CICS L8 mode TCB to the task. (An L8 mode TCB can also be allocated if the OPENAPI program is defined with EXECKEY=USER, but the storage protection facility is inactive.) After a task has been allocated an L8 mode TCB, that same TCB remains associated with the task until the transaction is detached. For more information on this field, see Clocks and time stamps.

261 (TYPE-S, 'S8CPUT', 12 BYTES)
The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS S8 mode TCB. A transaction is allocated a CICS S8 mode TCB when it uses the secure sockets layer (SSL) during client certificate negotiation. The S8 mode TCB remains associated with the same task for the life of the SSL request. For more information, see Clocks and time stamps.

262 (TYPE-S, 'KY8DISPT', 12 BYTES)
The total elapsed time during which the user task was dispatched by the CICS dispatcher on a CICS Key 8 mode TCB:

- An L8 mode TCB is allocated when a transaction calls an OPENAPI application program defined with EXECKEY=CICS or a task-related user exit program that has been enabled with the OPENAPI option. The TCB remains associated with the task until the transaction is detached.
- An S8 mode TCB is allocated when a transaction is using the secure sockets layer (SSL) during client certificate negotiation. The S8 mode TCB remains associated with the same task for the life of the SSL request.
- A T8 mode TCB is allocated when a transaction is using a JVM server to perform multithreaded processing. When a thread is allocated a T8 mode TCB, that same TCB remains associated with the thread until the processing completes.
- An X8 mode TCB is allocated when a transaction calls a C or C++ program that was compiled with the XPLINK option and that is defined with EXECKEY=CICS. The TCB remains associated with the task until the program ends.

This field is a component of the task dispatch time field, **USRDISPT** (field ID 007 in group DFHTASK).

263 (TYPE-S, 'KY8CPUT', 12 BYTES)
The processor time during which the user task was dispatched by the CICS dispatcher on a CICS Key 8 mode TCB. The usage of the CICS Key 8 mode TCBs is shown in the description for field **KY8DISPT** (field ID 262 in group DFHTASK). This field is a component of the task CPU time field, **USRCPUT** (field ID 008 in group DFHTASK).

264 (TYPE-S, 'KY9DISPT', 12 BYTES)
The total elapsed time during which the user task was dispatched by the CICS dispatcher on a CICS Key 9 mode TCB:

- An L9 mode TCB is allocated when a transaction calls an OPENAPI application program defined with EXECKEY=USER. The TCB remains associated with the task until the transaction is detached.
- An X9 mode TCB is allocated when a transaction calls a C or C++ program that was compiled with the XPLINK option and that is defined with EXECKEY=USER. The TCB remains associated with the task until the program ends.

This field is a component of the task dispatch time field, USRDISPT (field ID 007 in group DFHTASK).

265 (TYPE-S, 'KY9CPUT', 12 BYTES)
The processor time during which the user task was dispatched by the CICS dispatcher on a CICS Key 9 mode TCB. The usage of the CICS Key 9 mode TCBs is shown in the description for field **KY9DISPT** (field ID 264 in group DFHTASK). This field is a component of the task CPU time field, USRCPUT (field ID 008 in group DFHTASK).

266 (TYPE-S, 'L9CPUT', 12 BYTES)
The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS L9 mode TCB. When a transaction calls an OPENAPI application program that is defined with EXECKEY=USER it is allocated and uses a CICS L9 mode TCB. If the storage protection facility is inactive, an L8 mode TCB is used instead of an L9 mode TCB. When a task has been allocated an L9 mode TCB, that same TCB remains associated with the task until the transaction is detached. This field is a component of the total task CPU time field, USRCPUT (field ID 008 in group DFHTASK), and the task key 9 CPU time field, KY9CPUT (field ID 265 in group DFHTASK).

268 (TYPE-S, 'DSTCBMWT', 12 BYTES)
The elapsed time that the user task spent in TCB mismatch waits; that is, waiting because no available TCB matched the request, but at least one non matching TCB was free.

269 (TYPE-S, 'RODISPT', 12 BYTES)
The elapsed time during which the user task was dispatched by the CICS dispatcher on the CICS RO mode TCB. The RO TCB is used for loading programs, unless the command to load the program (EXEC CICS LOAD, XCTL, or LINK) is issued by an application that is currently running on an open TCB. In that situation, the open TCB is used to load the program instead of the RO TCB. The CICS RO mode TCB is also used for opening and closing CICS data sets, issuing RACF® calls, and similar tasks. This field is a component of the task dispatch time field, USRDISPT (group name: DFHTASK, field ID: 007) and the task miscellaneous TCB dispatch time field, MSDISPT (group name: DFHTASK, field ID: 257).

270 (TYPE-S, 'ROCPUT', 12 BYTES)
The processor time during which the user task was dispatched by the CICS dispatcher on the CICS RO mode TCB. The RO TCB is used for loading programs, unless the command to load the program (EXEC CICS LOAD, XCTL, or LINK) is issued by an application that is currently running on an open TCB. In that situation, the open TCB is used to load the program instead of the RO TCB. The CICS RO mode TCB is also used for opening and closing CICS data sets, issuing RACF calls, and similar tasks. This field is a component of the task CPU time field, USRCPUT (group name: DFHTASK, field ID: 008) and the task miscellaneous TCB CPU time field, MSCPUT (group name: DFHTASK, field ID: 258).

271 (TYPE-S, 'X8CPUT', 12 BYTES)
The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS X8 mode TCB. When a transaction calls a C or C++ program that was compiled with the XPLINK option, and that is defined with EXECKEY=CICS, it is allocated and uses a CICS X8 mode TCB. An X8 mode TCB can also be allocated if the program is defined with EXECKEY=USER, but the storage protection facility is inactive. After a task has been allocated an X8 mode TCB, that same TCB remains associated with the task until the program completes. This field is a component of the total task CPU time field, USRCPUT (field ID 008 in group DFHTASK), and the task key 8 CPU time field, KY8CPUT (field ID 263 in group DFHTASK).

272 (TYPE-S, 'X9CPUT', 12 BYTES)
The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS X9 mode TCB. When a transaction calls a C or C++ program that was compiled with the XPLINK option, and that is defined with EXECKEY=USER, it is allocated and uses a CICS X9 mode TCB. (If the

storage protection facility is inactive, an X8 mode TCB is used instead of an X9 mode TCB.) After a task has been allocated an X9 mode TCB, that same TCB remains associated with the task until the program completes. This field is a component of the total task CPU time field, USRCPUT (field ID 008 in group DFHTASK), and the task key 9 CPU time field, KY9CPUT (field ID 265 in group DFHTASK).

273 (TYPE-S, 'JVMITIME', 12 BYTES)
The elapsed time spent initializing the JVM environment. For more information, see Clocks and time stamps.

274 (TYPE-S, 'SMMVSSWT', 12 BYTES)
The time that the user task waited because MVS user region or extended user region was short on storage. For more information, see Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

275 (TYPE-S, 'JVMRTIME', 12 BYTES)
Reserved field, returns zero.

279 (TYPE-S, 'DSMMSCWT', 12 BYTES)
The elapsed time that the user task spent waiting because no TCB was available and a TCB was not created because of MVS storage constraints. This field is not populated. The field is a component of the task suspend time, SUSPTIME (014), field.

281 (TYPE-S, 'MAXSTDLY', 12 BYTES)
The elapsed time for which the user task waited to obtain a CICS SSL TCB (S8 mode), because the CICS system reached the limit set by the system initialization parameter MAXSSLTCBS. The S8 mode open TCBs are used exclusively by secure sockets layer (SSL) pthread requests issued by or on behalf of a user task. For more information, see Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

282 (TYPE-S, 'MAXXTDLY', 12 BYTES)
The elapsed time for which the user task waited to obtain a CICS XP TCB (X8 or X9 mode), because the CICS system reached the limit set by CICS for these types of TCB. The X8 and X9 mode open TCBs are used exclusively by C and C++ programs that were compiled with the XPLINK option. For more information, see Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

283 (TYPE-S, 'MAXTTDLY', 12 BYTES)
The elapsed time for which the user task waited to obtain a T8 TCB, because the CICS system reached the limit of available threads. The T8 mode open TCBs are used by a JVM server to perform multithreaded processing. Each T8 TCB runs under one thread. The thread limit is 2000 for each CICS region and each JVM server in a CICS region can have up to 256 threads. For more information, see Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

285 (TYPE-S, 'PTPWAIT', 12 BYTES)
The elapsed time for which the user task waited for the 3270 bridge partner transaction to complete. For more information, see Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

345 (TYPE-A, 'ICSTACDL', 4 BYTES)
Total length, in bytes, of the data in the containers of all the locally executed START CHANNEL requests issued by the user task. This total includes the length of any headers to the data.

346 (TYPE-A, 'ICSTRCCT', 4 BYTES)
Total number of interval control START CHANNEL requests, to be run on remote systems, issued by the user task.

347 (TYPE-A, 'ICSTRCDL', 4 BYTES)
Total length, in bytes, of the data in the containers of all the remotely executed START CHANNEL requests issued by the user task. This total includes the length of any headers to the data.

348 (TYPE-S, 'ROMODDLY', 12 BYTES)
The elapsed time for which the user task waited for redispatch on the CICS RO TCB. This time is the aggregate of the wait times between each event completion and user-task redispatch. The ROMODDLY field is a component of the task suspend time, SUSPTIME (014), field, and also the redispatch wait, DISPWTT (102), field.

349 (TYPE-S, 'SOMODDLY', 12 BYTES)
The elapsed time for which the user task waited for redispatch on the CICS SO TCB. This time is the aggregate of the wait times between each event completion and user-task redispatch. The SOMODDLY field is a component of the task suspend time, SUSPTIME (014), field, and also the redispatch wait, DISPWTT (102), field.

400 (TYPE-S, 'T8CPUT', 12 BYTES)
The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS T8 mode TCB. T8 mode TCBs are used by a JVM server to perform multithreaded processing. When a thread is allocated a T8 mode TCB, that same TCB remains associated with the thread until the processing completes. This field is a component of the total task CPU time field, USRCPUT (field ID 008 in group DFHTASK), and the task key 8 CPU time field, KY8CPUT (field ID 263 in group DFHTASK).

401 (TYPE-S, 'JVMTHDWT', 12 BYTES)
The elapsed time that the user task waited to obtain a JVM server thread because the CICS system had reached the thread limit for a JVM server in the CICS region. This field is a component of the task suspend time, SUSPTIME (014), field. This does not apply to Liberty JVM servers.

429 (TYPE-S, 'DSAPTHWT', 12 BYTES
The dispatcher allocated pthread wait time. This is the time that the transaction had to wait for a Liberty pthread to be allocated during links to Liberty programs. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

430 (TYPE-C, 'CECMCHTP', 4 BYTES)
The CEC machine type, in EBCDIC, for the physical hardware environment where the CICS region is running. CEC (central electronics complex) is a commonly used synonym for CPC (central processing complex).

431 (TYPE-C, 'CECMDLID', 16 BYTES)
The CEC model number, in EBCDIC, for the physical hardware environment where the CICS region is running.

432 (TYPE-C, 'LPARNAME', 8 BYTES)
The name, in EBCDIC, of the logical partition (LPAR) on the processor where the CICS region is running.

433 (TYPE-A, 'MAXTASKS', 4 BYTES)
The MXT or MAXTASKS value, expressed as a number of tasks, for the CICS region at the time the user task was attached.

434 (TYPE-A, 'CURTASKS', 4 BYTES)
The current number of active user transactions in the system at the time the user task was attached.

435 (TYPE-S, 'XSVFYPWD', 12 BYTES)
The total elapsed time that the user task spent verifying passwords, password phrases, PassTickets, and MFA tokens.

436 (TYPE-S, 'CPUTONCP', 12 BYTES)
The total task processor time on a standard processor for which the user task was dispatched on each CICS TCB under which the task ran.

This field is a component of the task CPU time field, USRCPUT (field ID 008 in group DFHTASK). To calculate the task processor time that was spent on a specialty processor (zIIP or zAAP), subtract the time recorded in the CPUTONCP field from the time recorded in the USRCPUT field.

For a description of TCB modes, see TCB statistics.

437 (TYPE-S, 'OFFLCPUT', 12 BYTES)
The total task processor time that was spent on a standard processor but was eligible for offload to a specialty processor (zIIP or zAAP).

This field is a component of the task CPU time field, USRCPUT (field ID 008 in group DFHTASK), and also a component of the standard CPU time field, CPUTONCP (field ID 436 in group DFHTASK). To calculate the task processor time spent on a standard processor that was not eligible for offload to a

specialty processor, subtract the time recorded in the OFFLCPUT field from the time recorded in the CPUTONCP field.

For a description of TCB modes, see TCB statistics.

438 (TYPE-S, 'XSVFYBAS', 12 BYTES)
The total elapsed time that the user task spent verifying basic authentication tokens (BASICAUTH).

For more information, see VERIFY TOKEN.

439 (TYPE-S, 'XSVFYKER', 12 BYTES)
The total elapsed time that the user task spent verifying Kerberos tokens (KERBEROS).

For more information, see VERIFY TOKEN.

440 (TYPE-S, 'XSVFYJWT', 12 BYTES)
The total elapsed time that the user task spent verifying JSON web tokens (JWT).

For more information, see VERIFY TOKEN.

451 (TYPE-C, 'ACAPPLNM', 64 BYTES)
The 64-character name of the application in the application context data.

452 (TYPE-C, 'ACPLATNM', 64 BYTES)
The 64-character name of the platform in the application context data.

453 (TYPE-A, 'ACMAJVER', 4 BYTES)
The major version of the application in the application context data, expressed as a 4-byte binary value.

454 (TYPE-A, 'ACMINVER', 4 BYTES)
The minor version of the application in the application context data, expressed as a 4-byte binary value.

455 (TYPE-A, 'ACMICVER', 4 BYTES)
The micro version of the application in the application context data, expressed as a 4-byte binary value.

456 (TYPE-C, 'ACOPERNM', 64 BYTES)
The 64-character name of the operation in the application context data.

470 (TYPE-A, 'ASTOTCT', 4 BYTES)
The total number of **EXEC CICS** asynchronous API commands that have been issued by the user task. Includes **RUN TRANSID**, **FETCH CHILD**, **FETCH ANY**, and **FREE CHILD** commands.

471 (TYPE-A, 'ASRUNCT', 4 BYTES)
The number of **EXEC CICS RUN TRANSID** commands that have been issued by the user task.

472 (TYPE-A, 'ASFTCHCT', 4 BYTES)
The number of **EXEC CICS FETCH CHILD** and **EXEC CICS FETCH ANY** commands that have been issued by the user task.

473 (TYPE-A, 'ASFREECT', 4 BYTES)
The number of **EXEC CICS FREE CHILD** commands that have been issued by the user task.

475 (TYPE-S, 'ASFTCHWT', 12 BYTES)
The elapsed time that the user task waited for a child task as a result of issuing an **EXEC CICS FETCH CHILD** or **EXEC CICS FETCH ANY** command which was not completed.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

476 (TYPE-S, 'ASRNATWT', 12 BYTES)
The elapsed time that the user task was delayed as a result of asynchronous child task limits managed by the asynchronous services domain.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference. This field is a component of the task suspend time, SUSPTIME (014), field.

Performance data in group DFHTEMP

Descriptions of the performance data fields in the DFHTEMP group, including the numeric identifer, type, and size of each field.

For a breakdown by individual temporary storage queue of the information provided in group DFHTEMP, you can request transaction resource monitoring. See Chapter 4, “Transaction resource class data: Listing of data fields,” on page 63 for details.

011 (TYPE-S, 'TSIOWTT', 12 BYTES)
Elapsed time for which the user task waited for VSAM temporary storage I/O. For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014), field.

044 (TYPE-A, 'TSGETCT', 4 BYTES)
Number of temporary storage GET requests to auxiliary or main temporary storage issued by the user task.

046 (TYPE-A, 'TSPUTACT', 4 BYTES)
Number of PUT requests to auxiliary temporary storage issued by the user task.

047 (TYPE-A, 'TSPUTMCT', 4 BYTES)
Number of PUT requests to main temporary storage issued by the user task.

092 (TYPE-A, 'TSTOTCT', 4 BYTES)
Total number of temporary storage requests issued by the user task. This field is the sum of the temporary storage READQ (TSGETCT), READQ shared (TSGETSCT), WRITEQ AUX (TSPUTACT), WRITEQ MAIN (TSPUTMCT), WRITEQ shared (TSPUTSCT), and DELETEQ requests issued by the user task.

178 (TYPE-S, 'TSSHWAIT', 12 BYTES)
Elapsed time that the user task waited for an asynchronous shared temporary storage request to a temporary storage data server to complete. For more information, see Clocks and time stamps, and Transaction wait (suspend) times in Reference.

Note: This field is a component of the task suspend time, SUSPTIME (014), field.

460 (TYPE-A, 'TSGETSCT', 4 BYTES)
Number of temporary storage GET requests from shared temporary storage issued by the user task.

461 (TYPE-A, 'TSPUTSCT', 4 BYTES)
Number of temporary storage PUT requests to shared temporary storage issued by the user task.

Performance data in group DFHTERM

Descriptions of the performance data fields in the DFHTERM group, including the numeric identifier, type, and size of each field.

002 (TYPE-C, 'TERM', 4 BYTES)
Terminal or session identification. This field is null if the task is not associated with a terminal or session.

009 (TYPE-S, 'TCIOWTT', 12 BYTES)
Elapsed time for which the user task waited for input from the terminal operator after issuing a RECEIVE request. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

034 (TYPE-A, 'TCMSGIN1', 4 BYTES)
Number of messages received from the principal terminal facility of the task, including LUTYPE6.1 and LUTYPE6.2 (APPC) but not MRO (IRC).

035 (TYPE-A, 'TCMSGOU1', 4 BYTES)
Number of messages sent to the principal terminal facility of the task, including LUTYPE6.1 and LUTYPE6.2 (APPC) but not MRO (IRC).

067 (TYPE-A, 'TCMSGIN2', 4 BYTES)
Number of messages received from the LUTYPE6.1 alternate terminal facilities by the user task.

068 (TYPE-A, 'TCMSGOU2', 4 BYTES)
Number of messages sent to the LUTYPE6.1 alternate terminal facilities by the user task.

069 (TYPE-A, 'TCALLOCT', 4 BYTES)
Number of TCTTE ALLOCATE requests issued by the user task for LUTYPE6.2 (APPC), LUTYPE6.1, and IRC sessions.

083 (TYPE-A, 'TCCHRIN1', 4 BYTES)
Number of characters received from the principal terminal facility of the task, including LUTYPE6.1 and LUTYPE6.2 (APPC) but not MRO (IRC).

084 (TYPE-A, 'TCCHROU1', 4 BYTES)
Number of characters sent to the principal terminal facility of the task, including LUTYPE6.1 and LUTYPE6.2 (APPC) but not MRO (IRC).

085 (TYPE-A, 'TCCHRIN2', 4 BYTES)
Number of characters received from the LUTYPE6.1 alternate terminal facilities by the user task. *(Not applicable to ISC APPC.)*

086 (TYPE-A, 'TCCHROU2', 4 BYTES)
Number of characters sent to the LUTYPE6.1 alternate terminal facilities by the user task. *(Not applicable to ISC APPC.)*

100 (TYPE-S, 'IRIOWTT', 12 BYTES)
Elapsed time for which the user task waited for control at this end of an MRO link. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

111 (TYPE-C, 'LUNAME', 8 BYTES)
The z/OS Communications Server SNA logical unit name (if available) of the terminal that is associated with this transaction. If the task is executing in an application-owning or file-owning region, the LUNAME is the generic applid of the originating connection for MRO, LUTYPE6.1, and LUTYPE6.2 (APPC). The LUNAME is blank if the originating connection is an external CICS interface (EXCI).

133 (TYPE-S, 'LU61WTT', 12 BYTES)
The elapsed time for which the user task waited for I/O on a LUTYPE6.1 connection or session. This time also includes the waits incurred for conversations across LUTYPE6.1 connections, but not the waits incurred because of LUTYPE6.1 syncpoint flows. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

134 (TYPE-S, 'LU62WTT', 12 BYTES)
The elapsed time for which the user task waited for I/O on a LUTYPE6.2 (APPC) connection or session. This time also includes the waits incurred for conversations across LUTYPE6.2 (APPC) connections, but not the waits incurred because of LUTYPE6.2 (APPC) syncpoint flows. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

135 (TYPE-A, 'TCM62IN2', 4 BYTES)
Number of messages received from the alternate facility by the user task for LUTYPE6.2 (APPC) sessions.

136 (TYPE-A, 'TCM62OU2', 4 BYTES)
Number of messages sent to the alternate facility by the user task for LUTYPE6.2 (APPC) sessions.

137 (TYPE-A, 'TCC62IN2', 4 BYTES)
Number of characters received from the alternate facility by the user task for LUTYPE6.2 (APPC) sessions.

138 (TYPE-A, 'TCC62OU2', 4 BYTES)
Number of characters sent to the alternate facility by the user task for LUTYPE6.2 (APPC) sessions.

165 (TYPE-A, 'TERMINFO', 4 BYTES)
Terminal or session information for the principal facility of this task, as identified in the 'TERM' field id 002. This field is null if the task is not associated with a terminal or session facility.

Byte 0
Identifies whether this task is associated with a terminal or session. This field can be set to one of the following values:

X'00'
None

X'01'
Terminal

X'02'
Session

Byte 1
If the principal facility for this task is a session (Byte 0 = x'02'), this field identifies the session type. This field can be set to one of the following values:

X'00'
None

X'01'
IRC

X'02'
IRC XM

X'03'
IRC XCF

X'04'
LU61

X'05'
LU62 Single

X'06'
LU62 Parallel

Byte 2
Identifies the access method defined for the terminal ID or session ID in field TERM. This field can be set to one of the following values:

X'00'
None

X'01'
Communications Server

X'02'
Reserved

X'03'
BSAM

X'04'
Reserved

X'05'
Reserved

X'06'
BGAM

X'07'
CONSOLE

Byte 3
Identifies the terminal or session type for the terminal id or session id in TERM.

- See RDO Typeterm

For a list of the typeterm definitions, see ASSIGN TERMCODE.

169 (TYPE-C, 'TERMCNNM', 4 BYTES)
Terminal session connection name. If the terminal facility associated with this transaction is a session, this field is the name of the owning connection (sysid).

A terminal facility can be identified as a session by using byte 0 of the terminal information, TERMINFO (165), field. If the value is x'02', the terminal facility is a session.

197 (TYPE-C, 'NETID', 8 BYTES)
NETID if a network qualified name has been received from the Communications Server. If it is a resource and the network qualified name has not yet been received, NETID is 8 blanks. In all other cases, it is nulls.

198 (TYPE-C, 'RLUNAME', 8 BYTES)
Real network name if a network qualified name has been received from the Communications Server. In all other cases, this field is the same as LUNAME (field ID 111). For non-Communications Server resources, it is nulls.

343 (TYPE-S, 'TCALWTT', 12 BYTES)
The elapsed time for which a user task waited for an allocate request for an MRO (Inter-Region Communication), LU6.1, or LU6.2 session. For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

This field is a component of the task suspend time, SUSPTIME (014), field.

Performance data in group DFHWEBB

Descriptions of the performance data fields in the DFHWEBB group, including the numeric identifier, type, and size of each field.

224 (TYPE-A, 'WBREADCT', 4 BYTES)
The number of CICS web support READ HTTPHEADER, READ FORMFIELD, and READ QUERYPARM requests issued by the user task.

225 (TYPE-A, 'WBWRITCT', 4 BYTES)
The number of CICS web support WRITE HTTPHEADER requests issued by the user task.

231 (TYPE-A, 'WBRCVCT', 4 BYTES)
The number of CICS web support RECEIVE requests issued by the user task.

232 (TYPE-A, 'WBCHRIN', 4 BYTES)
The number of bytes received by the CICS web support RECEIVE requests issued by the user task.

233 (TYPE-A, 'WBSENDCT', 4 BYTES)
The number of CICS web support SEND requests issued by the user task.

234 (TYPE-A, 'WBCHROUT', 4 BYTES)
The number of bytes sent by the CICS web support SEND requests issued by the user task.

235 (TYPE-A, 'WBTOTCT', 4 BYTES)
The total number of CICS web support requests issued by the user task.

236 (TYPE-A, 'WBREPRCT', 4 BYTES)
The number of reads from the repository in temporary storage issued by the user task.

237 (TYPE-A, 'WBREPWCT', 4 BYTES)
The number of writes to the repository in temporary storage issued by the user task.

238 (TYPE-A, 'WBEXTRCT', 4 BYTES)
The number of CICS web support EXTRACT requests issued by the user task.

239 (TYPE-A, 'WBBRWCT', 4 BYTES)
The number of CICS web support browsing requests for HTTPHEADER, FORMFIELD, and QUERYPARM (STARTBROWSE, READNEXT, and ENDBROWSE) issued by the user task.

331 (TYPE-A, 'WBREDOCT', 4 BYTES)
The number of CICS web support READ HTTPHEADER requests issued by the user task when CICS is an HTTP client.

332 (TYPE-A, 'WBWRTOCT', 4 BYTES)
The number of CICS web support WRITE HTTPHEADER requests issued by the user task when CICS is an HTTP client.

333 (TYPE-A, 'WBRCVIN1', 4 BYTES)
The number of CICS web support RECEIVE and CONVERSE requests issued by the user task when CICS is an HTTP client.

334 (TYPE-A, 'WBCHRIN1', 4 BYTES)
The number of bytes received by the CICS web support RECEIVE and CONVERSE requests issued by the user task when CICS is an HTTP client. This number includes the HTTP headers for the response.

335 (TYPE-A, 'WBSNDOU1', 4 BYTES)
The number of CICS web support SEND and CONVERSE requests issued by the user task when CICS is an HTTP client.

336 (TYPE-A, 'WBCHROU1', 4 BYTES)
The number of bytes sent by the CICS web support SEND and CONVERSE requests issued by the user task when CICS is an HTTP client. This number includes the HTTP headers for the request.

Note: When requests are made using the **WEB CONVERSE** command, the requests increment both the Send and Receive request counts (WBSNDOU1 and WBRCVIN1) and the counts of characters sent and received (WBCHRIN1 and WBCHROU1).

337 (TYPE-A, 'WBPARSCT', 4 BYTES)
The number of CICS web support PARSE URL requests issued by the user task.

338 (TYPE-A, 'WBBRWOCT', 4 BYTES)
The number of CICS web support BROWSE HTTPHEADER requests (STARTBROWSE, READNEXT, and ENDBROWSE) issued by the user task when CICS is an HTTP client.

339 (TYPE-S, 'WBURIOPN', 12 BYTES)
The total elapsed time that the user task was processing **WEB OPEN URIMAP** requests that are issued by the user task. For more information, see Clocks and time stamps.

340 (TYPE-A, 'WBIWBSCT', 4 BYTES)
The number of **EXEC CICS INVOKE SERVICE and EXEC CICS INVOKE WEBSERVICE** requests issued by the user task.

341 (TYPE-A, 'WBREPRDL', 4 BYTES)
The total length, in bytes, of the data read from the repository in temporary storage by the user task.

342 (TYPE-A, 'WBREPWDL', 4 BYTES)
The total length, in bytes, of the data written to the repository in temporary storage by the user task.

380 (TYPE-C, 'WBURIMNM', 8 BYTES)
For CICS web support, Atom feeds, and web service applications, the name of the URIMAP resource definition that was mapped to the URI of the inbound request that was processed by this task.

381 (TYPE-C, 'WBPIPLNM', 8 BYTES)
For web service applications, the name of the PIPELINE resource definition that was used to provide information about the message handlers that act on the service request processed by this task.

382 (TYPE-C, 'WBATMSNM', 8 BYTES)
For Atom feeds, the name of the ATOMSERVICE resource definition that was used to process this task.

383 (TYPE-C, 'WBSVCENM', 32 BYTES)
For web service applications, the name of the WEBSERVICE resource definition that was used to process this task.

384 (TYPE-C, 'WBSVOPNM', 64 BYTES)
For web service applications, the first 64 bytes of the web service operation name.

385 (TYPE-C, 'WBPROGNM', 8 BYTES)
For CICS web support, the name of the program from the URIMAP resource definition that was used to provide the application-generated response to the HTTP request processed by this task.

386 (TYPE-A, 'WBSFCRCT', 4 BYTES)
The number of **EXEC CICS SOAPFAULT CREATE** commands issued by the user task.

387 (TYPE-A, 'WBSFTOCT', 4 BYTES)
The total number of **EXEC CICS SOAPFAULT ADD**, **CREATE**, and **DELETE** commands issued by the user task.

388 (TYPE-A, 'WBISSFCT', 4 BYTES)
The total number of SOAP faults received in response to the **EXEC CICS INVOKE SERVICE** and **EXEC CICS INVOKE WEBSERVICE** commands issued by the user task.

390 (TYPE-A, 'WBSREQBL', 4 BYTES)
For web service applications, the SOAP request body length.

392 (TYPE-A, 'WBSRSPBL', 4 BYTES)
For web service applications, the SOAP response body length.

393 (TYPE-S, 'WBURIRCV', 12 BYTES)
The total elapsed time that the user task was processing **WEB RECEIVE** requests and the receiving side of **WEB CONVERSE** requests that are issued by the user task. The sessions these requests target to are opened by the **WEB OPEN URIMAP** command. For more information, see Clocks and time stamps.

394 (TYPE-S, 'WBURISND', 12 BYTES)
The total elapsed time that the user task was processing **WEB SEND** requests and the sending side of **WEB CONVERSE** requests that are issued by the user task. The sessions these requests target to are opened by the **WEB OPEN URIMAP** command. For more information, see Clocks and time stamps.

412 (TYPE-A, 'MLXSSTDL', 4 BYTES)
The total length of the documents that were parsed using the z/OS XML System Services parser.

413 (TYPE-A, 'MLXMLTCT', 4 BYTES)
The number of **EXEC CICS TRANSFORM** commands issued by the user task.

419 (TYPE-C, 'NJSAPPNM', 32 BYTES)
Node.js application name from which the task was started.

420 (TYPE-A, 'WSACBLCT', 4 BYTES)
The number of **EXEC CICS WSACONTEXT BUILD** commands issued by the user task.

421 (TYPE-A, 'WSACGTCT', 4 BYTES)
The number of **EXEC CICS WSACONTEXT GET** commands issued by the user task.

422 (TYPE-A, 'WSAEPCCT', 4 BYTES)
The number of **EXEC CICS WSAEPR CREATE** commands issued by the user task.

423 (TYPE-A, 'WSATOTCT', 4 BYTES)
The total number of EXEC CICS WS-Addressing commands issued by the user task.

424 (TYPE-A, 'WBJSNRQL', 4 BYTES)
For JSON web service applications, the JSON message request length.

425 (TYPE-A, 'WBJSNRPL', 4 BYTES)
For JSON web service applications, the JSON message response length.

Monitoring fields for URIMAP usage types

Table 6 on page 54 shows which fields in the DFHWEBB group apply to each kind of service provided by URIMAP resource definitions, as determined by the USAGE attribute and other attributes of the URIMAP resource definition.

Table 6. Monitoring fields for URIMAP usage types

Field id	USAGE (PIPELINE): web service	USAGE (ATOM): Atom feed	USAGE (SERVER): CICS web support dynamic response (with program)	USAGE (SERVER): CICS web support static response (with zFS file or document template)	USAGE (CLIENT)
339 WBURIOPN	null	null	null	null	The total elapsed time that the user task that the user task was processing **WEB OPEN URIMAP** requests that are issued by the user task.
380 WBURIMNM	URIMAP resource definition name	URIMAP resource definition name	URIMAP resource definition name	URIMAP resource definition name	null
381 WBPIPLNM	PIPELINE resource definition name	null	null	null	null
382 WBATMSNM	null	ATOMSERVICE resource definition name	null	null	null
383 WBSVCENM	WEBSERVICE resource definition name	null	null	null	null
384 WBSVOPNM	WEBSERVICE operation name	null	null	null	null
385 WBPROGNM	null	null	PROGRAM resource definition name	null	null
393 WBURIRCV	null	null	null	null	The total elapsed time that the user task was processing **WEB RECEIVE** requests and the receiving side of **WEB CONVERSE** requests that are issued by the user task.

Table 6. Monitoring fields for URIMAP usage types (continued)

Field id	USAGE (PIPELINE): web service	USAGE (ATOM): Atom feed	USAGE (SERVER): CICS web support dynamic response (with program)	USAGE (SERVER): CICS web support static response (with zFS file or document template)	USAGE (CLIENT)
394 WBURISND	null	null	null	null	The total elapsed time that the user task was processing **WEB SEND** requests and the sending side of **WEB CONVERSE** requests that are issued by the user task.

Performance data in group DFHWEBC

Descriptions of the performance data fields in the DFHWEBC group, including the numeric identifier, type, and size of each field.

379 (TYPE-S, 'WBSVINVK', 12 BYTES)
The total elapsed time that the user task was processing **INVOKE SERVICE** requests for WEBSERVICEs. For more information, see Clocks and time stamps.

Chapter 3. Exception class data: listing of data fields

The exception class data is listed in the order in which it appears in the exception data section of a monitoring record.

Exception records are fixed format. The format of the exception data section of a monitoring record can be mapped by the DSECT MNEXCDS.

To understand the format of the exception data section, see Exception data sections.

EXCMNTRN (TYPE-C, 4 BYTES)
Transaction identification.

EXCMNTER (TYPE-C, 4 BYTES)
Terminal identification. This field is null if the task is not associated with a terminal or session.

EXCMNUSR (TYPE-C, 8 BYTES)
User identification at task creation. This identifier can also be the remote user identifier for a task created as the result of receiving an ATTACH request across an MRO or APPC link with attach-time security enabled.

EXCMNTST (TYPE-C, 4 BYTES)
Transaction start type. The low-order byte (0 and 1) is set to:

"TO"
Attached from terminal input

"S"
Attached by automatic transaction initiation (ATI) without data

"SD"
Attached by automatic transaction initiation (ATI) with data

"QD"
Attached by transient data trigger level

"U "
Attached by user request

"TP"
Attached from terminal TCTTE transaction ID

"SZ"
Attached by Front End Programming Interface (FEPI)

EXCMNSTA (TYPE-T, 8 BYTES)
Start time of the exception.

EXCMNSTO (TYPE-T, 8 BYTES)
Finish time of the exception.

Note: The performance class exception wait time field, EXWTTIME (103), is a calculation based on subtracting the start time of the exception (EXCMNSTA) from the finish time of the exception (EXCMNSTO).

EXCMNTNO (TYPE-P, 4 BYTES)
Transaction identification number.

EXCMNTPR (TYPE-C, 4 BYTES)
Transaction priority when monitoring was initialized for the task (low-order byte).

EXCMNLUN (TYPE-C, 4 BYTES)
z/OS Communications Server logical unit name (if available) of the terminal associated with this transaction. This field is nulls if the task is not associated with a terminal.

EXCMNEXN (TYPE-A, 4 BYTES)
Exception sequence number for this task.

EXCMNRTY (TYPE-C, 8 BYTES)
Exception resource type. The possible values for EXCMNRTY are shown in Table 7 on page 59.

EXCMNRID (TYPE-C, 8 BYTES)
Exception resource identification. The possible values for EXCMNRID are shown in Table 7 on page 59.

EXCMNTYP (TYPE-A, 2 BYTES)
Exception type. This field can be set to one of the following values:

X'0001'
Exception because of a wait (EXCMNWT)

X'0002'
Exception because of a buffer wait (EXCMNBWT)

X'0003'
Exception because of a string wait (EXCMNSWT)

X'0004'
Exception because a policy rule has triggered (EXCMNPOL)

EXCMNTCN (TYPE-C, 8 BYTES)
Transaction class name. This field is null if the transaction is not in a transaction class.

EXCMNSRV (TYPE-C, 8 BYTES)
z/OS Workload Manager Service Class name for this transaction. This field is null if there are no transaction classification rules defined for CICS subsystems in the active z/OS Workload Manager (WLM) service policy, or if the transaction was WLM-classified in another CICS region.

EXCMNRPT (TYPE-C, 8 BYTES)
z/OS Workload Manager Report Class name for this transaction. This field is null if there are no transaction classification rules defined for CICS subsystems in the active z/OS Workload Manager (WLM) service policy, or if the transaction was WLM-classified in another CICS region.

EXCMNNPX (TYPE-C, 20 BYTES)
Fully qualified name by which the originating system is known to the z/OS Communications Server network. This name is assigned at attach time using either the NETNAME derived from the TCT (when the task is attached to a local terminal), or the NETNAME passed as part of an ISC APPC or IRC attach header. At least three passing bytes (X'00') are present at the right end of the name.

If the originating terminal is a z/OS Communications Server device across an ISC APPC or IRC link, the NETNAME is the *networkid.LUname*. If the terminal is not a z/OS Communications Server device, the NETNAME is *networkid.generic_applid*.

All originating information passed as part of an ISC LUTYPE6.1 attach header has the same format as the non-z/OS Communications Server terminal originators.

When the originator is communicating over an external CICS interface (EXCI) session, the name is a concatenation of the following information that is derived from the originating system:

```
'DFHEXCIU  |  .       |   MVS Id   |   Address space Id (ASID)'
 8 bytes   |  1 byte  |   4 bytes  |   4 bytes
```

That is, the name is a 17-byte LU name that consists of the following:

- An 8-byte eye-catcher set to 'DFHEXCIU'.
- A 1-byte field containing a period (.).
- A 4-byte field containing the MVS ID, in characters, under which the client program is running.
- A 4-byte field containing the address space ID (ASID) in which the client program is running. This field contains the 4-character EBCDIC representation of the 2-byte hex address space ID.

EXCMNNSX (TYPE-C, 8 BYTES)
Name by which the unit of work is known within the originating system. This last name is assigned at attach time, either by using an STCK-derived token (when the task is attached to a local terminal), or the unit of work ID is passed as part of an ISC APPC or IRC attach header.

The first six bytes of this field are a binary value derived from the clock of the originating system and wrapping round at intervals of several months. The last two bytes of this field are for the period count. These bytes can change during the life of the task as a result of syncpoint activity.

Note: When MRO or ISC is used, the EXCMNNSX field must be combined with the EXCMNNPX field to uniquely identify a task, because the EXCMNNSX field is unique only to the originating CICS system.

EXCMNTRF (TYPE-C, 8 BYTES)
Transaction flags; a string of 64 bits used for signaling transaction definition and status information. For details, see field 164 (TRANFLAG) in performance data group DFHTASK.

EXCMNFCN (TYPE-C, 4 BYTES)
Transaction facility name. This field is null if the transaction is not associated with a facility. The transaction facility type (if any) can be identified by using byte 0 of the transaction flags field, EXCMNTRF.

EXCMNCPN (TYPE-C, 8 BYTES)
The name of the currently running program for this user task when the exception condition occurred.

EXCMNBTR (TYPE-C, 4 BYTES)
3270 Bridge transaction identification.

EXCMNURI (TYPE-C, 16 BYTES)
RRMS/MVS unit-of-recovery ID (URID).

EXCMNRIL (TYPE-A, 4 BYTES)
Exception resource ID length.

EXCMNRIX (TYPE-C, 256 BYTES)
Exception resource ID (extended).

EXCMNNID (TYPE-C, 8 BYTES)
NETID if a network qualified name has been received from z/OS Communications Server. For a z/OS Communications Server resource when the network qualified name has not yet been received, NETID is eight blanks. In all other cases, this field is nulls.

EXCMNRLU (TYPE-C, 8 BYTES)
Real network name if a network qualified name has been received from z/OS Communications Server. In all other cases, this field is the same as LUNAME (field id 111). For non-z/OS Communications Server resources, this field is nulls.

The following table shows the value and relationships of the fields EXCMNTYP, EXCMNRTY, and EXCMNRID.

Table 7. Possible values of EXCMNTYP, EXCMNRTY, and EXCMNRID. The relationship between exception type, resource type, and resource identification.

EXCMNTYP Exception type	EXCMNRTY Resource type	EXCMNRID Resource ID	MEANING
EXCMNPOL	'AID'	rule_id 1	All conditions of a AID threshold policy system rule have been met
EXCMNPOL	'ASYNC'	rule_id 1	The threshold of a AYSNC request policy task rule has been exceeded
EXCMNPOL	'BUNDLE'	rule_id 1	All conditions of a bundle enable status or bundle available status policy system rule have been met
EXCMNPOL	'CONTAINR'	rule_id 1	The threshold of a container storage policy task rule has been exceeded.
EXCMNPOL	'DATABASE'	rule_id 1	The threshold of a database request policy task rule has been exceeded
EXCMNPOL	'DBCTLCON'	rule_id 1	All conditions of a DBCTL connection status policy system rule have been met

Table 7. Possible values of EXCMNTYP, EXCMNRTY, and EXCMNRID. The relationship between exception type, resource type, and resource identification. *(continued)*

EXCMNTYP Exception type	EXCMNRTY Resource type	EXCMNRID Resource ID	MEANING
EXCMNPOL	'DB2CONN'	rule_id 1	All conditions of a DB2 connection status policy system rule have been met
EXCMNPOL	'EXECCICS'	rule_id 1	The threshold of a EXEC CICS request policy task rule has been exceeded
EXCMNPOL	'FILE'	rule_id 1	All conditions of a file enable status or file open status policy system rule have been met or the threshold of a file request policy task rule has been exceeded
EXCMNPOL	'IPCONN'	rule_id 1	All conditions of a IPCONN connection status policy system rule have been met
EXCMNPOL	'MESSAGE' 2	rule_id 1	All conditions of a message policy system rule have been met
EXCMNPOL	'MQ'	rule_id 1	The threshold of a IBM MQ request policy task rule has been exceeded
EXCMNPOL	'MQCONN'	rule_id 1	All conditions of a MQ connection status policy system rule have been met
EXCMNPOL	'MROCONN'	rule_id 1	All conditions of a MRO connection status policy system rule have been met
EXCMNPOL	'NAMECTR'	rule_id 1	The threshold of a named counter request policy task rule has been exceeded
EXCMNPOL	'PIPELINE'	rule_id 1	All conditions of a pipeline enable status policy system rule have been met
EXCMNPOL	'PROGRAM'	rule_id 1	All conditions of a program enable status policy system rule have been met or the threshold of a program request policy task rule has been exceeded
EXCMNPOL	'START'	rule_id 1	The threshold of a start request policy task rule has been exceeded
EXCMNPOL	'STORAGE'	rule_id 1	The threshold of a storage or storage request policy task rule has been exceeded
EXCMNPOL	'SYNCPT'	rule_id 1	The threshold of a syncpoint request policy task rule has been exceeded
EXCMNPOL	'TASK' 2	rule_id 1	All conditions of a task or tclass threshold policy system rule have been met
EXCMNPOL	'TDQUEUE'	rule_id 1	The threshold of a TD Queue request policy task rule has been exceeded
EXCMNPOL	'TIME'	rule_id 1	The threshold of a time policy task rule has been exceeded
EXCMNPOL	'TRANDUMP'	rule_id 1	The threshold of a transaction dump threshold policy system rule has been exceeded
EXCMNPOL	'TRANID'	rule_id 1	All conditions of a unhandled transaction abend policy system rule have been met

Table 7. Possible values of EXCMNTYP, EXCMNRTY, and EXCMNRID. The relationship between exception type, resource type, and resource identification. *(continued)*

EXCMNTYP Exception type	EXCMNRTY Resource type	EXCMNRID Resource ID	MEANING
EXCMNPOL	'TSQUEUE'	rule_id 1	The threshold of a TS Queue bytes or TS Queue request policy task rule has been exceeded
EXCMNWT	'CFDTLRSW'	poolname	Wait for coupling facility data tables locking (request) slot
EXCMNWT	'CFDTPOOL'	poolname	Wait for coupling facility data tables non-locking (request) slot
EXCMNWT	'STORAGE'	'UDSA'	Wait for UDSA storage
EXCMNWT	'STORAGE'	'EUDSA'	Wait for EUDSA storage
EXCMNWT	'STORAGE'	'CDSA'	Wait for CDSA storage
EXCMNWT	'STORAGE'	'ECDSA'	Wait for ECDSA storage
EXCMNWT	'STORAGE'	'SDSA'	Wait for SDSA storage
EXCMNWT	'STORAGE'	'ESDSA'	Wait for ESDSA storage
EXCMNWT	'STORAGE'	'RDSA'	Wait for RDSA storage
EXCMNWT	'STORAGE'	'ERDSA'	Wait for ERDSA storage
EXCMNWT	'STORAGE'	'GCDSA'	Wait for GCDSA storage
EXCMNWT	'STORAGE'	'GUDSA'	Wait for GUDSA storage
EXCMNWT	'STORAGE'	'GSDSA'	Wait for GSDSA storage
EXCMNWT	'TEMPSTOR'	TS Qname	Wait for temporary storage
EXCMNSWT	'FILE'	filename	Wait for string associated with file
EXCMNSWT	'LSRPOOL'	filename	Wait for string associated with LSRPOOL
EXCMNSWT	'TEMPSTOR'	TS Qname	Wait for string associated with DFHTEMP
EXCMNBWT	'LSRPOOL'	LSRPOOL	Wait for buffer associated with LSRPOOL
EXCMNBWT	'TEMPSTOR'	TS Qname	Wait for buffer associated with DFHTEMP

Note:

1 **'rule_id'**
The name of a rule_id is a concatenation of the bundle_id, policy name, and rule name: **<bundle id>.<policy name>.<rule name>**.

EXCMNRID contains the first 8 characters of the rule_id, and the full name is in EXCMNRIX.

2 **'MESSAGE' and 'TASK'**
The filtering for the following policy system rules is deferred to the long running system task CMPE, which is attached during CICS initialization. To get the MN exception records emitted to SMF, ensure to set **MNEXC=ON** in the system initialization table (SIT); setting it at a later time when CICS is running (for example, by using **CEMT SET MONITOR**) does not work, because CMPE is already attached before exception class monitoring is enabled.

- Message
- Transaction class tasks
- User tasks

Chapter 4. Transaction resource class data: Listing of data fields

The transaction resource class data is listed in the order in which it appears in the transaction resource data section of a monitoring record.

All the transaction resource data records produced by a single CICS run have the same format, with a resource record header followed by a resource data section for each resource being monitored. The format of the transaction resource data section of a monitoring record can be mapped by the DSECT DFHMNRDS.

To understand the format of the transaction resource data section, see Transaction resource data sections.

Header fields

These fields are the transaction header fields in a transaction resource monitoring record.

MNR_ID_TRANID (TYPE-C, 4 BYTES)
Transaction identifier.

MNR_ID_TERMID (TYPE-C, 4 BYTES)
Terminal identifier. This identification field is null if the task is not associated with a terminal or session.

MNR_ID_USERID (TYPE-C, 8 BYTES)
User identification at task creation. This can also be the remote user identifier for a task created as the result of receiving an ATTACH request across an MRO or APPC link with attach-time security enabled.

MNR_ID_STYPE (TYPE-C, 4 BYTES)
Transaction start type. The high-order byte (0 and 1) can have one of the following values:

"TO"
Attached from terminal input

"S "
Attached by automatic transaction initiation (ATI) without data

"SD"
Attached by automatic transaction initiation (ATI) with data

"QD"
Attached by the transient data trigger level

"U "
Attached by a user request

"TP"
Attached from a terminal TCTTE transaction ID

"SZ"
Attached by the Front End Programming Interface (FEPI).

MNR_ID_START (TYPE-T, 8 BYTES)
Start time of the transaction.

MNR_ID_STOP (TYPE-T, 8 BYTES)
Stop time of the transaction.

MNR_ID_TASKNO (TYPE-A, 4 BYTES)
The transaction identification number (the task number allocated to the transaction at task attach).

MNR_ID_LUNAME (TYPE-C, 8 BYTES)
z/OS Communications Server logical unit name (if available) of the terminal associated with this transaction. If the task is running in an application-owning or file-owning region, the LUNAME is the generic applid of the originating connection for MRO, LUTYPE6.1, and LUTYPE6.2 (APPC). The LUNAME is blank if the originating connection is an external CICS interface (EXCI).

MNR_ID_PGMNAME (TYPE-C, 8 BYTES)
The name of the first program invoked at attach-time. For more information, see performance data field 071 (PGMNAME) in the DFHPROG group.

MNR_ID_UOW_PX (TYPE-C, 20 BYTES)
This field contains the same information as the performance data field NETUOWPX. For the details, see performance data field 097 (NETUOWPX) in the DFHTASK group.

MNR_ID_UOW_SX (TYPE-C, 8 BYTES)
This field contains the same information as the performance class data field NETUOWSX. For the details, see performance data field 098 (NETUOWSX) in the DFHTASK group.

MNR_ID_RSYSID (TYPE-C, 4 BYTES)
The name (system ID) of the remote system to which this transaction was routed, either statically or dynamically. For more information, see performance data field 130 (RSYSID) in the DFHCICS group.

MNR_ID_TRN_FLAGS (TYPE-A, 8 BYTES)
Transaction flags, a string of 64 bits used for signaling transaction definition and status information. For the details, see performance data field 164 (TRANFLAG) in the DFHTASK group.

MNR_ID_FCTYNAME (TYPE-C, 4 BYTES)
Transaction facility name. This field is null if the transaction is not associated with a facility.
You can identify the transaction facility type (if any) using byte 0 of the transaction flags (MNR_ID_TRN_FLAGS) field. For details, see performance data field 163 (FCTYNAME) in the DFHTASK group.

MNR_ID_RTYPE (TYPE-C, 4 BYTES)
Transaction resource monitoring record type (low-order byte-3). Currently this record type can have only one value, T, indicating a record output for task termination. For more information about record types, see performance data field 112 (RTYPE) in the DFHCICS group.

MNR_ID_TERMINFO (TYPE-A, 4 BYTES)
Terminal or session information for the task principal facility. For more information about terminal information, see performance data field 165 (TERMINFO) in the DFHTERM group.

MNR_ID_TERMCNNM (TYPE-C, 4 BYTES)
Terminal session connection name. If the terminal facility associated with this transaction is a session, this field is the name of the owning connection (system ID). For more information, see performance data field 169 (TERMCNNM) in the DFHTERM group.

MNR_ID_RES_FLAGS (TYPE-A, 4 BYTES)
Resource flags, a string of 32 bits used for signaling resource status information.

Byte 0
Resource status information:

Bit 0
The maximum number of files to be monitored (defined in the MCT) has been exceeded by the transaction (X'80')

Bit 1
The maximum number of temporary storage queues to be monitored (defined in the MCT) has been exceeded by the transaction (X'40')

Bit 2
The maximum number of distributed program link requests to be monitored (defined in the MCT) has been exceeded by the transaction (X'20')

Bits 3-7
Reserved.

Bytes 1-3
Reserved.

MNR_ID_ISIPICNM (TYPE-C, 8 BYTES)
The name of the IPIC (IPCONN) entry of the TCP/IP service that attached the user task. For more information, see field 305 (ISIPICNM) in the DFHSOCK performance-class data group.

MNR_ID_CLIPADDR (TYPE-C, 40 BYTES)
The IP address of the originating client or Telnet client. For more information, see field 318 (CLIPADDR) in the DFHSOCK performance-class data group.

MNR_ID_ORIGIN_NETWKID (TYPE-C, 8 BYTES)
The network identifier from which this work request (transaction) originated. For more information, see field 359 (ONETWKID) in the DFHCICS performance data group.

MNR_ID_ORIGIN_APPLID (TYPE-C, 8 BYTES)
The applid of the CICS region where this work request (transaction) originated; for example, the region in which the CWXN task ran. For more information, see field 360 (OAPPLID) in the DFHCICS performance data group.

MNR_ID_ORIGIN_ATT_TIME (TYPE-T, 8 BYTES)
The time when the originating task, for example, the CWXN task, was started. For more information, see field 361 (OSTART) in the DFHCICS performance data group.

MNR_ID_ORIGIN_TRANNUM (TYPE-P, 4 BYTES)
The number of the originating task; for example, the CWXN task. For more information, see field 362 (OTRANNUM) in the DFHCICS performance data group.

MNR_ID_ORIGIN_TRANID (TYPE-C, 4 BYTES)
The transaction ID (TRANSID) of the originating task; for example, the CWXN task. For more information, see field 363 (OTRAN) in the DFHCICS performance data group.

MNR_ID_ORIGIN_USERID (TYPE-C, 8 BYTES)
The originating Userid-2 or Userid-1, for example, from CWBA, depending on the originating task. For more information, see field 364 (OUSERID) in the DFHCICS performance data group.

MNR_ID_ORIGIN_USER_CORR (TYPE-C, 64 BYTES)
The originating user correlator. For more information, see field 365 (OUSERCOR) in the DFHCICS performance data group.

MNR_ID_ORIGIN_TCPIPSERV (TYPE-C, 8 BYTES)
The name of the originating TCPIPSERVICE. For more information, see field 366 (OTCPSVCE) in the DFHCICS performance data group.

MNR_ID_ORIGIN_PORTNUM (TYPE-A, 4 BYTES)
The port number used by the originating TCPIPSERVICE. For more information, see field 367 (OPORTNUM) in the DFHCICS performance data group.

MNR_ID_ORIGIN_CLIPADDR (TYPE-C, 40 BYTES)
The IP address of the originating client or Telnet client. For more information, see field 372 (OCLIPADR) in the DFHCICS performance data group.

MNR_ID_ORIGIN_CLIPPORT (TYPE-A, 4 BYTES)
The TCP/IP port number of the originating client or Telnet client. For more information, see field 369 (OCLIPORT) in the DFHCICS performance data group.

MNR_ID_ORIGIN_TRANFLAG (TYPE-A, 8 BYTES)
The originating transaction flags. This 64-bit string is used for signaling transaction definition and status information. For more information, see field 370 (OTRANFLG) in the DFHCICS performance data group.

MNR_ID_ORIGIN_FCTYNAME (TYPE-C, 8 BYTES)
The facility name of the originating transaction. If the originating transaction is not associated with a facility, this field is null. For more information, see field 371 (OFCTYNME) in the DFHCICS performance data group.

MNR_PHD_NTWKID (TYPE-C, 8 BYTES)
The network identifier of the CICS system of an immediately previous task in another CICS region with which this task is associated. For more information, see field 373 (PHNTWKID) in the DFHCICS performance data group.

MNR_PHD_APPLID (TYPE-C, 8 BYTES)
The APPLID from previous hop data. This is the APPLID of the CICS system of a previous task in another CICS system with which this task is associated. For more information, see field 374 (PHAPPLID) in the DFHCICS performance data group. For more information about previous hop data, see Previous hop data characteristics.

MNR_PHD_ATTACH_TIME (TYPE-T, 8 BYTES)
The start time of the immediately previous task in another CICS system with which this task is associated. For more information, see field 375 (PHSTART) in the DFHCICS performance data group.

MNR_PHD_TRANNUM (TYPE-P, 4 BYTES)
The task number of the immediately previous task in another CICS system with which this task is associated. For more information, see field 376 (PHTRANNO) in the DFHCICS performance data group.

MNR_PHD_TRANID (TYPE-C, 4 BYTES)
The transaction ID (TRANSID) of the immediately previous task in another CICS system with which this task is associated. For more information, see field 377 (PHTRAN) in the DFHCICS performance data group.

MNR_PHD_COUNT (TYPE-A, 4 BYTES)
The number of times there has been a request from one CICS system to another CICS region to initiate a task with which this task is associated. For more information, see field 378 (PHCOUNT) in the DFHCICS performance data group.

MNR_PTD_ATTACH_TIME (TYPE-T, 8 BYTES)
The start time of the immediately previous or parent task in the same CICS system with which this task is associated. For more information, see field 480 (PTSTART) in the DFHCICS performance data group.

MNR_PTD_TRANNUM (TYPE-P, 4 BYTES)
The task number of the immediately previous or parent task in the same CICS system with which this task is associated. For more information, see field 481 (PTTRANNO) in the DFHCICS performance data group.

MNR_PTD_TRANID (TYPE-C, 4 BYTES)
The transaction ID (TRANSID) of the immediately previous or parent task in the same CICS system with which this task is associated. For more information, see field 482 (PTTRAN) in the DFHCICS performance data group.

MNR_PTD_COUNT (TYPE-A, 4 BYTES)
The number of times there has been a request from one task to initiate another task in the same CICS system with which this task is associated, such as by an **EXEC CICS RUN TRANSID** or **START** command. For more information, see field 483 (PTCOUNT) in the DFHCICS performance data group.

MNR_ID_TRNGRPID (TYPE-C, 28 BYTES)
The transaction group ID of the originating task.

File entry fields

These fields are in each file entry in a transaction resource monitoring record.

For information about transaction file accesses in performance class monitoring data, see DFHFILE group.

MNR_FILE_NAME (TYPE-C, 8 BYTES)
The CICS 8-character name of the file to which the following data fields refer.

MNR_FILE_GET (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of GET requests issued by the user task for this file. The count part of this field (the low-order 24 bits) contains the number of GET requests issued against the file.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_FILE_PUT (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of PUT requests issued by the user task for this file. The count part of this field (the low-order 24 bits) contains the number of PUT requests issued against the file.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_FILE_BRWSE (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of BROWSE requests issued by the user task for this file. The count part of this field (the low-order 24 bits) contains the number of BROWSE requests issued against the file.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_FILE_ADD (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of ADD requests issued by the user task for this file. The count part of this field (the low-order 24 bits) contains the number of ADD requests issued against the file.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_FILE_DEL (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of DELETE requests issued by the user task for this file. The count part of this field (the low-order 24 bits) contains the number of DELETE requests issued against the file.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_FILE_TOTAL (TYPE-S, 8 BYTES)
The total elapsed time that the user task waited for completion of all requests issued by the user task for this file. The count part of this field (the low-order 24 bits) contains the number of all requests issued against the file.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_FILE_AM_RQ (TYPE-A, 4 BYTES)
Number of times the user task called file access-method interfaces. See also performance data field 070 (FCAMCT) in the DFHFILE group.

MNR_FILE_IO_WT (TYPE-S, 8 BYTES)
The total I/O wait time on this file. The count part of this field (the low-order 24 bits) contains the number of requests issued against the file which waited for I/O.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_RLS_FILE_IO_WT (TYPE-S, 8 BYTES)
The elapsed time in which the user task waited for RLS file I/O on this file. The count part of this field (the low-order 24 bits) contains the number of requests issued against the RLS file which waited for I/O.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_CFDT_IO_WT (TYPE-S, 8 BYTES)
The elapsed time in which the user task waited for a data table access request to the coupling facility data table server to complete for this file. The count part of this field (the low-order 24 bits) contains the number of requests to the coupling facility data table server for the file.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

Temporary storage queue entry fields

These fields are in each temporary storage queue entry in a transaction resource monitoring record.

For information about transaction temporary storage queue accesses in performance class monitoring data, see DFHTEMP group.

MNR_TSQUEUE_NAME (TYPE-C, 16 BYTES)
The CICS 16-character name of the temporary storage queue to which the following data fields refer.

MNR_TSQUEUE_GET (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of GET requests issued by the user task for this temporary storage queue. The count part of this field (the low-order 24 bits) contains the number of GET requests issued against the temporary storage queue.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_TSQUEUE_PUT_AUX (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of PUT requests to auxiliary temporary storage, issued by the user task for this temporary storage queue. The count part of this field (the low-order 24 bits) contains the number of PUT requests to auxiliary temporary storage issued against the temporary storage queue.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_TSQUEUE_PUT_MAIN (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of PUT requests to main temporary storage, issued by the user task for this temporary storage queue. The count part of this field (the low-order 24 bits) contains the number of PUT requests to main temporary storage issued against the temporary storage queue.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_TSQUEUE_TOTAL (TYPE-S, 8 BYTES)
The total elapsed time that the user task waited for completion of all requests issued by the user task for this temporary storage queue. The count part of this field (the low-order 24 bits) contains the number of all requests issued against the temporary storage queue.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_TSQUEUE_GET_ITEML (TYPE-A, 4 BYTES)
The total length of all items obtained from this temporary storage queue.

MNR_TSQUEUE_PUT_AUX_ITEML (TYPE-A, 4 BYTES)
The total length of all items written to the auxiliary temporary storage queue.

MNR_TSQUEUE_PUT_MAIN_ITEML (TYPE-A, 4 BYTES)
The total length of all items written to the main temporary storage queue.

MNR_TSQUEUE_IO_WT (TYPE-S, 8 BYTES)
The total I/O wait time on this temporary storage queue. The count part of this field (the low-order 24 bits) contains the number of requests issued against the temporary storage queue which waited for I/O.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_SHR_TSQUEUE_IO_WT (TYPE-S, 8 BYTES)
The total I/O wait time on the shared temporary storage queue. The count part of this field (the low-order 24 bits) contains the number of requests issued against the shared temporary storage queue which waited for I/O.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_TSQUEUE_GET_SHR (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of GET requests to shared temporary storage, issued by the user task for this temporary storage queue. The count part of this field (the low-order 24 bits) contains the number of GET requests to shared temporary storage issued against the temporary storage queue.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_TSQUEUE_PUT_SHR (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for completion of PUT requests to shared temporary storage, issued by the user task for this temporary storage queue. The count part of this field (the

low-order 24 bits) contains the number of PUT requests to shared temporary storage issued against the temporary storage queue.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_TSQUEUE_GET_SHR_ITEML (TYPE-A, 4 BYTES)
The total length of all items obtained from this shared temporary storage queue.

MNR_TSQUEUE_PUT_SHR_ITEML (TYPE-A, 4 BYTES)
The total length of all items written to this shared temporary storage queue.

DPL entry fields

These fields are in each distributed program link entry in a transaction resource monitoring record.

For information about transaction program accesses in performance class monitoring data, see DFHPROG group.

MNR_DPL_PROGRAM_NAME (TYPE-C, 8 BYTES)
The name of the program to which the following data fields refer.

MNR_DPL_SYSID (TYPE-C, 4 BYTES)
The name of the remote system to which this program was routed for the distributed program link.

MNR_DPL_LINK_REQS (TYPE-C, 4 BYTES)
The number of distributed program link requests issued by the user task for this program and sysid combination.

URIMAP entry fields

These fields are in each URIMAP entry in a transaction resource monitoring record.

For information about URIMAP accesses in performance class monitoring data, see DFHWEBB group.

MNR_URIMAP_NAME (TYPE-C, 8 BYTES)
The CICS 8-character name of the URIMAP to which the following data fields refer. The URIMAP is a client URIMAP that is used on the **WEB OPEN** command issued by the user task.

MNR_URIMAP_CIPHER (TYPE-A, 4 BYTES)
The code for the cipher suite, if any, that was selected during the SSL handshake for use on the connection established by **WEB OPEN URIMAP**, for example X'0000002F'.

MNR_URIMAP_WEBOPEN (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for the completion of **WEB OPEN URIMAP** requests that are issued by the user task for this URIMAP. The count part of this field (the low-order 24 bits) contains the number of **WEB OPEN** requests that are issued against the URIMAP.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_URIMAP_WEBRECV (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for the completion of **WEB RECEIVE** requests and the receiving side of **WEB CONVERSE** requests that are issued by the user task against a session token for this URIMAP. The count part of this field (the low-order 24 bits) contains the number of **WEB RECEIVE** and **WEB CONVERSE** requests issued.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

MNR_URIMAP_WEBSEND (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for the completion of **WEB SEND** requests and the sending side of **WEB CONVERSE** requests that are issued by the user task against a session token for this URIMAP. The count part of this field (the low-order 24 bits) contains the number of **WEB SEND** and **WEB CONVERSE** requests issued.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

WEBSERVICE entry fields

These fields are in each WEBSERVICE entry in a transaction resource monitoring record.

For information about WEBSERVICE accesses in performance class monitoring data, see DFHWEBC group.

MNR_WEBSVC_NAME (TYPE-C, 32 BYTES)
The name of the WEBSERVICE resource definition that was used for the **INVOKE SERVICE** command issued by the user task.

MNR_WEBSVC_PIPE (TYPE-C, 8 BYTES)
The name of the PIPELINE resource definition that was used to provide information about the message handlers that acted on the service request processed by this task.

MNR_WEBSVC_INVK (TYPE-S, 8 BYTES)
The elapsed time that the user task waited for the completion of **INVOKE SERVICE** requests issued by the user task for this WEBSERVICE. The count part of this field (the low-order 24 bits) contains the number of **INVOKE SERVICE** requests issued against the WEBSERVICE.

For more information, see Clocks and time stamps and Transaction wait (suspend) times in Reference.

Chapter 5. Identity class data: Listing of data fields

The identity class data is listed in the order in which it appears in the identity class data section of a monitoring record.

All the identity class data records produced by a single CICS run have the same format, with an identity record header followed by an identity data section for each transaction being monitored. The format of the identity class data section of a monitoring record can be mapped by the DSECT DFHMNIDS.

To understand the format of the identity class data section, see Identity class data sections.

Header fields

These fields are the header fields in an identity class monitoring record.

MNI_ID_TRANID (TYPE-C, 4 BYTES)
Transaction identifier.

MNI_ID_TERMID (TYPE-C, 4 BYTES)
Terminal identifier. This identification field is null if the task is not associated with a terminal or session.

MNI_ID_USERID (TYPE-C, 8 BYTES)
User identification at task creation, or the remote user identifier for a task that is created as the result of receiving an ATTACH request across an MRO or APPC link with attach-time security enabled.

MNI_ID_STYPE (TYPE-C, 4 BYTES)
Transaction start type. The high-order bytes (0 and 1) can have one of the following values:

"TO"
Attached from terminal input

"S "
Attached by automatic transaction initiation (ATI) without data

"SD"
Attached by automatic transaction initiation (ATI) with data

"QD"
Attached by the transient data trigger level

"U "
Attached by a user request

"TP"
Attached from a terminal TCTTE transaction ID

"SZ"
Attached by the Front End Programming Interface (FEPI)

MNI_ID_START (TYPE-T, 8 BYTES)
Start time of the transaction.

MNI_ID_STOP (TYPE-T, 8 BYTES)
Stop time of the transaction.

MNI_ID_TASKNO (TYPE-A, 4 BYTES)
The transaction identification number (the task number allocated to the transaction at task attach).

MNI_ID_LUNAME (TYPE-C, 8 BYTES)
z/OS Communications Server logical unit name (if available) of the terminal associated with this transaction. If the task is running in an application-owning or file-owning region, the LUNAME is the generic applid of the originating connection for MRO, LUTYPE6.1, and LUTYPE6.2 (APPC). The LUNAME is blank if the originating connection is an external CICS interface (EXCI).

MNI_ID_PGMNAME (TYPE-C, 8 BYTES)
The name of the first program called at attach-time. For more information, see field 071 (PGMNAME) in the DFHPROG performance data group.

MNI_ID_UOW_PX (TYPE-C, 20 BYTES)
This field contains the same information as the performance class data field NETUOWPX. See NETUOWPX, in group DFHTASK for details.

MNI_ID_UOW_SX (TYPE-C, 8 BYTES)
This field contains the same information as the performance class data field NETUOWSX. See NETUOWSX, in group DFHTASK for details.

MNI_ID_RSYSID (TYPE-C, 4 BYTES)
The name (system ID) of the remote system to which this transaction was routed, either statically or dynamically. For more information, see field 130 (RSYSID) in the DFHCICS performance data group.

MNI_ID_TRN_FLAGS (TYPE-A, 8 BYTES)
Transaction flags, a string of 64 bits used for signaling transaction definition and status information. For details, see field 164 (TRANFLAG) in the DFHTASK performance data group.

MNI_ID_FCTYNAME (TYPE-C, 4 BYTES)
Transaction facility name. This field is null if the transaction is not associated with a facility. You can identify the transaction facility type (if any) using byte 0 of the transaction flags (MNR_ID_TRN_FLAGS) field. For details, see field 163 (FCTYNAME) in the DFHTASK performance data group.

MNI_ID_RTYPE (TYPE-C, 4 BYTES)
Transaction resource monitoring record type (low-order byte 3). Currently this record type can have only one value, T, indicating a record produced for task termination. For more information about record types, see field 112 (RTYPE) the DFHCICS performance data group.

MNI_ID_TERMINFO (TYPE-A, 4 BYTES)
Terminal or session information for the task principal facility. For more information about terminal information, see field 165 (TERMINFO) in the DFHTERM performance data group.

MNI_ID_TERMCNNM (TYPE-C, 4 BYTES)
Terminal session connection name. If the terminal facility associated with this transaction is a session, this field is the name of the owning connection (system ID). For more information, see field 169 (TERMCNNM) in the DFHTERM performance data group.

MNI_ID_ISIPICNM (TYPE-C, 8 BYTES)
The name of the IPIC (IPCONN) entry of the TCP/IP service that attached the user task. For more information, see field 305 (ISIPICNM) in the DFHSOCK performance-class data group.

MNI_ID_CLIPADDR (TYPE-C, 40 BYTES)
The IP address of the originating client or Telnet client. For more information, see field 318 (CLIPADDR) in the DFHSOCK performance-class data group.

MNI_ID_ORIGIN_NETWKID (TYPE-C, 8 BYTES)
The network identifier from which this work request (transaction) originated. For more information, see field 359 (ONETWKID) in the DFHCICS performance data group.

MNI_ID_ORIGIN_APPLID (TYPE-C, 8 BYTES)
The applid of the CICS region where this work request (transaction) originated; for example, the region in which the CWXN task ran. For more information, see field 360 (OAPPLID) in the DFHCICS performance data group.

MNI_ID_ORIGIN_ATT_TIME (TYPE-T, 8 BYTES)
The time when the originating task, for example, the CWXN task, was started. For more information, see field 361 (OSTART) in the DFHCICS performance data group.

MNI_ID_ORIGIN_TRANNUM (TYPE-P, 4 BYTES)
The number of the originating task; for example, the CWXN task. For more information, see field 362 (OTRANNUM) in the DFHCICS performance data group.

MNI_ID_ORIGIN_TRANID (TYPE-C, 4 BYTES)
The transaction ID (TRANSID) of the originating task; for example, the CWXN task. For more information, see field 363 (OTRAN) in the DFHCICS performance data group.

MNI_ID_ORIGIN_USERID (TYPE-C, 8 BYTES)
The originating Userid-2 or Userid-1, for example, from CWBA, depending on the originating task. For more information, see field 364 (OUSERID) in the DFHCICS performance data group.

MNI_ID_ORIGIN_USER_CORR (TYPE-C, 64 BYTES)
The originating user correlator. For more information, see field 365 (OUSERCOR) in the DFHCICS performance data group.

MNI_ID_ORIGIN_TCPIPSERV (TYPE-C, 8 BYTES)
The name of the originating TCPIPSERVICE. For more information, see field 366 (OTCPSVCE) in the DFHCICS performance data group.

MNI_ID_ORIGIN_PORTNUM (TYPE-A, 4 BYTES)
The port number used by the originating TCPIPSERVICE. For more information, see field 367 (OPORTNUM) in the DFHCICS performance data group.

MNI_ID_ORIGIN_CLIPADDR (TYPE-C, 40 BYTES)
The IP address of the originating client or Telnet client. For more information, see field 372 (OCLIPADR) in the DFHCICS performance data group.

MNI_ID_ORIGIN_CLIPPORT (TYPE-A, 4 BYTES)
The TCP/IP port number of the originating client or Telnet client. For more information, see field 369 (OCLIPORT) in the DFHCICS performance data group.

MNI_ID_ORIGIN_TRANFLAG (TYPE-A, 8 BYTES)
The originating transaction flags. This 64-bit string is used for signaling transaction definition and status information. For more information, see field 370 (OTRANFLG) in the DFHCICS performance data group.

MNI_ID_ORIGIN_FCTYNAME (TYPE-C, 8 BYTES)
The facility name of the originating transaction. If the originating transaction is not associated with a facility, this field is null. For more information, see field 371 (OFCTYNME) in the DFHCICS performance data group.

MNI_PHD_NETWKID (TYPE-C, 8 BYTES)
The network identifier of the CICS system of an immediately previous task in another CICS system with which this task is associated. For more information, see field 373 (PHNTWKID) in the DFHCICS performance data group.

MNI_PHD_APPLID (TYPE-C, 8 BYTES)
The APPLID from previous hop data. This is the APPLID of the CICS system of a previous task in another CICS system with which this task is associated. For more information, see field 374 (PHAPPLID) in the DFHCICS performance data group.

MNI_PHD_ATTACH_TIME (TYPE-T, 8 BYTES)
The start time of the immediately previous task in another CICS system with which this task is associated. For more information, see field 375 (PHSTART) in the DFHCICS performance data group.

MNI_PHD_TRANNO (TYPE-P, 4 BYTES)
The task number of the immediately previous task in another CICS system with which this task is associated. For more information, see field 376 (PHTRANNO) in the DFHCICS performance data group.

MNI_PHD_TRANID (TYPE-C, 4 BYTES
The transaction ID (TRANSID) of the immediately previous task in another CICS system with which this task is associated. For more information, see field 377 (PHTRAN) in the DFHCICS performance data group.

MNI_PHD_COUNT (TYPE-A, 4 BYTES)
The number of times there has been a request from one CICS system to another CICS system to initiate a task with which this task is associated. For more information, see field 378 (PHCOUNT) in the DFHCICS performance data group.

MNI_PTD_ATTACH_TIME (TYPE-T, 8 BYTES)
The start time of the immediately previous or parent task in the same CICS system with which this task is associated. For more information, see field 480 (PTSTART) in the DFHCICS performance data group.

MNI_PTD_TRANNUM (TYPE-P, 4 BYTES)
The task number of the immediately previous or parent task in the same CICS system with which this task is associated. For more information, see field 481 (PTTRANNO) in the DFHCICS performance data group.

MNI_PTD_TRANID (TYPE-C, 4 BYTES)
The transaction ID (TRANSID) of the immediately previous or parent task in the same CICS system with which this task is associated. For more information, see field 482 (PTTRAN) in the DFHCICS performance data group.

MNI_PTD_COUNT (TYPE-A, 4 BYTES)
The number of times there has been a request from one task to initiate another task in the same CICS system with which this task is associated, such as by a **RUN TRANSID** or **START** command. For more information, see field 483 (PTCOUNT) in the DFHCICS performance data group.

Data entry fields

Each identity record consists of an identity record header, an identity record identification section, and two identity data entries (an entry for the distinguished name and an entry for the realm). Each identity data entry consists of an entry identifier field, an entry length field, and a variable length entry field.

MNI_ENTRY_IDENT
Data entry identifier.

MNI_ENTRY_LENGTH
Length of the data entry that is specified by the data entry identifier.

MNI_ENTRY_FIELD
Data entry field.

Table 8. Identity record data entry fields

Data entry identifier decimal (hexadecimal)	Data entry length	Format	Description
1 (1)	1 - 246	UTF-8	A distinguished name, which uniquely identifies the user.
2 (2)	1 - 255	UTF-8	A realm, which identifies the set of resources to which the authentication information requested (that is, the user ID and password) applies.

Chapter 6. Generic alert and resolution structures

This appendix describes the structure of SNA generic alerts and resolutions as they are used by CICSPlex SM.

The generic alert structure

The CICSPlex SM Alert MS major vector contains generic alert information in a number of subvectors.

The CICSPlex SM Alert MS major vector contains the following subvectors.

"Generic Alert Data" (X'92') MS subvector
This identifies the Alert Description code as "IMPENDING PROBLEM: THRESHOLD HAS BEEN REACHED" (X'4012').

"Probable Causes" (X'93') MS subvector
This identifies a single code point specifying "PERFORMANCE DEGRADED" (X'4000').

"Cause Undetermined" (X'97') MS subvector
This contains:

- A "Recommended Actions" (X'81') common subfield. This identifies one code point specifying "REVIEW" (X'00A1').
- Two "Detailed Data" (X'82') common subfields containing:

  1. Data ID of "THRESHOLD PARAMETER" (X'7111'), with EBCDIC encoding. The contents are dependant on the ALERTVER CMAS System Parameter (EYUPARM) used by the CMAS identified in the ACTNDEF associated with the event.

     ALERTVER(0) and ALERTVER(1) records contain the following characters:

     0-2
     Creator (SAM] MRM] APM)

     3-5
     RTA Event Severity (VLS | VLW | LW | HW | HS | VHS)

     6-13
     RTA Event name (RTADEF name |STATDEF name | !!SAM*xxx*)

  2. Data ID of "PROBLEM DATA" (X'F511'), with EBCDIC encoding, containing the following characters:

     0-29
     The text of the "Enter Msg" from the action definition (ACTNDEF)

     ALERTVER(1) records also contain the following:

     14-16
     Alert Version (001)

     17-19
     Priority (001-255)

     20-27
     Sequence

     28-29
     Evaluation Logical Operator (GT | GE |EQ |NE |LE |LT)

     – Data is only available for RTADEF events. The field contains spaces for STATDEF and SAM events.

     30-32
     Evaluation Type (VAL | THR) - VALuation or THReshold.

– Data is only available for RTADEF events. The field contains spaces for STATDEF and SAM events.

33-44
Evaluation Resource Table Attribute

– Data is only available for RTADEF events. The field contains spaces for STATDEF and SAM events.

45-88
Evaluation Data

– Data is only available for RTADEF events which have an Evaluation Type of VAL. The field contains spaces for all other events.

89-132
Evaluation Last Evaluated Data

– Data is only available for RTADEF events. The field contains spaces for STATDEF and SAM events.

"Product Set ID" (X'10') MS common subvector
a "Product ID" (X'11') common subvector that identifies the product as IBM Software (X'04') and contains:

- A "Product Number" (X'08') Product ID subfield that identifies the product number as 5655S97.
- A "Product Common Name" (X'06') Product ID subfield that identifies the common name as CICSPLEX.SM.
- A "Product Common Level" (X'04') Product ID subfield that identifies the version, release, and modification levels.

"Hierarchy/Resource List" (X'05') MS common subvector
This contains:

- A "Hierarchy Name List" (X'10') Hierarchy/Resource List subfield, which contains the following list elements:

Element	Resource Type	Resource Name
1	Service point (X'81')	RTA CONTEXT
2	Unspecified device (X'00')	RTA SCOPE
3	Unspecified device (X'00')	RTA RESOURCE TYPE or SAM EVENT NAME

- An "Associated Resource List" (X'11') subfield, which contains a 16-character EBCDIC resource name that identifies either the event name or the instance key from the CICSPlex SM resource table associated with the event. For SAM events the "Associated Resource List" is set as follows:

  !!SAMMAX, !!SAMOPS or !!SAMNRM
  blank

  !!SAMSDM
  The 6-character system dump code

  !!SAMSOS
  The character DSA name or SOS location

  !!SAMSTL
  The character stall reason

  !!SAMTDM
  CCCCTTTTUUUUUUUU, where CCCC is the 4-character transaction dump code, TTTT is the 4-character transaction dump name and UUUUUUUU is the User Id associated with the dumping task.

"Incident Identification" (X'4A') MS common subvector

This contains an "Incident Identification" (X'01') Incident Identification subfield. This uses encoding type X'01'. The fields are as follows:

Field
Contents; Length

Netid:
Periods; 8 characters

Network addressable unit:
APPLID of the originating CMAS; 8 characters

Application name:
CICSplex name; 8 characters

Unique id:
CMAS name concatenated with GMT timestamp; 16 characters

The resolution structure

The Resolution (X'0002') MS major vector has the same structure as the Alert MS major vector, except for two "Detailed Data" common subfields of the "Cause Undetermined" MS subvector:

- the first "Detailed Data" common subfield of the "Cause Undetermined" MS subvector always contains spaces in the "Evaluation Last Evaluated Data" (89-132) field of the ALERTVER(1) version of the record.
- the second of the two "Detailed Data" common subfields of the "Cause Undetermined" MS subvector contains the text of the "Exit Msg" from the action definition rather than the "Enter Msg" text.

Notices

This information was developed for products and services offered in the United States of America. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 rights 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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (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

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may 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 want 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119 Armonk,
NY 10504-1785
United States of America

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 Client Relationship Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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 actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Programming interface information

IBM CICS supplies some documentation that can be considered to be Programming Interfaces, and some documentation that cannot be considered to be a Programming Interface.

Programming Interfaces that allow the customer to write programs to obtain the services of CICS Transaction Server for z/OS, Version 5 Release 6 (CICS TS 5.6) are included in the following sections of the online product documentation:

- Developing applications
- Developing system programs
- CICS TS security
- Developing for external interfaces
- Application development reference
- Reference: system programming
- Reference: connectivity

Information that is NOT intended to be used as a Programming Interface of CICS TS 5.6, but that might be misconstrued as Programming Interfaces, is included in the following sections of the online product documentation:

- Troubleshooting and support
- CICS TS diagnostics reference

If you access the CICS documentation in manuals in PDF format, Programming Interfaces that allow the customer to write programs to obtain the services of CICS TS 5.6 are included in the following manuals:

- Application Programming Guide and Application Programming Reference
- Business Transaction Services

- Customization Guide
- C++ OO Class Libraries
- Debugging Tools Interfaces Reference
- Distributed Transaction Programming Guide
- External Interfaces Guide
- Front End Programming Interface Guide
- IMS Database Control Guide
- Installation Guide
- Security Guide
- CICS Transactions
- CICSPlex System Manager (CICSPlex SM) Managing Workloads
- CICSPlex SM Managing Resource Usage
- CICSPlex SM Application Programming Guide and Application Programming Reference
- Java Applications in CICS

If you access the CICS documentation in manuals in PDF format, information that is NOT intended to be used as a Programming Interface of CICS TS 5.6, but that might be misconstrued as Programming Interfaces, is included in the following manuals:

- Data Areas
- Diagnosis Reference
- Problem Determination Guide
- CICSPlex SM Problem Determination Guide

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Apache, Apache Axis2, Apache Maven, Apache Ivy, the Apache Software Foundation (ASF) logo, and the ASF feather logo are trademarks of Apache Software Foundation.

Gradle and the Gradlephant logo are registered trademark of Gradle, Inc. and its subsidiaries in the United States and/or other countries.

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.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Red Hat®, and Hibernate® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Spring Boot is a trademark of Pivotal Software, Inc. in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Zowe™, the Zowe logo and the Open Mainframe Project™ are trademarks of The Linux Foundation.

The Stack Exchange name and logos are trademarks of Stack Exchange Inc.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability
These terms and conditions are in addition to any terms of use for the IBM website.

Personal use
You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use
You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights
Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM online privacy statement

IBM Software products, including software as a service solutions, (*Software Offerings*) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information (PII) is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect PII. If this Software Offering uses cookies to collect PII, specific information about this offering's use of cookies is set forth below:

For the CICSPlex SM Web User Interface (main interface):
Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name and other PII for purposes of session management, authentication, enhanced user usability, or other usage tracking or functional purposes. These cookies cannot be disabled.

For the CICSPlex SM Web User Interface (data interface):
Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's user name and other PII for purposes of session management, authentication, or other usage tracking or functional purposes. These cookies cannot be disabled.

For the CICSPlex SM Web User Interface ("hello world" page):
Depending upon the configurations deployed, this Software Offering may use session cookies that do not collect PII. These cookies cannot be disabled.

For CICS Explorer®:

Depending upon the configurations deployed, this Software Offering may use session and persistent preferences that collect each user's user name and password, for purposes of session management, authentication, and single sign-on configuration. These preferences cannot be disabled, although storing a user's password on disk in encrypted form can only be enabled by the user's explicit action to check a check box during sign-on.

If the configurations deployed for this Software Offering provide you, as customer, the ability to collect PII from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM Privacy Policy and IBM Online Privacy Statement, the section entitled *Cookies, Web Beacons and Other Technologies* and the IBM Software Products and Software-as-a-Service Privacy Statement.

Index