

Distributed Activity - Server

This topic shows detailed information about “Accounting - Distributed Activity - Server”.

This block shows the information provided for the server of the distributed activity.

In the following example both layouts are shown, the report layout followed by the trace layout.

Accounting - Distributed Activity - Server

The field labels shown in the following sample layout of “Accounting - Distributed Activity - Server” are described in the following section.

Report:

```

---- DISTRIBUTED ACTIVITY -----
SERVER          : CONVERSATIONS INITIATED:      #COMMT(1)SENT:      MESSAGES SENT      :
PRODUCT ID     : #CONVERSATIONS QUEUED  :      #ROLLB(1)SENT:      MESSAGES RECEIVED:
PRODUCT VERSION : CONVERSATION TERMINATED:    SQL SENT           : BYTES SENT         :
METHOD         : #RLUP THREADS         :      ROWS RECEIVED:    BYTES RECEIVED    :
REQUESTER ELAP.TIME:
SERVER ELAPSED TIME:
SERVER CPU TIME  :
DBAT WAITING TIME :
#DDF ACCESSES   :

#COMMIT(2) SENT :      #BACKOUT(2) SENT      :      #BKOUT(2) R.R:      #LASTAGN.SENT      :
SUCCESSFULLY ALLOC.CONV:    TRANSACT.SENT:      STMT BOUND AT SER:
MAX OPEN CONVERSATIONS :    MSG.IN BUFFER:      #FORGET RECEIVED :
#CONT->LIM.BL.FTCH SWCH:    #PREPARE SENT:
#COMMIT(2) RESP.RECV. :
    
```

Trace:

```

---- DISTRIBUTED ACTIVITY -----
SERVER          : CONVERSATION TERMINATED:    NBR RLUP THREADS :
PRODUCT ID     : COMMIT(1)SENT           : MESSAGES SENT     :
PRODUCT VERSION : ROLLB(1)SENT           : MESSAGES RECEIVED:
METHOD         : SQL SENT                : BYTES SENT        :
REQUESTER ELAP.TIME:
SERVER ELAPSED TIME:
SERVER CPU TIME  :
DBAT WAITING TIME :
CONVERSATIONS INITIATED:
CONVERSATIONS QUEUED :

COMMIT(2) SENT  :      SUCCESSFULLY ALLOC.CONV:    MSG.IN BUFFER      :
BACKOUT(2) SENT :    MAX OPEN CONVERSATIONS :    PREPARE SENT       :
                  CONT->LIM.BL.FTCH SWCH:    LAST AGN.SENT     :
                  COMMIT(2) RESP.RECEIVED:    STMT BOUND AT SER:
                  BKOUT(2) R.R           :    FORGET RECEIVED  :
                  TRANSACT.SENT          :
    
```

REPORT - SERVER

The name of the remote location with which this information is associated. If the local location is the requester, this field is a server location. If the local location is a server location, this field is the requester location. An allied thread is created at a DB2 requester, and a database access thread is created at a DB2 server. An accounting record is for either a requester or a server, but not for both.

This field is invalid if summary rollup data is present. In Accounting this field is set to *ROLSUM*.

Field Name: QLACLOCN

This is an *exception* field.

REPORT - PRODUCT ID

Distributed Activity - Server

The original DB2 field specifies the information in the following field names of the remote requester or server location:

PRODUCT ID

It consists of 3 characters and can have the following values:

Original ID from DB2	Shown as
DSN	DB2
ARI	SQL/DS
QSQ	DB2/400
SQL	COMMON SERV
JCC	JDBC DRIVER
N/P	
Other	Original ID from DB2

Note:

- If the record was written at the application requester location, or if summary rollup data is available, N/P is shown in Accounting TRACE and REPORT.
- In Accounting FILE and SAVE DDF tables, BLANK is shown.

PRODUCT VERSION (PROD VERSION)

It consists of 5 digits and is shown as *VvvRrrMm*, where:

vv Version level
rr Release level
m Modification level

Note: For DDF/RRSAF rollup records, the product ID and product version contain a value derived from the last thread to rollup. For query parallelism rollup threads, the value is being derived from the parent record.

Field Name: QLACPRID

REPORT - PRODUCT VERSION

The original DB2 field specifies the information in the following field names of the remote requester or server location:

PRODUCT ID

It consists of 3 characters and can have the following values:

Original ID from DB2	Shown as
DSN	DB2
ARI	SQL/DS
QSQ	DB2/400
SQL	COMMON SERV
JCC	JDBC DRIVER
N/P	
Other	Original ID from DB2

Note:

- If the record was written at the application requester location, or if summary rollup data is available, N/P is shown in Accounting TRACE and REPORT.
- In Accounting FILE and SAVE DDF tables, BLANK is shown.

PRODUCT VERSION (PROD VERSION)

It consists of 5 digits and is shown as *VvvRrrMm* , where:

<i>vv</i>	Version level
<i>rr</i>	Release level
<i>m</i>	Modification level

Note: For DDF/RRSAF rollup records, the product ID and product version contain a value derived from the last thread to rollup. For query parallelism rollup threads, the value is being derived from the parent record.

Field Name: QLACPRID

Report - METHOD

The method of access: DB2 private protocol, DRDA protocol, or both.

This field is invalid if unique or summary rollup data is present. It can have the following value in:

- Accounting Trace and Report: N/P
- The Accounting FILE and SAVE PROGRAM table: blank

Field Name: ADPROTOC

Report - REQUESTER ELAP.TIME

The elapsed time at the requester. It includes the total of DB2 and network time.

Field Name: ADDSELRQ

Report - SERVER ELAPSED TIME

The elapsed database access agent time at the server location. This value is updated at the requester location.

Special Considerations:

- This value is reported only for DB2 private protocol. If only DRDA protocol, N/C is shown.
- If both DB2 private protocol and DRDA protocol are used, then only the elapsed time associated with the DB2 private protocol is reported, and this can be misleading.
- This value is calculated by accumulating the difference between the store clock values obtained after receiving a request message and before sending the associated reply message.
- When block fetch is used, this time can be longer than the time for ADDSELRQ (ELAPSED REQ).
- Compare this value with the accounting class 2 time (allied agent time in DB2) to see if the distributed-allied thread using the database access agent spends too much time in remote processing.

Field Name: ADDSELSR

Distributed Activity - Server

This is an *exception* field.

Report - SERVER CPU TIME

The database access agent CPU time spent at the server location. This value is updated at the requester location, is intended for problem determination only, and should not be used for charge back.

Special Considerations:

1. This value is reported only for DB2 private protocol. If only DRDA protocol is used, *N/C* is shown.
2. If both DB2 private protocol and DRDA protocol are used, then only the CPU time associated with the DB2 private protocol is reported, and this can be misleading.
3. This value is calculated by accumulating the amount of CPU time spent by the database access thread at the DB2 server location each time a request message is processed.
4. Certain programming techniques can cause this value to not be received at the requester location (and therefore not included in this field), even though the CPU time was spent at the server location and was properly measured and sent to the requester location.

Field Name: ADDSSRSR

This is an *exception* field.

Report - DBAT WAITING TIME

Total elapsed time spent waiting for an available database access agent slot.

Field Name: QLACMDWT

Report - #DDF ACCESSES

The number of occurrences of the remote location and method pair.

Field Name: ASDDF

Report - #COMMIT(2) SENT

The number of commit requests sent to the participant (two-phase commit operations only). This value is maintained at the participant, indicating that the participant was read only.

Field Name: QLACCRSE

Report - CONVERSATIONS INITIATED

The number of conversations (both successful and unsuccessful) initiated by the requester location to be executed at the server location. This number is maintained at the requester.

Field Name: QLACCNVS

Report - #CONVERSATIONS QUEUED

A number of conversation requests queued by DDF that are waiting for allocation. This value is maintained at the requester location.

If the value is a large number, you might want to increase the limit for the number of conversations.

Field Name: QLACCNVQ

This is an *exception* field.

Report - CONVERSATION TERMINATED

The number of terminated conversations in the server block (DB2 private protocol only). It is maintained at the requester location.

This number can be different from the number of successful conversation allocations, because some conversations might not have been terminated when the accounting record was written.

Field Name: QLACCNVT

This is an *exception* field.

Report - #RLUP THREADS

The number of threads to roll data into this QLAC data section. Non-rollup QLACs have a value of 1 and rollup QLACs have a value of 1 or more.

Field Name: QLACRLNU

Report - #BACKOUT(2) SENT

The number of backout requests sent to the participant (two-phase commit operations only).

Field Name: QLACBKSE

Report - #COMMT(1)SENT

The number of commit requests sent to the server (single-phase commit protocol) and committed requests sent to the participant (two-phase commit protocol).

Field Name: QLACCOMS

This is an *exception* field.

Report - #ROLLB(1)SENT

The number of abort requests sent to the server (single-phase commit protocol) and backout requests sent to the participant (two-phase commit protocol).

Field Name: QLACABRS

This is an *exception* field.

Report - SQL SENT

The number of SQL statements sent to the server location. This value is maintained at the requesting location.

Field Name: QLACSQLS

This is an *exception* field.

Report - ROWS RECEIVED

The number of rows of data retrieved from the server location. This value is maintained at the requester location.

Special Considerations:

1. The number of rows received from the server location does not include either the SQLDA or SQLCA.
2. Block fetch can significantly affect the number of rows sent across the network. When used with non-UPDATE cursors, block fetch puts as many rows as possible into the message buffer, and transmits the buffer across the network without requiring a VTAM message. Consequently,

Distributed Activity - Server

more rows of data might be sent from the server location than are received by the reporting (requester) location. This is especially true when DB2 private protocol is used because multiple blocks can be transmitted from the server with no intervening messages sent by the requester.

Field Name: QLACROWR

This is an *exception* field.

Report - MESSAGES SENT

The number of messages sent to the location. It is maintained at the location where the messages originated.

Field Name: QLACMSGS

This is an *exception* field.

Report - MESSAGES RECEIVED

The number of messages received from the location. This value is maintained at the location where the messages were received.

More messages might be sent from the server location than are received by the requester because of the way in which distributed SQL statements are processed internally.

Field Name: QLACMSGR

This is an *exception* field.

Report - BYTES SENT

The number of bytes the server location sent to the requester location. This value is maintained at the server location.

More bytes of data might be sent from the server location than are received by the requester due to the way in which distributed SQL statements are processed internally.

Field Name: QLACBYTS

This is an *exception* field.

Report - BYTES RECEIVED

The number of bytes the server location received from the requester location.

More bytes of data might be sent from the server location than are received by the requester, because of the way in which distributed SQL statements are processed internally.

Field Name: QLACBYTR

This is an *exception* field.

Report - BLOCKS RECEIVED

The number of blocks received using block fetch. This value is maintained at the requester location.

Field Name: QLACBRBF

This is an *exception* field.

Trace - SERVER

Distributed Activity - Server

The name of the remote location with which this information is associated. If the local location is the requester, this field is a server location. If the local location is a server location, this field is the requester location. An allied thread is created at a DB2 requester, and a database access thread is created at a DB2 server. An accounting record is for either a requester or a server, but not for both.

This field is invalid if summary rollup data is present. In Accounting this field is set to *ROLSUM*.

Field Name: QLACLOCN

This is an *exception* field.

TRACE - PRODUCT ID

The original DB2 field specifies the information in the following field names of the remote requester or server location:

PRODUCT ID

It consists of 3 characters and can have the following values:

Original ID from DB2	Shown as
DSN	DB2
ARI	SQL/DS
QSQ	DB2/400
SQL	COMMON SERV
JCC	JDBC DRIVER
N/P	
Other	Original ID from DB2

Note:

- If the record was written at the application requester location, or if summary rollup data is available, N/P is shown in Accounting TRACE and REPORT.
- In Accounting FILE and SAVE DDF tables, BLANK is shown.

PRODUCT VERSION (PROD VERSION)

It consists of 5 digits and is shown as *VvvRrrMm*, where:

vv Version level
rr Release level
m Modification level

Note: For DDF/RRSAF rollup records, the product ID and product version contain a value derived from the last thread to rollup. For query parallelism rollup threads, the value is being derived from the parent record.

Field Name: QLACPRID

TRACE - PRODUCT VERSION

The original DB2 field specifies the information in the following field names of the remote requester or server location:

Distributed Activity - Server

PRODUCT ID

It consists of 3 characters and can have the following values:

Original ID from DB2	Shown as
DSN	DB2
ARI	SQL/DS
QSQ	DB2/400
SQL	COMMON SERV
JCC	JDBC DRIVER
N/P	
Other	Original ID from DB2

Note:

- If the record was written at the application requester location, or if summary rollup data is available, N/P is shown in Accounting TRACE and REPORT.
- In Accounting FILE and SAVE DDF tables, BLANK is shown.

PRODUCT VERSION (PROD VERSION)

It consists of 5 digits and is shown as *VvvRrrMm* , where:

<i>vv</i>	Version level
<i>rr</i>	Release level
<i>m</i>	Modification level

Note: For DDF/RRSAF rollup records, the product ID and product version contain a value derived from the last thread to rollup. For query parallelism rollup threads, the value is being derived from the parent record.

Field Name: QLACPRID

Trace - METHOD

The method of access: DB2 private protocol, DRDA protocol, or both.

This field is invalid if unique or summary rollup data is present. It can have the following value in:

- Accounting Trace and Report: N/P
- The Accounting FILE and SAVE PROGRAM table: blank

Field Name: ADPROTOC

Trace - REQUESTER ELAP.TIME

The elapsed time at the requester. It includes the total of DB2 and network time.

Field Name: ADDSELRQ

Trace - SERVER ELAPSED TIME

The elapsed database access agent time at the server location. This value is updated at the requester location.

Special Considerations:

- This value is reported only for DB2 private protocol. If only DRDA protocol, *N/C* is shown.
- If both DB2 private protocol and DRDA protocol are used, then only the elapsed time associated with the DB2 private protocol is reported, and this can be misleading.
- This value is calculated by accumulating the difference between the store clock values obtained after receiving a request message and before sending the associated reply message.
- When block fetch is used, this time can be longer than the time for ADDSELRQ (ELAPSED REQ).
- Compare this value with the accounting class 2 time (allied agent time in DB2) to see if the distributed-allied thread using the database access agent spends too much time in remote processing.

Field Name: ADDSELSR

This is an *exception* field.

Trace - SERVER CPU TIME

The database access agent CPU time spent at the server location. This value is updated at the requester location, is intended for problem determination only, and should not be used for charge back.

Special Considerations:

1. This value is reported only for DB2 private protocol. If only DRDA protocol is used, *N/C* is shown.
2. If both DB2 private protocol and DRDA protocol are used, then only the CPU time associated with the DB2 private protocol is reported, and this can be misleading.
3. This value is calculated by accumulating the amount of CPU time spent by the database access thread at the DB2 server location each time a request message is processed.
4. Certain programming techniques can cause this value to not be received at the requester location (and therefore not included in this field), even though the CPU time was spent at the server location and was properly measured and sent to the requester location.

Field Name: ADDSSRSR

This is an *exception* field.

Trace - DBAT WAITING TIME

Total elapsed time spent waiting for an available database access agent slot.

Field Name: QLACMDWT

Trace - CONVERSATIONS INITIATED

The number of conversations (both successful and unsuccessful) initiated by the requester location to be executed at the server location. This number is maintained at the requester.

Field Name: QLACCNVS

Trace - CONVERSATIONS QUEUED

A number of conversation requests queued by DDF that are waiting for allocation. This value is maintained at the requester location.

Distributed Activity - Server

If the value is a large number, you might want to increase the limit for the number of conversations.

Field Name: QLACCNVQ

This is an *exception* field.

Trace - #COMMIT(2) SENT

The number of commit requests sent to the participant (two-phase commit operations only). This value is maintained at the participant, indicating that the participant was read only.

Field Name: QLACCRSE

This is an *exception* field.

Trace - #BACKOUT(2) SENT

The number of backout requests sent to the participant (two-phase commit operations only).

Field Name: QLACBKSE

This is an *exception* field.

Trace - CONVERSATION TERMINATED

The number of terminated conversations in the server block (DB2 private protocol only). It is maintained at the requester location.

This number can be different from the number of successful conversation allocations, because some conversations might not have been terminated when the accounting record was written.

Field Name: QLACCNVT

This is an *exception* field.

Trace - COMMT(1)SENT

The number of commit requests sent to the server (single-phase commit protocol) and committed requests sent to the participant (two-phase commit protocol).

Field Name: QLACCOMS

This is an *exception* field.

Trace - ROLLB(1)SENT

The number of abort requests sent to the server (single-phase commit protocol) and backout requests sent to the participant (two-phase commit protocol).

Field Name: QLACABRS

This is an *exception* field.

Trace - SQL SENT

The number of SQL statements sent to the server location. This value is maintained at the requesting location.

Field Name: QLACSQLS

This is an *exception* field.

Trace - ROWS RECEIVED

The number of rows of data retrieved from the server location. This value is maintained at the requester location.

Special Considerations:

1. The number of rows received from the server location does not include either the SQLDA or SQLCA.
2. Block fetch can significantly affect the number of rows sent across the network. When used with non-UPDATE cursors, block fetch puts as many rows as possible into the message buffer, and transmits the buffer across the network without requiring a VTAM message. Consequently, more rows of data might be sent from the server location than are received by the reporting (requester) location. This is especially true when DB2 private protocol is used because multiple blocks can be transmitted from the server with no intervening messages sent by the requester.

Field Name: QLACROWR

This is an *exception* field.

Trace - NBR RLUP THREADS

The number of threads to roll data into this QLAC data section. Non-rollup QLACs have a value of 1 and rollup QLACs have a value of 1 or more.

Field Name: QLACRLNU

Trace - MESSAGES SENT

The number of messages sent to the location. It is maintained at the location where the messages originated.

Field Name: QLACMSGS

Trace - MESSAGES RECEIVED

The number of messages received from the location. This value is maintained at the location where the messages were received.

More messages might be sent from the server location than are received by the requester because of the way in which distributed SQL statements are processed internally.

Field Name: QLACMSGR

Trace - BYTES SENT

The number of bytes the server location sent to the requester location. This value is maintained at the server location.

More bytes of data might be sent from the server location than are received by the requester due to the way in which distributed SQL statements are processed internally.

Field Name: QLACBYTS

Trace - BYTES RECEIVED

The number of bytes the server location received from the requester location.

More bytes of data might be sent from the server location than are received by the requester, because of the way in which distributed SQL statements are processed internally.

Field Name: QLACBYTR

Distributed Activity - Server

Trace - BLOCKS RECEIVED

The number of blocks received using block fetch. This value is maintained at the requester location.

Field Name: QLACBRBF