

Updates that apply to IBM® DB2® Analytics Accelerator Loader for z/OS® V2R1 User's Guide (SC27-6777-00)

**Date of change:** March 2017

**Topic:** Multiple

**Change description:** Documentation changes made in support of PTF UI45762 APAR PI77145 – Stored procedure support for high availability load

- Topic: “What’s new” in chapter “Overview”
- Topic: “Features and benefits” in chapter “Overview”
- New topic: “Using a stored procedure to perform a high availability load” in chapter “Loading DB2 table data to multiple accelerators”
- New topic: “Canceling an HALOAD stored procedure call” in chapter “Loading DB2 table data to multiple accelerators”

**Topic: “What’s new”**

Add the following description:

The Accelerator Loader high availability load feature can be invoked as a batch job or by using a stored procedure call.

**Topic: “Features and benefits”**

Add the following description to subtopic “High availability load”:

You can invoke the high availability load feature by running the HALOAD utility as a batch job or using a stored procedure call.

**New topic: “Using a stored procedure to perform a high availability load”**

Add this new topic to chapter “Loading DB2 table data to multiple accelerators.”

You can invoke the Accelerator Loader high availability load feature using a stored procedure call.

To invoke the HALOAD utility using a stored procedure, use the standard DB2 utility stored procedure DSNUTILU. Call the DSNUTILU stored procedure as you would for a DB2 utility, but specify the HALOAD utility command in place of a DB2 utility command. By using the DSNUTILU stored procedure to process the HALOAD utility, you do not need to create and maintain another stored procedure, and can simply use a stored procedure already implemented as part of DB2.

### Restrictions and considerations

The following restrictions and considerations apply when using a stored procedure to perform a high availability load:

- If loading multiple accelerators using the HALOAD utility, all the accelerators must be configured to the same DB2. You can also use the HALOAD utility to load a single accelerator.
- The Accelerator Loader product library must be included in the STEPLIB of the procedure that runs the WLM environment for DSNUTILU.
- The HALOAD command does not use a utility ID. Because of this, an executing HALOAD command cannot be canceled by the DB2 -TERM utility command using the *utility-id* value passed on the stored procedure call. For more information, see “Canceling an HALOAD stored procedure call.”

### Procedure

To perform a high availability load from within an application program, use the SQL CALL statement to invoke the DSNUTILU stored procedure, specifying the HALOAD command as the utility. For the HALOAD utility output, the calling program fetches rows from the SYSIBM.SYSPRINT temporary table, which is the same process as when calling a DB2 utility.

The following information is specific to using the DSNUTILU stored procedure to run the HALOAD utility.

**Note:** For more information on using the DSNUTILU stored procedure, including a sample program and a complete description of the DSNUTILU syntax options, see the *IBM DB2 Utility Guide and Reference*, in Appendix B “DB2-supplied stored procedures for utility operations” or “DSNUTILU stored procedure”.

**Syntax diagram:**

The following syntax diagram shows the SQL CALL statement for invoking the HALOAD utility as a stored procedure:

```
>>-CALL--DSNUTILU--(--utility-id, restart, utstmt, retcode--)-<<----
```

**Syntax definitions:*****utility-id***

This input parameter is ignored for HALOAD. Because HALOAD does not invoke a DB2 utility, a utility ID is not used. Although this parameter is not used with HALOAD, a valid value must be provided for DSNUTILU.

***restart***

This input parameter is ignored for HALOAD. The HALOAD utility cannot be restarted. Although this parameter is not used with HALOAD, a valid value must be provided for DSNUTILU. It is recommended to specify NO for this option.

***utstmt***

Specifies the HALOAD utility control statement, such as shown in the following example:

```
HALOAD ACCEL (IDAA01, IDAA02) FROM TABLE SCHEMA.TBNAME
```

*utstmt* is a required input parameter.

For more information on the syntax for loading multiple accelerators, see “Loading data to multiple accelerators.”

***retcode***

Specifies the HALOAD utility highest return code. *retcode* is a required output parameter.

**Example**

The following example shows the SQL CALL statement for invoking the HALOAD utility as a stored procedure:

```
CALL SYSPROC.DSNUTILU('HALOADID',
                      'NO',
                      'HALOAD ACCEL (IDAA01, IDAA02) FROM TABLE SCHEMA.TBNAME',
                      HALOAD_RC)
```

The following example shows the output when performing a high availability load through the DSNUTILU stored procedure. The output is the same when invoking HALOAD directly or as a stored procedure.

```
HLOU4005I 017 08:31:53.39 High Availability Load Utility execution started.
HLOU4004I 017 08:32:00.90 Task: 01, Load completed for table: HLO15.HALD4BT1, partition: 0
HLOU4004I 017 08:32:09.40 Task: 02, Load completed for table: HLO15.HALD4BT2, partition: 0
HLOU4015I 017 08:32:11.91 Messages from accelerator V81AACC1...
HLOU5720I AQT20014I The following data was transferred to the "HLO15"."HALD4BT1" table:
Scope: Full table, number
HLOU5720I of rows: 8, amount of data: 0 MB, time: 10 seconds.
HLOU5720I AQT20014I The following data was transferred to the "HLO15"."HALD4BT2" table:
Scope: Full table, number
HLOU5720I of rows: 4, amount of data: 0 MB, time: 5 seconds.
HLOU5720I AQT10000I The operation was completed successfully.
HLOU4015I 017 08:32:11.91 Messages from accelerator IDAAS03 ...
HLOU5720I AQT20014I The following data was transferred to the "HLO15"."HALD4BT1" table:
Scope: Full table, number
HLOU5720I of rows: 8, amount of data: 0 MB, time: 10 seconds.
HLOU5720I AQT20014I The following data was transferred to the "HLO15"."HALD4BT2" table:
Scope: Full table, number
HLOU5720I of rows: 4, amount of data: 0 MB, time: 5 seconds.
HLOU5720I AQT10000I The operation was completed successfully.
HLOU4015I 017 08:32:11.91 Messages from accelerator IDAAS02 ...
HLOU5720I AQT20014I The following data was transferred to the "HLO15"."HALD4BT1" table:
Scope: Full table, number
HLOU5720I of rows: 8, amount of data: 0 MB, time: 10 seconds.
HLOU5720I AQT20014I The following data was transferred to the "HLO15"."HALD4BT2" table:
Scope: Full table, number
HLOU5720I of rows: 4, amount of data: 0 MB, time: 5 seconds.
HLOU5720I AQT10000I The operation was completed successfully.
```

**New topic: “Canceling an HALOAD stored procedure call”**

Add this new topic to chapter “Loading DB2 table data to multiple accelerators.”

The HALOAD command does not use a utility ID. Because of this, an executing HALOAD utility cannot be canceled by the DB2 -TERM utility command using the *utility-id* value passed on the stored procedure call. To cancel the HALOAD stored procedure call, use one of the following methods:

- If the HALOAD utility is running, you can cancel the thread through which the HALOAD utility attempts to access DB2. This will result in an S04E ABEND and the stored procedure will terminate.
- If the HALOAD utility hangs in the WLM address space, canceling the thread will not terminate the stored procedure. If this occurs, you can cancel the WLM address space that is running the DSNUTILU stored procedure call for the HALOAD utility. The WLM address space ID (ASID) can be identified from the Accelerator Loader started task message HLOS0101I. This message reports the intercept session information, including the ASID as *session\_asid*. For more information, locate the message ID in “Messages and codes.”
- In the Accelerator Loader studio, you can cancel the ACCEL\_LOAD\_TABLES stored procedure.