What's new in Optim High Performance Unload

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.
Optim™ High Performance Unload V6.1 GA includes the following new features:

Support of Db2® V11.1

Optim High Performance Unload can support Db2 V11.1 databases.

Support of Db2 native encryption based on a local keystore file

It is now possible to perform tasks retrieving data from databases or backups encrypted with Db2 native encryption. This ability is limited to non-compressed encrypted backups. Environments which encryption key is stored into a local keystore file are supported.

Interface with Big Data environments

Optim High Performance Unload can be used as an interface with Big Data destinations, either through a standard unload from a Db2 environment and a subsequent ingestion into the destination considered, or through an automatic migration performing these two tasks in one step.

New JSON output format

Optim High Performance Unload supports a new JSON output format.

Storing user-defined credentials for Db2 keystore file access

There is the ability to store user-defined Optim High Performance Unload credentials in order to run tasks retrieving data from an encrypted database or backup, for which accessing to a Db2 keystore file is necessary.

Native support of Db2 variables

Optim High Performance Unload tasks based on an SQL statement containing the specification of a Db2 variable can now be processed in native mode.

Replacement of the QUIESCE option by the FLUSH BUFFERPOOLS option

The QUIESCE option is now deprecated and replaced by the new FLUSH BUFFERPOOLS option.

Extension of the data-partitions selection capability

The ability exists to select and extract the content of a data-partitioned table via the name of the tablespace where the data-partitions are located.

Extension of the shared data-partitions processing

Sharing the processing of the data-partitions of a data-partitioned table can now be enabled to a task retrieving the data from a live database. Previously, this ability only existed for tasks getting the data from backups.

Reading a partition map from a file

When repartitioning data, it is now possible to use a file generated with the db2gpmap Db2 utility.

Setting a timeout for the opening of pipes used during an automatic migration

It might be necessary to increase the timeout applying to the opening of pipes used during an automatic migration, particularly if the latency is too long at the beginning of the loading side of the migration. Such a timeout can now be set in the configuration file.

Stack size setting

You can use the **`stacksize`** configuration file parameter to tune the maximum size for the stack related to an Optim High Performance Unload process.

Replacement value for null values

When using the DELIMITED output format, a specific value can now be specified in order to use it to replace null values into the output file generated.

Optim High Performance Unload V6.1 Fixpack 1 includes the following new features:

Support of Db2 native encryption based on a KMIP server

It is now possible to perform tasks retrieving data from databases or backups encrypted with Db2 native encryption. This ability is limited to non-compressed encrypted backups. Environments which encryption key is stored on a KMIP server are supported.

Support of strings having the CODEUNITS32 option set

The strings having the CODEUNITS32 option set are now supported, so that tables having columns with the CODEUNITS32 option set can be processed correctly.

Choosing the target partition map size

When performing a data migration, it might occur that the size of the partition map applying to the target database is different from the one applying to the source database. In this case, specifying explicitly the size of the partition map to be considered on the target database is necessary through the usage of the REPART option of the TARGET ENVIRONMENT clause.

Support of a remote database cataloged locally as a destination

One can now consider using a remote database cataloged locally as a destination, either for a data migration or the generation of a Db2 Load command, through an appropriate specification of the LOADDEST clause.

Interface with a Db2 Warehouse environment as a destination

One can now consider using a Db2 Warehouse environment as a destination, either for a data migration or the generation of an upload command, through an appropriate specification of the LOADDEST clause.

Ability to configure several destinations of the same type

The syntax of the LOADDEST clause and the usage of the configuration file for destinations have been improved in order to support the configuration of several destinations of the same type.

Enlarged support of Db2 Load modifiers

The list of Db2 Load modifiers supported and which can be specified into the FORMAT clause has been enlarged.

Optim High Performance Unload V6.1 Fixpack 2 includes the following new features:

Support of Db2 native encryption based on a PKCS#11 keystore on an HSM material

It is now possible to perform tasks retrieving data from databases or backups encrypted with Db2 native encryption. This ability is limited to non-compressed encrypted backups. Environments which encryption key is stored into a PKCS#11 keystore on an HSM material are supported.

Native support of the BOOLEAN Db2 data type

Optim High Performance Unload tasks based on an SQL statement referring to a BOOLEAN Db2 data type column can now be processed in native mode.

Native support of the BINARY and VARBINARY Db2 data types

Optim High Performance Unload tasks based on an SQL statement referring to a BINARY or VARBINARY Db2 data type column can now be processed in native mode.

Interface with a Db2 Warehouse environment as a data source

One can now consider using a Db2 Warehouse environment as a data source, for a data unload from it.

Interface with an Amazon S3 environment as a destination

One can now consider using an Amazon S3 environment as a destination, either for a data migration or the generation of an upload command, through an appropriate specification of the LOADDEST clause.

Interface with a Swift environment as a destination

One can now consider using a Swift environment as a destination, either for a data migration or the generation of an upload command, through an appropriate specification of the LOADDEST clause.

Ignoring data-partitions excluded by a WHERE clause

When processing a data-partitioned table with an SQL statement containing the specification of a WHERE clause excluding all the records within a given set of data-partitions, Optim High Performance Unload only reads the data-partitions corresponding to the WHERE clause scope.

Optim High Performance Unload V6.1 Fixpack 3 includes the following new features:

Support of both compressed and encrypted backups

It is now possible to perform tasks retrieving data from both compressed and encrypted backups.

Support of backups compressed with the NX842 hardware compression on AIX

It is now possible to perform tasks retrieving data from backups compressed with the NX842 hardware compression on AIX platforms.

Support of the ASC format for automatic migration

When performing a data migration towards a DPF environment, it might be necessary to keep the precision of REAL and DOUBLE data type columns. The ASC format specified with the BINARYNUMERICS modifier allows to keep these data types precision. The ASC format can now be specified for a data migration.

Ability of setting values to Db2 variables

It is now possible to set values to Db2 variables, in order for them to be applied to the SQL statement processed.

Disabling the padding on (VAR)GRAPHIC and DBCLOB data of odd size in ASC format

You can use the **`graphic_even_padding`** configuration file parameter to disable the padding to an even size of (VAR)GRAPHIC and DBCLOB data in ASC format.

Generating a CSV type header in DEL or DELIMITED format output files

It is now possible to generate a CSV type header in DEL or DELIMITED format output files, by specifying the existing HEADER option.

Ability of regularly synchronize files to disk

You can use the **`syncsize`** configuration file parameter to enable the ability of regularly synchronize output files to disk, in order to avoid a huge memory consumption by the disk caching.

Support of columns specification into the pattern of the LIKE operator

It is now possible to get LIKE expressions specified with a column into their pattern processed in native mode.

Parallel processing of columns of column-organized tables

You can now have the columns of a column-organized table read in parallel. The number of columns to be read in parallel can be chosen, either globally by setting the **`blu_parallelism`** configuration file parameter, or specifically for a given unload by specifying the BLU_PARALLELISM option into a control file.

Parent topic:

→ Optim High Performance Unload overview