

ibec_CompareTables

Compares the data of specified [tables](#) and creates a script of all discrepancies.

Description

This function compares the data of two tables and creates a discrepancy script. Both tables must have a [primary key](#).

It is possible to include milliseconds into [time/timestamp](#) values when comparing table data. Use the IncludeMilliseconds or IncludeMsecs option for this.

If the WHERE clause is specified it will be applied to all tables listed in the list of tables to compare.

Since IBExpert version 2020.09.13 a lot of additional statistical/progress information is passed to a callback block. See the IBEBlock page in the [Table Data Comparer](#) form for a full list of available data.

Syntax

```
function ibec_CompareTables(MasterDB : variant; SubscriberDB : variant;  
                           MasterTable : string; SubscriberTable :string;  
                           ScriptFile : string; Options : string;  
                           CallbackProc : variant) : variant;
```

Parameters

MasterDB	A handle to the reference database, maybe 0 or NULL if the current connection is used as a reference connection.
SubscriberDB	A handle to the comparative database, maybe 0 or NULL if the current connection is used as a comparative connection.
MasterTable, SubscriberTable	Names of the reference and comparative tables .
ScriptFile	Name of the script file which will contain the discrepancy script.
Options	List of options, delimited with a semicolon; possible options are:
NoConnect	If this option is specified in the option string the result script will not contain CONNECT statement and corresponding SET NAMES, SET SQL DIALECT and SET CLIENTLIB directives.
OmitDeletes	Missing records will not be checked by the data comparison. You can also use ProcessDeletes=0.
OmitInserts	New records will not be checked by the data comparison. You can also use ProcessInserts=0.
OmitUpdates	Modified records will not be checked by the data comparison. You can also use ProcessDeletes=0.
UpdateAllColumns	If this option is specified UPDATE statements will include non-modified columns too.

MasterDB	A handle to the reference database, maybe 0 or NULL if the current connection is used as a reference connection.
AppendMode	If this option is specified and the file ScriptFile already exists the resulting script will be appended to the ScriptFile. Otherwise a new file will be created.
UseHashFunc[tion]	This option allows the comparison of blobs using the HASH function. This option will be ignored if one of the servers - source or comparative - doesn't support the HASH function.
UseBlocks	This option forces the packing of INSERT/UPDATE/DELETE statements into Firebird blocks (EXECUTE BLOCK) if possible.
UseUniques	This option has been implemented to allow the comparison of tables which have a UNIQUE constraint instead of a PRIMARY KEY.
BlobsAsHex	If this option is enabled the values of blob fields will be written directly into the result script in hexadecimal notation, i.e. UPDATE MYTABLE SET BLOB_FIELD = x'4E6572766556E';. Hexadecimal notation for binary values is available in Firebird 2.5 and Firebird 3. The length of the binary value which may be encoded in hexadecimal notation is limited to 32765 bytes in Firebird 2.5 and to 65535 bytes in Firebird 3. If the blob value exceeds the limit it will be written in the lob-file as is.
IncludeOldValues	If this option is enabled, the old values of fields (for UPDATE statements) will be included into the result script as comments. See Table Data Comparer .
IgnoreSync	If the Ignore synchronization errors option is ON all INSERT/UPDATE/DELETE errors during online synchronization are ignored, allowing an attempt to execute all statements in the synchronization queue. See Table Data Comparer.
QueueLimit	A positive value means limitation by number of statements. A negative value means limitation by memory size (in megabytes). Default value is 0 - no limits.
Examples:	Examples:
QueueLimit=100 - number of statements in the queue must not exceed 100.	QueueLimit=100 - number of statements in the queue must not exceed 100.
QueueLimit=-5 - the queue must not occupy more than 5 megabytes.	QueueLimit=-5 - the queue must not occupy more than 5 megabytes.
Return	By default, this function returns null, but now it is possible to force it to return the total number of different records: specify Return=DiffCount in the option string.
CallbackProc	A callback IBEBlock which will be executed for each record processed whilst comparing data. The callback IBEBlock must have at least one input parameter, which will be used to pass a number of processed records within it.
where	The WHERE clause can be used to restrict the record sets which will be compared. Please refer to the examples below .
CommitBlock or CommitBlocks	This option performs a COMMIT after each Firebird block.
CommitAfter=<number>	This specifies the number of statements between COMMITs.
SyncOnline	This performs online synchronization. This allows you to update the target database simultaneously with the comparison process. Online synchronization is performed in a separate thread.

It is possible to compare more than one table in a single operation. Simply specify the list of necessary tables, delimited with a comma or semicolon, as MasterTable and SubscriberTable. For example:

```
ibec_CompareTables(DB1, DB2, 'TABLE1, TABLE2, "Table3"',
                  'TABLE1, TABLE2, "Table3"',
                  'D:\Diff.sql', 'UpdateOrInsert', cbb);'
```

The UpdateOrInsert option (and UseUpdateOrInsert) is now also valid. This allows you to generate UPDATE OR INSERT statements instead of UPDATE/INSERT for Firebird 2.1 databases (see example above).

Example of usage

```
execute ibeblock
returns (
  TotalTime double precision = 0 comment 'Time spent (seconds)')
as
declare ibeblock cbb (LogData variant, StatData variant)
returns (Threshold integer = 0)
as
begin
  ibec_progress('Records processed: ' || LogData[0]);
  Threshold = 50;
end;
begin
  create connection MasterDB dbname 'localhost:c:\MasterDB.fdb'
  password 'masterkey' user 'SYSDBA'
  clientlib 'C:\Program Files\Firebird\bin\fbclient.dll';

  create connection SubscriberDB dbname 'localhost:c:\SubscriberDB.fdb'
  password 'masterkey' user 'SYSDBA'
  sql_dialect 3
  clientlib 'C:\Program Files\Firebird\bin\fbclient.dll';

  StartTime = ibec_gettickcount();
  ibec_CompareTables(MasterDB, SubscriberDB, 'IBE$$TEST_DATA',
'IBE$$TEST_DATA',
  'E:\CompRes.sql', 'OmitUpdates', cbb);
  ibec_CompareTables(MasterDB, SubscriberDB, 'IBE$$TEST_DATA',
'IBE$$TEST_DATA',
  'E:\CompRes.sql', 'AppendMode; OmitDeletes; OmitInserts;
UpdateAllColumns', cbb);

  close connection MasterDB;
  close connection SubscriberDB;

  EndTime = ibec_gettickcount();
  TotalTime = (EndTime - StartTime) / 1000;
```

```
suspend;  
end
```

Example with the where clause

```
ibec_CompareTables(MasterDB, SubscriberDB, 'MYTABLE', 'MYTABLE',  
'E:\CompRes.sql',  
'OmitUpdates; Where="WHERE RECORD_DATE > '12-NOV-2005'"', cbb);
```

It is also possible to omit the WHERE keyword:

```
Where="RECORD_DATE > '12-NOV-2005' "
```

IBExpert will add it automatically.

You can also use the WhereClause as an optional name:

```
WhereClause="RECORD_DATE > '12-NOV-2005' "
```

From:
<https://www.ibexpert.com/docu/> - **IBExpert**

Permanent link:
https://www.ibexpert.com/docu/doku.php?id=05-ibexpert-ibeblock-functions:05-06-database-functions:ibec_comparetables

Last update: **2023/07/06 03:05**

