Generator (Firebird 2: Sequence)

Generators are automatic sequential counters, spanning the whole database. They are necessary because all operations in Firebird/InterBase® are subject to transaction control.

A generator is a database object and is part of the database's metadata. It is a sequential number, incorporating a whole-numbered 64 bit value BIGINT (SQL dialect 3) since InterBase® 6/Firebird (in SQL dialect 1 it is a 32 bit value INTEGER), that can automatically be inserted into a column. It is often used to ensure a unique value in an internal primary key.

A database can contain any number of generators and they can be used and updated in any transaction. They are the only transaction-independent part of Firebird/InterBase®. For each operation a new number is generated, regardless whether this transaction is ultimately committed or rolled back (this consequently leads to “missing numbers”). Therefore generators are best suited for automatic internal sequential numbering for internal primary keys.

SEQUENCE was introduced in Firebird 2.0. It is the SQL-99-compliant synonym for GENERATOR. SEQUENCE is a syntax term described in the SQL specification, whereas GENERATOR is a legacy InterBase® syntax term.

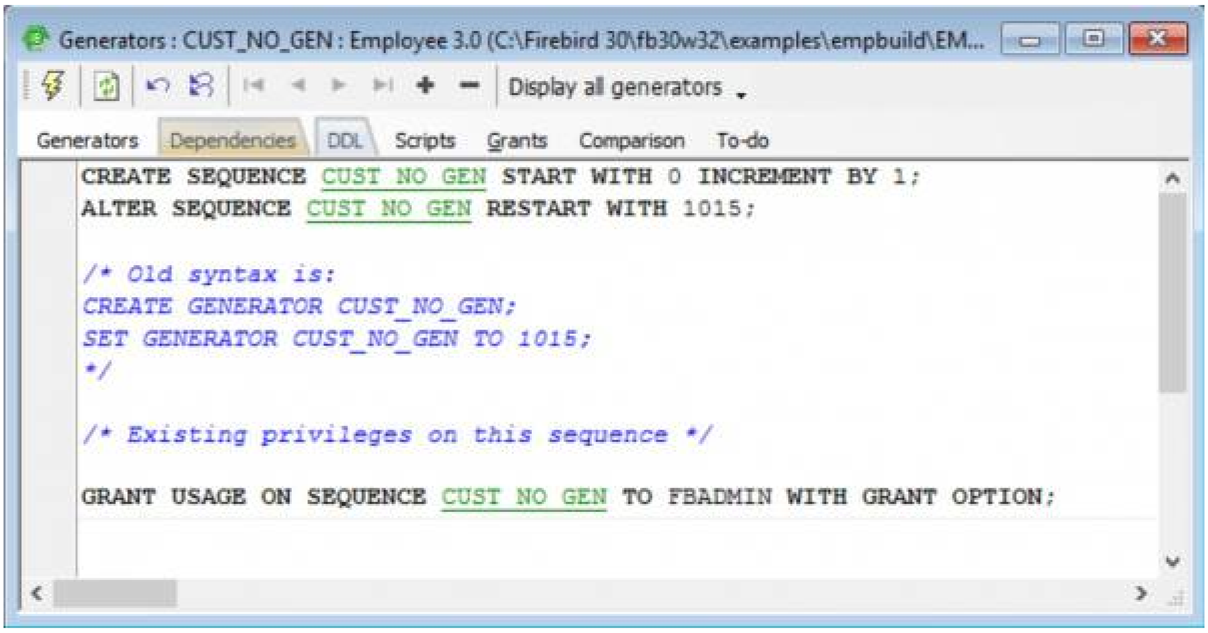
It is recommended Firebird 2.0 users use the standard SEQUENCE syntax:

- CREATE SEQUENCE
- NEXT VALUE FOR
- ALTER SEQUENCE
- DROP SEQUENCE

A sequence generator is a mechanism for generating successive exact numeric values, one at a time. A sequence generator is a named schema object. In dialect 3 it is a BIGINT, in dialect 1 it is an INTEGER. It is often used to implement guaranteed unique IDs for records, to construct columns that behave like AUTOINC fields found in other RDBMSs. Further information regarding SEQUENCE can be found in the Firebird 2.0.4 Release Notes.

Since Firebird 3 it is possible to specify the initial value for autoincrement fields (GENERATED BY IDENTITY) and restart them with the specified value.

For legacy reasons, IBExpert will still continue to use the term GENERATOR alongside the term SEQUENCE.


Generators : CUST_NO_GEN : Employee 3.0 (C:\Firebird 30\fb30w32\examples\empbuild\EM...
Display all generators
Generators
Dependencies
DDL
Scripts
Grants
Comparison
To-do

```
CREATE SEQUENCE CUST_NO_GEN START WITH 0 INCREMENT BY 1;
ALTER SEQUENCE CUST_NO_GEN RESTART WITH 1015;

/* Old syntax is:
CREATE GENERATOR CUST_NO_GEN;
SET GENERATOR CUST_NO_GEN TO 1015;
*/

/* Existing privileges on this sequence */

GRANT USAGE ON SEQUENCE CUST_NO_GEN TO FBADMIN WITH GRANT OPTION;
```

Generators can be created either directly in the SQL Editor or using the DB Explorer (refer to New Generator for details).

Generally a generator is used to determine unique identification numbers for primary keys. A BEFORE INSERT TRIGGER can be defined for this, which increases the current value using the GEN_ID() function, and automatically enters it in the respective table field. Please refer to Create a trigger for a generator for more information. A generator can also be called from a stored procedure or an application.

A database can contain any number of generators. Although up until the most recent InterBase® version 7.x the number of generators was limited to one data page. One generator uses 8 bytes, which means approximately 115 generators fit onto one page (at 1K). This limitation has been solved in the InterBase® 7.x version, and was solved in the Firebird 1.0 version. Using Firebird you can create more than 32,000 generators per database.

The current generator value of existing generators is not stored in a table but on its own system data pages, as the table contents are subject to transactional changes. The generator value is also secured when backing up.

Generators are database objects and are part of the database's metadata, and can be created, modified and dropped as all other Firebird/InterBase® objects in the IBExpert DB Explorer.

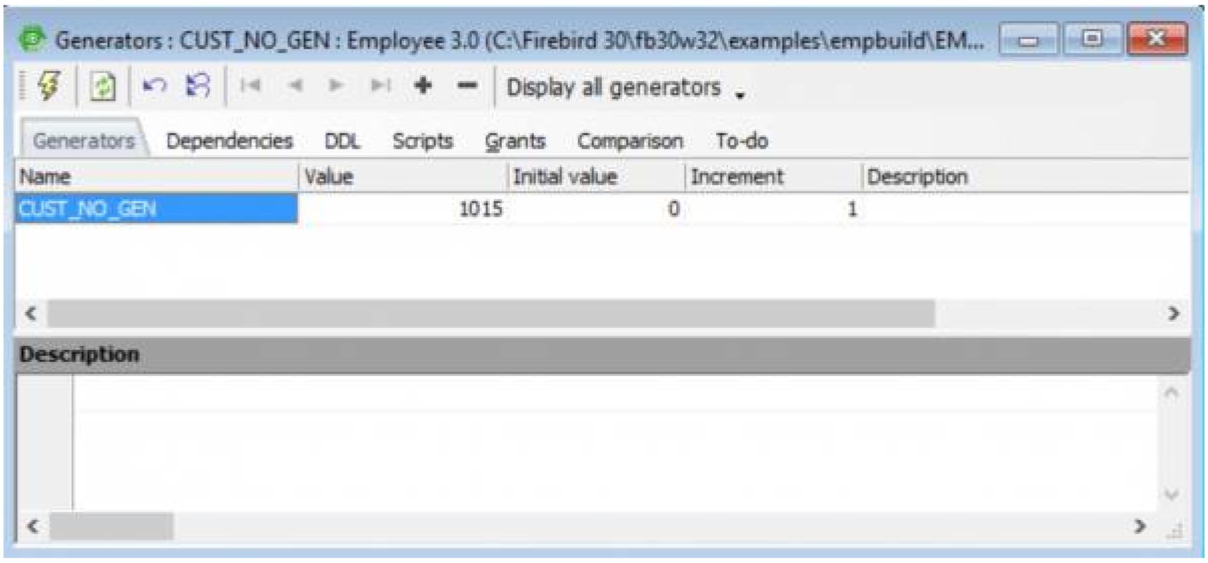
back to top of page

New generator

A new generator can be created in a connected database in a number of ways:

1. By using the menu item Database / New Generator, the respective icon in the New Database Object toolbar, or using the DB Explorer right mouse button (or key combination [Ctrl + N]), when the generator heading of the relevant connected database is highlighted, to start the *New Generator*

Editor:


Generators : CUST_NO_GEN : Employee 3.0 (C:\Firebird 30\fb30w32\examples\empbuild\EM...

Display all generators

Generators | Dependencies | DDL | Scripts | Grants | Comparison | To-do

Name	Value	Initial value	Increment	Description
CUST_NO_GEN	1015	0	1	

Description

2. Alternatively, a new generator can be created in the DB Explorer on the Fields page by double-clicking (or using the space bar when inserting a new field) to check the *Autoinc* box:

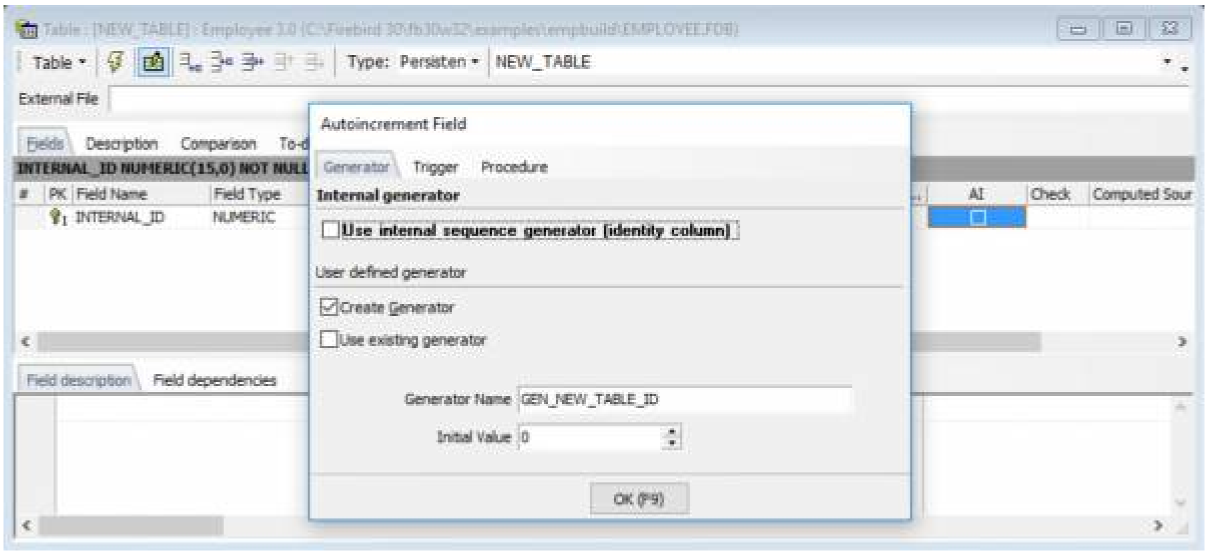


Table : [NEW_TABLE] : Employee 3.0 (C:\Firebird 30\fb30w32\examples\empbuild\EMPLOYEE.FDB)

Type: Persisten NEW_TABLE

INTERNAL_ID NUMERIC(15,0) NOT NULL

#	PK	Field Name	Field Type
1		INTERNAL_ID	NUMERIC

Autoincrement Field

Generator | Trigger | Procedure

Internal generator

☐ Use internal sequence generator [identity column]

User defined generator

☑ Create Generator

☐ Use existing generator

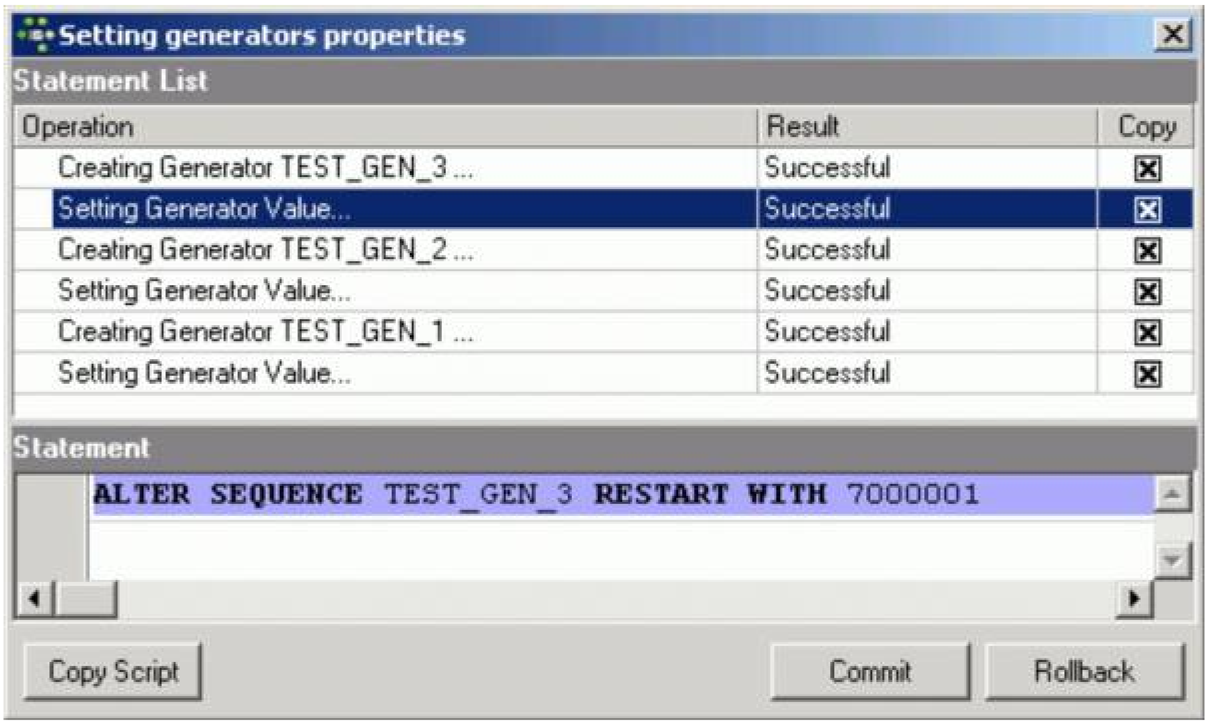
Generator Name: GEN_NEW_TABLE_ID

Initial Value: 0

3. Or in the Field Editor under Autoincrement (started by double-clicking on an existing INTEGER or SMALLINT field in the Table Editor).

4. Or directly in the IBExpert SQL Editor, and then saved as a generator.

Using the the new generator name simply needs to be specified along with the initial generator value. Several generators can be created in the Generator Editor and compiled simultaneously:



Operation	Result	Copy
Creating Generator TEST_GEN_3 ...	Successful	☒
Setting Generator Value...	Successful	☒
Creating Generator TEST_GEN_2 ...	Successful	☒
Setting Generator Value...	Successful	☒
Creating Generator TEST_GEN_1 ...	Successful	☒
Setting Generator Value...	Successful	☒

```
ALTER SEQUENCE TEST_GEN_3 RESTART WITH 7000001
```

Using the *Display all Generators* button on the Generator Editor toolbar, all generators for the database can be listed and an existing generator selected. (For internal numbering purposes, the same generator may be used on several fields, for example all internal primary key IDs, within the database.)

Using the *Autoinc* page in the Table and Field Editors, the *Create Generator* box simply needs to be checked, and the name and starting value defined.

It is also possible to select an existing generator for the specified field here (simply click *Use Existing Generator* and select from the drop-down list):

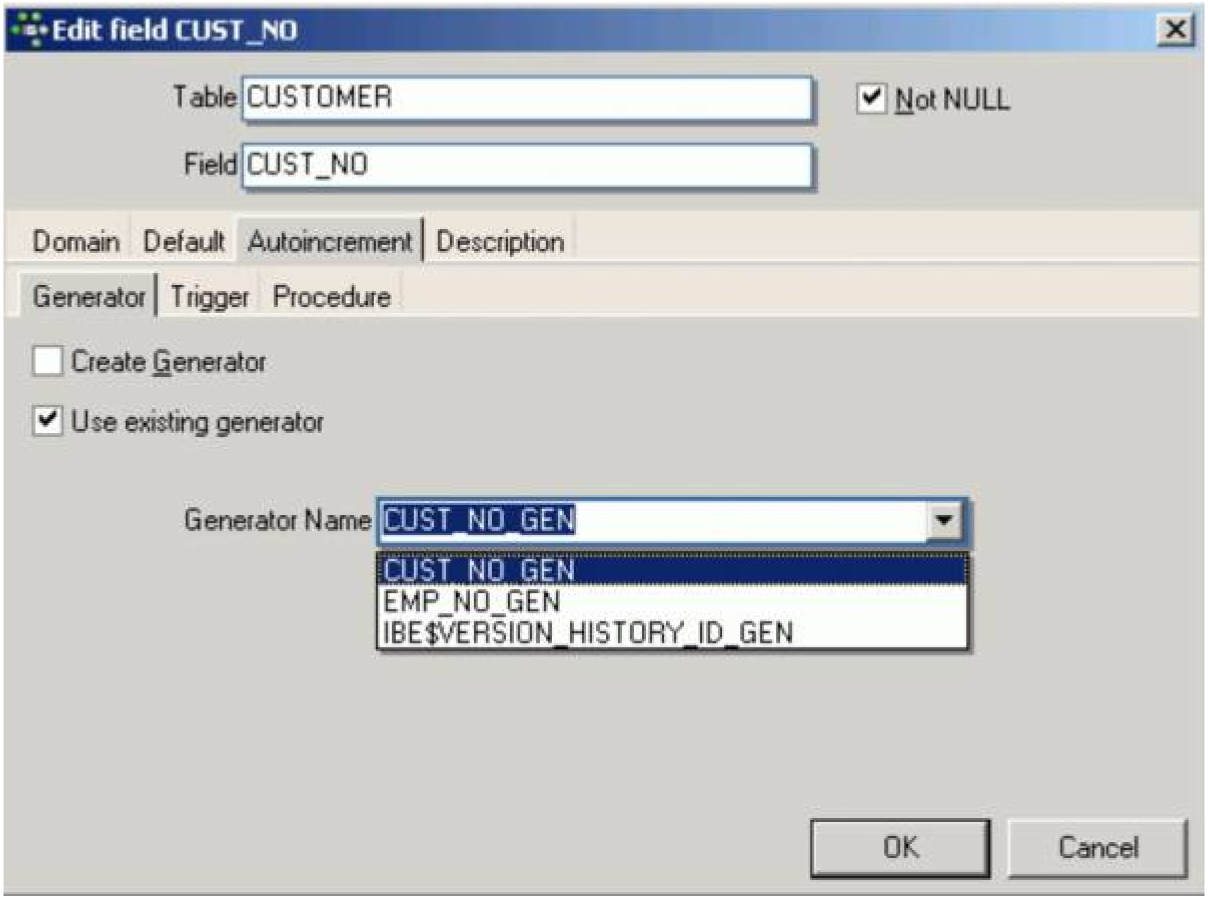


Table: CUSTOMER
Field: CUST_NO
[✓] Not NULL
[ ] Create Generator
[✓] Use existing generator
Generator Name: CUST_NO_GEN

CUST_NO_GEN
EMP_NO_GEN
IBE$VERSION_HISTORY_ID_GEN

For those preferring direct SQL input, the syntax is as follows:

```
CREATE GENERATOR <Generator_Name>;
```

To include a description text when creating generators, add:

```
COMMENT ON SEQUENCE <Generator_Name> IS 'Description'
```

This statement also sets the initial generator value to zero. To establish a different starting value, use the SET GENERATOR statement, for example:

```
SET GENERATOR <Generator_Name> TO n;
```

where n is the initial generator value. SET GENERATOR can also be used to reset an existing generator's value. This however requires care, as usually the column(s) that receives the generator value is/are defined to be unique. For example, you would not normally reset customer IDs except under unusual and controlled circumstances.

To increment the generator use the STEP_VALUE parameter (can be positive or negative):

```
GEN_ID(<Generator_Name>, STEP_VALUE)
```

If this parameter is not used, the default STEP_VALUE with an increment of 1 applies.

back to top of page

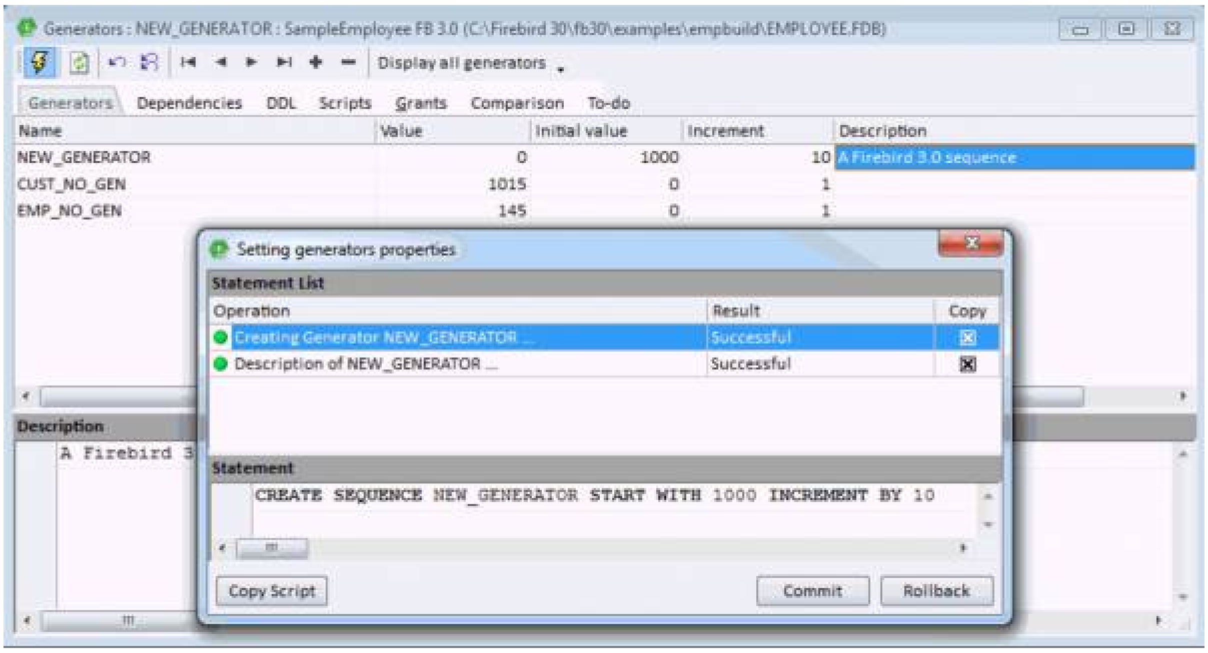
Generator Editor

The Generator Editor can be started using the Database / *New Generator* menu item; from the DB Explorer, using the right mouse-click menu or double-clicking on an existing generator; or directly from the Field or Table Editor / *Autoincrement*.

Please refer to New Generator when creating a generator for the first time.

The Generator Editor has its own toolbar (see Generator Editor toolbar) and offers the following options:

- Generators page
- Dependencies
- DDL
- Scripts
- Comparison
- To-Do



Name	Value	Initial value	Increment	Description
NEW_GENERATOR	0	1000	10	A Firebird 3.0 sequence
CUST_NO_GEN	1015	0	1	
EMP_NO_GEN	145	0	1	

Operation	Result	Copy
Creating Generator NEW_GENERATOR ...	Successful	☒
Description of NEW_GENERATOR ...	Successful	☒

```
CREATE SEQUENCE NEW_GENERATOR START WITH 1000 INCREMENT BY 10
```

Generators page

Here it is possible to create new generators, select an existing generator, and alter a generator. Please refer to New Generator or Alter Generator for details.

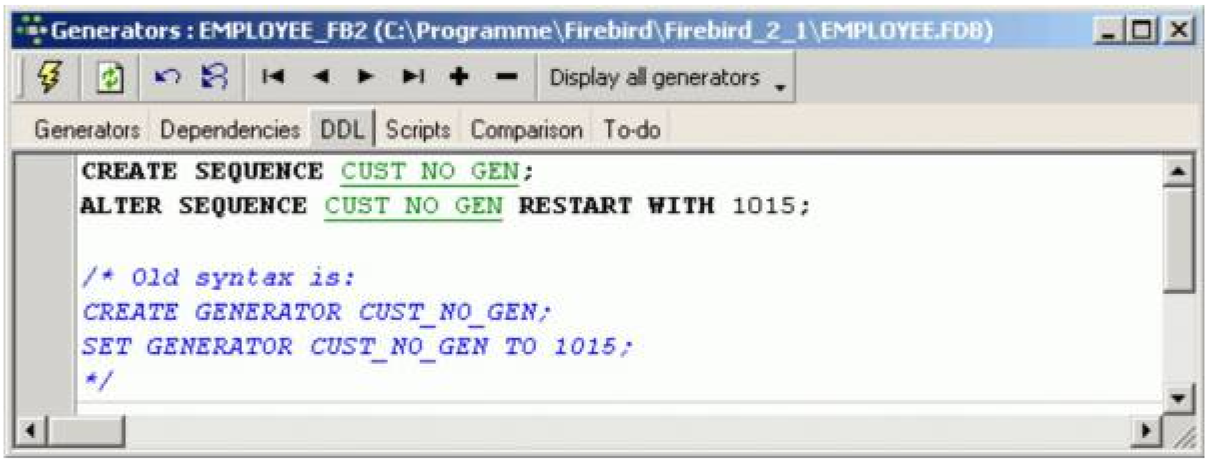
In Firebird 2.0 the RDB$DESCRIPTION field was added to RDB$GENERATORS, so now it is now possible to include a description text when creating generators.

Dependencies

Please refer to Table Editor / Dependencies.

DDL

Please refer to Table Editor / DDL.


Generators : EMPLOYEE_FB2 (C:\Programme\Firebird\Firebird_2_1\EMPLOYEE.FDB)
Display all generators
Generators
Dependencies
DDL
Scripts
Comparison
To-do

```
CREATE SEQUENCE CUST_NO_GEN;
ALTER SEQUENCE CUST_NO_GEN RESTART WITH 1015;

/* Old syntax is:
CREATE GENERATOR CUST_NO_GEN;
SET GENERATOR CUST_NO_GEN TO 1015;
*/
```

Scripts

Creating - displays the CREATE GENERATOR statement for the generator selected on the Generators page. If all generators are displayed on the Generator page (Display All Generators button), all corresponding CREATE statements appear on this page.

Setting Values - displays the SET GENERATOR statement for the generator selected on the *Generators page*. Again, if all generators are displayed on the Generator page (*Display All Generators button*), all SET statements appear on this page.

Full - displays the full SQL text for the generator selected on the *Generators page* (or all generators).

Please note that the *Scripts* page is for display only. It is not possible to make any amendments on this page.

Comparison

Please refer to Table Editor / Comparison.

To-Do

Please refer to Table Editor / To-Do.

back to top of page

Edit generator/alter generator

A generator may be altered to specify a new value. The value of a generator can be changed as often as wished.

This can be performed in IBExpert using the DB Explorer's Generator Editor, opened either by double-clicking on the generator name, or right-clicking and selecting *Edit Generator* [Ctrl + O]. Simply enter the new figure in the *Value* column, compile and commit.

The SQL syntax for altering a sequence is as follows:

```
SET SEQUENCE <sequence_name> RESTART WITH n
```

The SQL syntax for altering a generator is as follows:

```
SET GENERATOR <generator_name> TO n
```

where n is the new value. This new value is immediately effective.

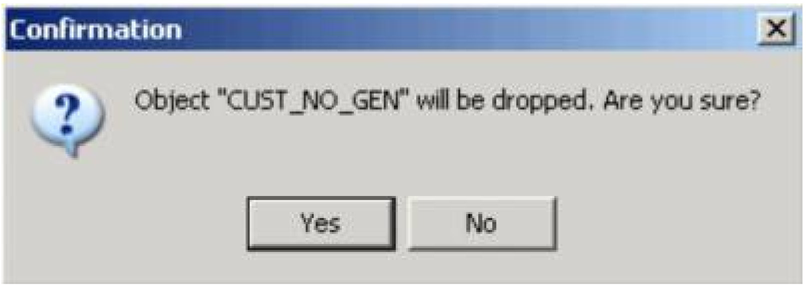
Please refer to the ALTER SEQUENCE and SET GENERATOR statements for further information.

back to top of page

Drop generator/delete generator

In IBExpert, a generator can be dropped from the DB Explorer by selecting the generator to be deleted and using the '-' icon on the Generator Editor toolbar or [Shift + Del].

IBExpert asks for confirmation and displays the SQL statement:



Object "CUST_NO_GEN" will be dropped. Are you sure?

before finally dropping when the statement is committed.

For those preferring to use SQL, the syntax is as follows:

```
DROP GENERATOR <generator_name>;
```

Note: The DROP GENERATOR command was introduced in Firebird 1, and does not exist in earlier InterBase® versions. If you need to delete a generator in an older InterBase® version, you will need to delete it from the system table, RDB$GENERATORS:

```
DELETE FROM RDB$GENERATORS
  WHERE RDB$GENERATOR_NAME='GEN01';
```

Beware that this command deletes the specified generator regardless of any dependencies that may exist.