

DELPHI in Depth : Cds.ApplyUpdates(**MaxErrors**)

Delphi DEV ทั้งหมดที่ใช้ TClientDataset

ผมกำลังจะพาไปทำความเข้าใจ MaxErrors กันแบบรู้ลึกถึงแก่นแท้กันเลยว่า เราควรเลือกใช้

MaxErrors แบบไหน ระหว่าง

Cds.ApplyUpdates(-1);

Cds.ApplyUpdates(0);

Cds.ApplyUpdates(1);

Cds.ApplyUpdates(N); โดย N แทนใดๆ ที่เป็นบวก

ก่อนที่จะไปถึงขั้นตอนการทำ Demo เพื่อให้เข้าใจไอ้ตัว MaxErrors

ขยายความให้รู้จัก MaxErrors กันก่อน

MaxErrors คือ parameter ที่มีไว้เพื่อ ควบคุมเมื่อเกิด Errors ในขณะที่กำลังบันทึกข้อมูล (Updates)

Error ที่เกิดขึ้น ในช่วงเวลาที่ทำการบันทึกข้อมูลนั้น เป็นสิ่งที่ไม่อยากให้เกิดขึ้นแน่นอน แต่เมื่อมันเกิดขึ้นจริง ๆ จะจัดการกับ Error นั้นยังไง

ยกตัวอย่างเช่น กำลังบันทึก 3 record รายการแรก ก็เกิด error เลย จะทำการบันทึกรายการที่ 2 หรือ 3 ต่อไปหรือไม่ หรือยกเลิกการทำงานตอนนี้เลย หรือทำต่อไปอีกสัก 2 หรือ 3 record ก่อนยกเลิกจะ ได้มัย?

Cds.ApplyUpdates(MaxErrors);

MaxErrors Parameter ตัวนี้แหละ ที่จะเป็นตัวช่วยเราในการจัดการกับ Errors ที่จะเกิดขึ้น

ผมกำลังจะพาไปทำความรู้จักกับ MaxErrors แบบละเอียดพร้อมยกตัวอย่างประกอบเพื่อให้เห็นภาพทีละ Step

### MaxErrors = -1

เมื่อ กำหนด MaxErrors เท่ากับ -1

เป็นการบอกโปรแกรมของเราว่า เมื่อเราสั่ง บันทึก (Updates) เมื่อ โปรแกรมพบ Error ให้ข้ามไป และทำการ Updates รายการ record อื่น ๆ ต่อไปเรื่อย ๆ จนกว่าจะหมด (หมดคือข้อมูลใน Change cache) เมื่อทำจนครบทุกรายการที่รอ Updates ใน Change cache แล้วก็ทำการ Commit ลง DB ได้เลย

และไอ้รายการที่เกิด Errors จะค้างอยู่ใน Change cache รอให้เราทำการตรวจสอบพร้อมให้ให้ call ApplyUpdates ได้อีก หรือ จะทำการยกเลิก (CancelUpdates) ก็ได้

ในกรณีที่ไม่มีพบ Error ใด ๆ ก็คือ DataSetProvider จะทำการ Commit ข้อมูลลงไปใน DB ได้เลย

อ่านถึงตรงนี้อาจจะไม่เห็นภาพเท่าไร

มาลองดูตัวอย่างจาก โปรแกรมที่เขียนด้วย Delphi กันดีกว่า

ตัวอย่างนี้ผมใช้ MySQL เป็น DB และมี 1 ตารางเพื่อใช้ทำ Demo ค้าง Data Schema ด้านล่าง

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
▶ (Null)	(Null)	(Null)	(Null)	(Null)

MySQL Database

-----

-- Table structure for customers

-----

```

DROP TABLE IF EXISTS `customers`;

CREATE TABLE `customers` (
  `cust_code` varchar(10) NOT NULL,
  `cust_name` varchar(80) DEFAULT NULL,
  `cust_tel` varchar(25) DEFAULT NULL,
  `sys_create_date` datetime DEFAULT NULL,
  `sys_update_date` datetime DEFAULT NULL,
  PRIMARY KEY (`cust_code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

SET FOREIGN_KEY_CHECKS=1;

```

### **Case 1 Insert 3 Row แบบไม่เกิด Error ใด ๆ เลย แล้วตั้ง Cds.ApplyUpdates(-1);**

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
1	Cust name = 1	Tel 1	29/6/2018 10:22:16	
2	Cust name = 2	Tel 2	29/6/2018 10:22:16	
3	Cust name = 3	Tel 3	29/6/2018 10:22:16	

หลังจากตั้ง ApplyUpdates(-1)

DataSetProvider จะสร้าง INSERT SQL statement ขึ้นเองดังภาพ (เรื่อง DataSetProvider เขียนใน

บทความถัดไป)

Setting properties (For developer only) << Error log >> Provider Monitor

Debug Provider Generate SQL Command Clear all log

```
--> 29/06/2018 10:23:00.60
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`)
values
(?, ?, ?, ?)

--> 29/06/2018 10:23:00.60
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`)
values
(?, ?, ?, ?)

--> 29/06/2018 10:23:00.60
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`)
values
(?, ?, ?, ?)
```

Debug Provider Generate Params

```
--> 29/06/2018 10:23:00.60
Param[0] = 1
Param[1] = Cust name = 1
Param[2] = Tel 1
Param[3] = 29/6/2018 10:22:16

--> 29/06/2018 10:23:00.60
Param[0] = 2
Param[1] = Cust name = 2
Param[2] = Tel 2
Param[3] = 29/6/2018 10:22:16

--> 29/06/2018 10:23:00.60
Param[0] = 3
Param[1] = Cust name = 3
Param[2] = Tel 3
Param[3] = 29/6/2018 10:22:16
```

ลองเปิดดูใน MySQL Database ดูข้อมูล 3 รายการที่ต้องเข้าไปใน DB

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
1	Cust name = 1	Tel 1	2018-06-29 10:22:16	(Null)
2	Cust name = 2	Tel 2	2018-06-29 10:22:16	(Null)
3	Cust name = 3	Tel 3	2018-06-29 10:22:16	(Null)

**Case 2 Insert 3 Row แบบกำหนดให้เกิด Error 2 รายการ แล้วสั่ง Cds.ApplyUpdates(-1);**

โดยจะสั่ง Insert ข้อมูลดังนี้

Test ApplyUpdates(MaxErrors) , While MaxErrors = -1

Append 3 records Save

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
1	New customer 1	Tel 1	29/6/2018 10:29:08	29/6/2018 10:29:17
4	New Customer 4	Tel 2	29/6/2018 10:29:08	29/6/2018 10:29:20
3	New Cusomter 3	Tel 3	29/6/2018 10:29:08	29/6/2018 10:29:25

รายการที่ 1 จะเกิด Error แน่นนอนเพราะจะไปซ้ำกับใน DB

รายการที่ 2 จะบันทึกผ่าน และข้ามไปไปทำรายการที่ 3

รายการที่ 3 จะเกิด Error แน่นนอนเพราะจะไปซ้ำกับใน DB

จะตั้ง Cds.ApplyUpdates(-1);

แล้วคุณผล จะมี 1 รายการ ผ่านเข้าไปใน Database คือ รายการ Cust\_Code : 4

และจะมี 2 รายการค้างใน Change cache ดังภาพ

## Debug Provider

**Debug Provider Generate SQL Command**

Clear all log

**Debug Provider Generate Params**

```

--> 29/06/2018 10:23:00.60
insert into `customers`
(`cust_code`,`cust_name`,`cust_tel`,`sys_create_date`)
values
(?, ?, ?, ?)

--> 29/06/2018 10:23:00.60
insert into `customers`
(`cust_code`,`cust_name`,`cust_tel`,`sys_create_date`)
values
(?, ?, ?, ?)

--> 29/06/2018 10:23:00.60
insert into `customers`
(`cust_code`,`cust_name`,`cust_tel`,`sys_create_date`)
values
(?, ?, ?, ?)

--> 29/06/2018 10:33:02.62
insert into `customers`
(`cust_code`,`cust_name`,`cust_tel`,`sys_create_date`,`sys_update_date`)
values
(?, ?, ?, ?, ?)

```

```

--> 29/06/2018 10:23:00.60
Param[0] = 1
Param[1] = Cust name = 1
Param[2] = Tel 1
Param[3] = 29/6/2018 10:22:16

--> 29/06/2018 10:23:00.60
Param[0] = 2
Param[1] = Cust name = 2
Param[2] = Tel 2
Param[3] = 29/6/2018 10:22:16

--> 29/06/2018 10:23:00.60
Param[0] = 3
Param[1] = Cust name = 3
Param[2] = Tel 3
Param[3] = 29/6/2018 10:22:16

--> 29/06/2018 10:33:02.62
Param[0] = 4
Param[1] = New Customer 4
Param[2] = Tel 2
Param[3] = 29/6/2018 10:29:08
Param[4] = 29/6/2018 10:29:20

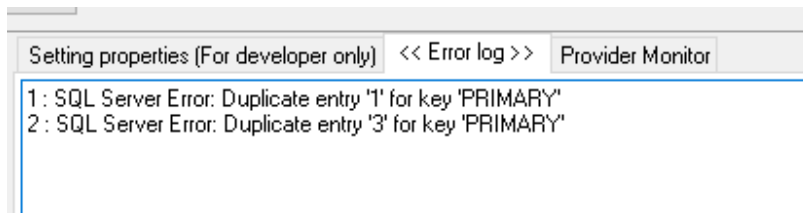
```

Insert ลง 1 รายการคือ cust\_code = 4

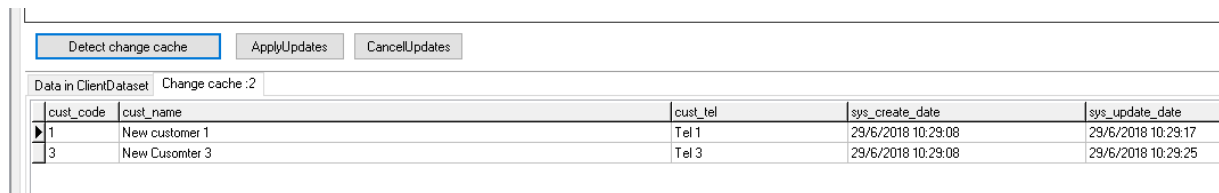
ตรวจสอบ ใน MySQL Database

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
1	Cust name = 1	Tel 1	2018-06-29 10:22:16	(Null)
2	Cust name = 2	Tel 2	2018-06-29 10:22:16	(Null)
3	Cust name = 3	Tel 3	2018-06-29 10:22:16	(Null)
4	New Customer 4	Tel 2	2018-06-29 10:29:08	2018-06-29 10:29:20

กลับมาที่ Program จะเห็น Errors 2 รายการดังภาพ

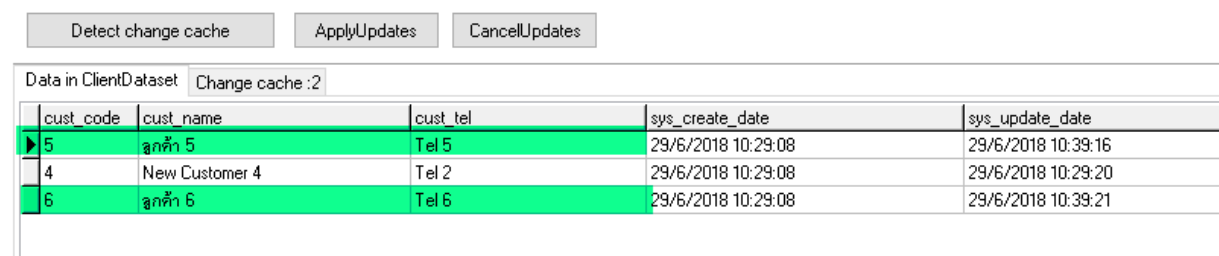


และจะมี Change cache ค้างอยู่ 2 รายการ คือรายการที่ 1 และรายการที่ 3 ดังภาพ



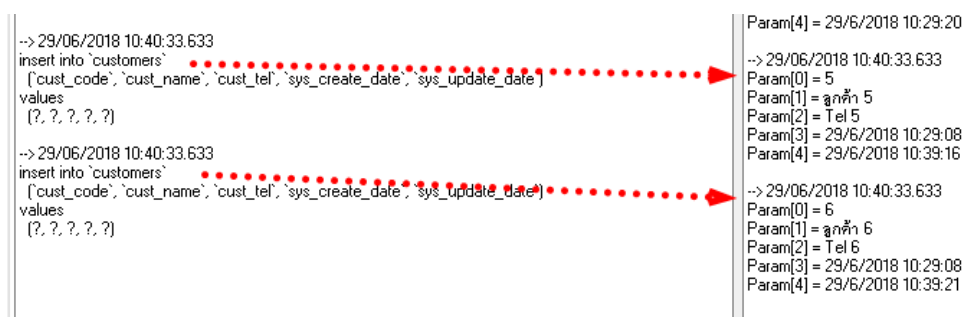
ในหนังสือที่เขียนเกี่ยวกับ ClientDataset บอกว่า รายการที่ค้างใน Change Cache สามารถแก้ไขให้ถูกต้อง และพร้อมที่จะสั่ง ApplyUpdates ได้อีก โดย จะขอแก้ cust\_code : 3 ไปเป็น 5 และ cust\_code : 4 ไปเป็น 6 แล้วทำการ ApplyUpdates อีกรอบ

ดูภาพประกอบ



หลังจากแก้ไขข้อมูลใน Data in ClientDataset กด ApplyUpdates

Debug Provider จะทำการสร้าง INSERT SQL statement 2 รายการดังภาพ



ทำการตรวจสอบที่ MySQL DB (จะเห็นข้อมูลเข้าไปใน Database 2 รายการดังภาพ)

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
▶ 1	Cust name = 1	Tel 1	2018-06-29 10:22:16	(Null)
2	Cust name = 2	Tel 2	2018-06-29 10:22:16	(Null)
3	Cust name = 3	Tel 3	2018-06-29 10:22:16	(Null)
4	New Customer 4	Tel 2	2018-06-29 10:29:08	2018-06-29 10:29:20
5	ลูกค้า 5	Tel 5	2018-06-29 10:29:08	2018-06-29 10:39:16
6	ลูกค้า 6	Tel 6	2018-06-29 10:29:08	2018-06-29 10:39:21

ต่อไปจะลองตรวจสอบ Change cache ว่ามีรายการอะไรค้างรอการ Updates หรือไม่

โปรแกรมแจ้ง Change cache = 0 ดังภาพ

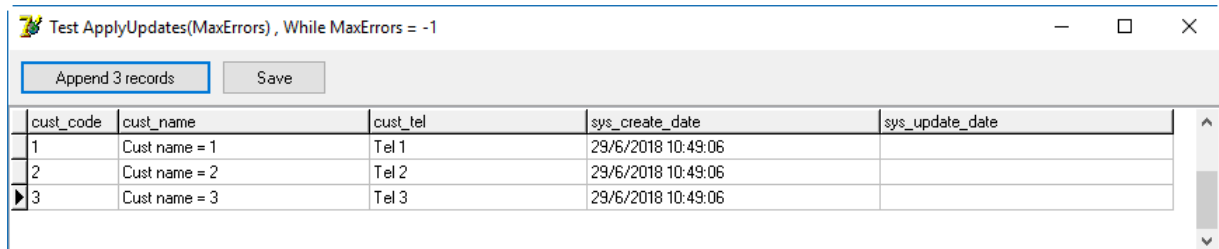
Data in ClientDataset **Change cache :0**

cust_code	cust_name
▶	

### **Case 3 Insert 3 Row แบบกำหนดให้เกิด Error 3 รายการ แล้วสั่ง Cds.ApplyUpdates(-1):**

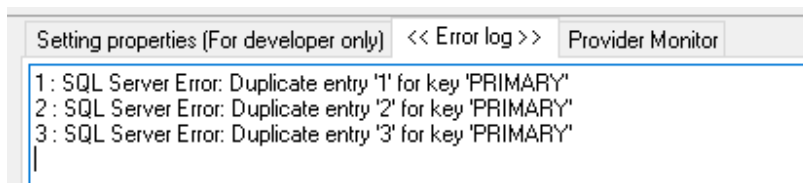
หลังจาก Error จะมีข้อมูลค้างใน Change cache 3 รายการ จากนั้นจะทำการยกเลิกทิ้งไปทั้ง 3 รายการ

สั่งให้ Insert 3 Row ด้วย cust\_code = 1, 2, 3



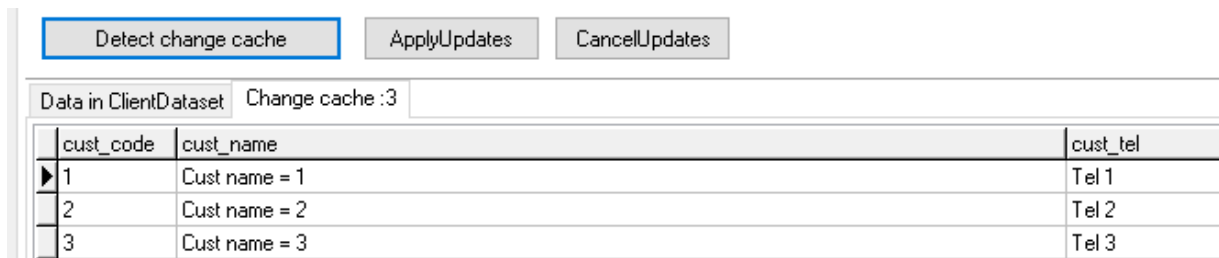
cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
1	Cust name = 1	Tel 1	29/6/2018 10:49:06	
2	Cust name = 2	Tel 2	29/6/2018 10:49:06	
3	Cust name = 3	Tel 3	29/6/2018 10:49:06	

จากนั้น ApplyUpdates(-1) จะเกิด Errors 3 รายการดังภาพ



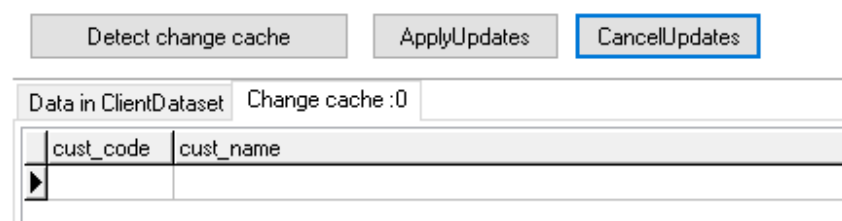
```
Setting properties (For developer only) << Error log >> Provider Monitor
1: SQL Server Error: Duplicate entry '1' for key 'PRIMARY'
2: SQL Server Error: Duplicate entry '2' for key 'PRIMARY'
3: SQL Server Error: Duplicate entry '3' for key 'PRIMARY'
```

และมี Change cache ค้างอยู่ 3 รายการ ดังภาพ



cust_code	cust_name	cust_tel
1	Cust name = 1	Tel 1
2	Cust name = 2	Tel 2
3	Cust name = 3	Tel 3

สั่งยกเลิกทิ้ง ทั้ง 3 รายการ ที่ค้างใน Change cache ด้วยคำสั่ง Cds.CancelUpdates; (กดปุ่ม CancelUpdates)



cust_code	cust_name



หลังจากสั่ง CancelUpdates ข้อมูลใน Change cache จะหายไปหมด คือหากตรวจสอบด้วย คำสั่ง Cds.ChangeCount จะได้ค่า 0

#### **Case 4 Insert 3 Row โดยกำหนดไม่ให้มี Error เลย แล้วสั่ง Cds.ApplyUpdates(0);**

Test ApplyUpdates(MaxErrors) , While MaxErrors = 0

Append 3 records    Save

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
7	ลูกค้า 7	7	29/6/2018 10:59:20	29/6/2018 10:59:35
8	ลูกค้า 8	8	29/6/2018 10:59:20	29/6/2018 10:59:36
9	ลูกค้า 9	9	29/6/2018 10:59:20	29/6/2018 10:59:36

ทำการ insert 3 row โดยกำหนด cust\_code = 7,8,9 ซึ่งจะไม่เกิด Error แน่ๆ เนื่องจากในฐานข้อมูลมี cust\_code = 1,2,3,4,5,6 จากนั้นสั่ง ApplyUpdates(0)

Debug provider

Debug Provider Generate SQL Command	Clear all log	Debug Provider Generate Params
<pre>--&gt; 29/06/2018 11:01:23.623 insert into 'customers' ('cust_code', 'cust_name', 'cust_tel', 'sys_create_date', 'sys_update_date') values (?, ?, ?, ?, ?)  --&gt; 29/06/2018 11:01:23.623 insert into 'customers' ('cust_code', 'cust_name', 'cust_tel', 'sys_create_date', 'sys_update_date') values (?, ?, ?, ?, ?)  --&gt; 29/06/2018 11:01:23.623 insert into 'customers' ('cust_code', 'cust_name', 'cust_tel', 'sys_create_date', 'sys_update_date') values (?, ?, ?, ?, ?)</pre>		<pre>--&gt; 29/06/2018 11:01:23.623 Param[0] = 7 Param[1] = ลูกค้า 7 Param[2] = 7 Param[3] = 29/6/2018 10:59:20 Param[4] = 29/6/2018 10:59:35  --&gt; 29/06/2018 11:01:23.623 Param[0] = 8 Param[1] = ลูกค้า 8 Param[2] = 8 Param[3] = 29/6/2018 10:59:20 Param[4] = 29/6/2018 10:59:36  --&gt; 29/06/2018 11:01:23.623 Param[0] = 9 Param[1] = ลูกค้า 9 Param[2] = 9 Param[3] = 29/6/2018 10:59:20 Param[4] = 29/6/2018 10:59:36</pre>

ตรวจสอบใน Database server จะเห็น 3 รายการเข้าไปใน Database

4	New Customer 4	Tel 2	2018-06-29 10:29:08	2018-06-29 10:29:08
5	ลูกค้า 5	Tel 5	2018-06-29 10:29:08	2018-06-29 10:29:08
6	ลูกค้า 6	Tel 6	2018-06-29 10:29:08	2018-06-29 10:29:08
7	ลูกค้า 7	7	2018-06-29 10:59:20	2018-06-29 10:59:20
8	ลูกค้า 8	8	2018-06-29 10:59:20	2018-06-29 10:59:20
9	ลูกค้า 9	9	2018-06-29 10:59:20	2018-06-29 10:59:20

ตรวจสอบ Change cache

Data in ClientDataset **Change cache :0**

cust_code	cust_name

**Case 5 Insert 3 Row โดยกำหนดให้มี Error 1 รายการ ในรายการที่ 2 จากนั้นสั่ง**

**Cds.ApplyUpdates(0);**

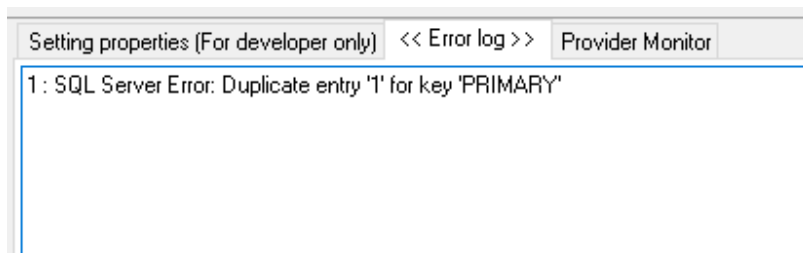
ในรายการ ที่ 1 และรายการที่ 3 จะต้องเกิด Error เรื่อง PRIMARY KEY แน่แน่นอน เนื่องจากใน Database ของเรานั้นมี รายการ ดังกล่าวอยู่แล้ว ส่วนรายการที่ 2 ต้อง Insert ผ่าน ลองสั่ง Updates ด้วย ApplyUpdates(0) เพื่อดูผล

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
1	Cust name = 1	Tel 1	29/6/2018 11:05:30	
10	ลูกค้า 10	1010	29/6/2018 11:05:30	29/6/2018 11:07:51
3	Cust name = 3	Tel 3	29/6/2018 11:05:30	

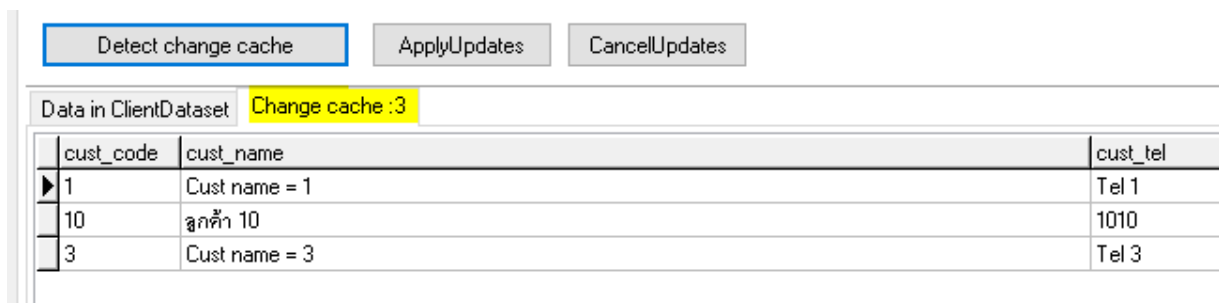
Debug provider ไม่มีรายการใด ๆ ทำงานเลย



มี Error 1 รายการใน Error log



มี Change cache ค้างอยู่ 3 รายการ



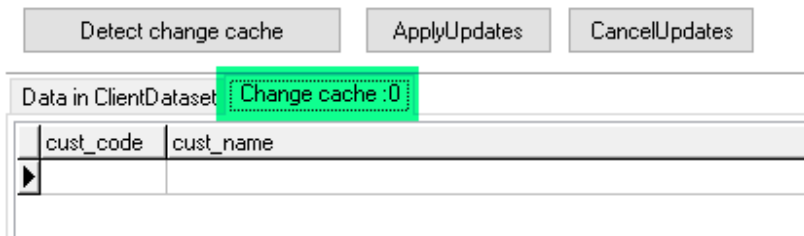
เนื่องจาก Case 5 นี้ ทำการ Test ApplyUpdates(0) และมี Error รายการที่ 1 เลย

โปรแกรมจึงยกเลิกการทำงานทั้งหมด เนื่องจาก สั่ง ApplyUpdates โดยใช้ MaxErrors = 0

นั่นหมายความว่า ไม่ต้องการให้เกิด Error ใด ๆ เลย จึงจะบันทึกข้อมูลลงใน MySQL Database จริง

MaxErrors = 0 ขอมให้เกิด Errors ได้แค่ 0 ครั้ง หรือไม่ให้เกิดเลยนั่นเอง

กลับไปโปรแกรม จะสั่ง CancelUpdates 3 รายการ ช่างค้นไปก่อนด้วยกด สั่ง CancelUpdates  
จะได้ดังภาพด้านล่าง



### **Case 6 Insert 3 Row โดยกำหนดให้มี Error 2 รายการ ในรายการที่ 2 และ 3 จากนั้นสั่ง**

#### **Cds.ApplyUpdates(0);**

รายที่ 1 จะไม่เกิด Error (DatasetProvider จะสร้าง INSERT SQL statement ได้)

รายที่ 2 และ 3 จะเกิด Error (PRIMARY)

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
10	Customer 10	10	29/6/2018 11:15:52	29/6/2018 11:16:14
7		7	29/6/2018 11:15:52	29/6/2018 11:16:12
8		8	29/6/2018 11:15:52	29/6/2018 11:16:13

เมื่อโปรแกรมทำการ ApplyUpdates(0) ในรายการที่ 1 จะไม่เกิด Error ใด ๆ

DatasetProvider จะสร้าง INSERT SQL statement ได้ดังภาพ (กำลังจะ Insert Cust\_Code : 10)

## Debug Provider Generate SQL Command

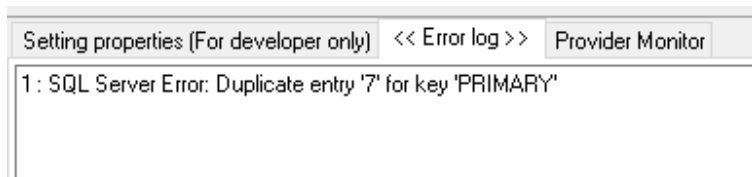
Clear all log

```
--> 29/06/2018 11:18:13.613
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`, `sys_update_date`)
values
(?, ?, ?, ?, ?)
```

## Debug Provider Generate Params

```
--> 29/06/2018 11:18:13.613
Param[0] = 10
Param[1] = Customer 10
Param[2] = 10
Param[3] = 29/6/2018 11:15:52
Param[4] = 29/6/2018 11:16:14
```

เมื่อโปรแกรม ApplyUpdates(0) รายการที่ 2 จะพบ Error และฟ้อง Error ออกมา



ตรวจสอบ Change cache จะเห็นว่ามี 3 รายการที่ค้างใน Change cache

นั่นหมายความว่าเมื่อโปรแกรม พบ Error ครั้งที่ 1 โปรแกรมแสดง Error ออกมาและจะยกเลิกการทำงานทั้งหมดทันที และสั่ง Rollback รายการที่รอ Commit ด้วย ล่องไปอยู่ใน MySQL Database จะเห็นว่า รายการ cust\_code = 10 ยังไม่เข้าไปในฐานข้อมูลจริง ดังภาพประกอบ

cust_code	cust_name	cust_tel	sys_create_date
1	Cust name = 1	Tel 1	2018-06-29 10:22:16
2	Cust name = 2	Tel 2	2018-06-29 10:22:16
3	Cust name = 3	Tel 3	2018-06-29 10:22:16
4	New Customer 4	Tel 2	2018-06-29 10:29:08
5	ลูกค้า 5	Tel 5	2018-06-29 10:29:08
6	ลูกค้า 6	Tel 6	2018-06-29 10:29:08
7	ลูกค้า 7	7	2018-06-29 10:59:20
8	ลูกค้า 8	8	2018-06-29 10:59:20
9	ลูกค้า 9	9	2018-06-29 10:59:20

กลับไปโปรแกรม สั่ง CancelUpdates

Data in ClientDataset    Change cache :0

	cust_code	cust_name
▶		

สรุป การเลือกใช้ ApplyUpdates(0)

สั้น ๆ ง่าย ต้องไม่เกิด Error ใด ๆ เลย จึงจะทำการ Commit ลงฐานข้อมูลจริงๆ

และหากเกิด Error ขึ้น ในครั้งที่ 1 โปรแกรมจะยกเลิกการทำงานทันที

และจะยังคงค้าง Change cache ไว้ทั้งหมด ซึ่งสามารถ แก้ไขและ Updates ไปอีกรอบ หรือจะทำการ CancelUpdates ทิ้งเสียก็ได้

\*\* เปรียบเทียบ

### ApplyUpdates(-1) vs ApplyUpdates(0)

หากเป็น ApplyUpdates(0) ต้องผ่านใน ๆ ทุก ๆ รายการใน Change cache จึงจะ Commit ลงฐาน

หากเป็น ApplyUpdates(-1) พบ Error ก็ข้ามไป หากพบ อีกก็ข้ามไปเรื่อย จนหมด Change cache

จากนั้น จะ Commit รายการที่ไม่เกิด Error และจะเหลือ Change cache แค่รายการที่ Updates ไม่ผ่าน

## ApplyUpdates(1) หรือ 2 หรือ ...N

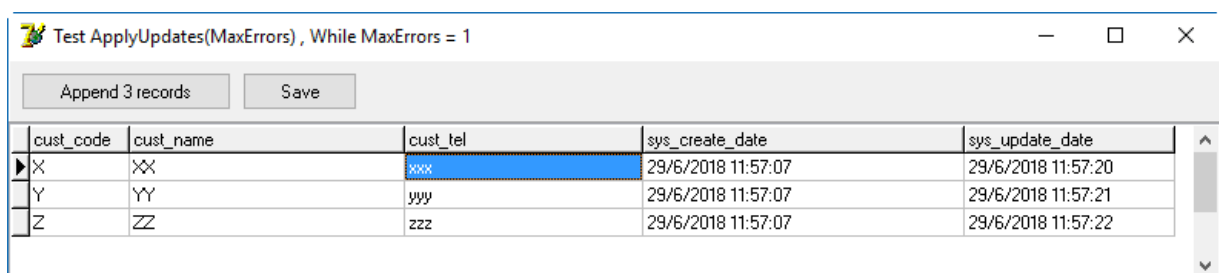
ในที่ขอยกตัวอย่าง ApplyUpdates(1)

ความหมายสั้นของ MaxErrors = 1 ขอมให้เกิด Error ได้ น้อยกว่าหรือเท่ากับ MaxErrors นั่นคือ 1 ครั้ง จากตัวอย่างนี้

หากเป็นไปตามเงื่อนไขข้างต้น โปรแกรม จะทำการ Commit ข้อมูลไปลงฐานให้ และรายการที่ Updates ไม่ผ่าน จะค้างอยู่ใน Change cache

ในทางตรงกันข้ามเมื่อ เกิด Errors มากกว่า MaxErrors หรือจากตัวอย่างนี้ หากมากกว่า 1 โปรแกรม จะ Rollback (ไม่บันทึกลงฐานข้อมูลจริง) และจะมีรายการใน Change cache ทั้งหมด รอให้แก้ไข เพื่อ Updates อีกรอบ หรือ CancelUpdates

## Case 7 Insert 3 Row โดยกำหนด ไม่ให้มี Error จากนั้นสั่ง Cds.ApplyUpdates(1)



cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
X	XX	xxx	29/6/2018 11:57:07	29/6/2018 11:57:20
Y	YY	yyy	29/6/2018 11:57:07	29/6/2018 11:57:21
Z	ZZ	zzz	29/6/2018 11:57:07	29/6/2018 11:57:22

ทดสอบด้วยข้อมูลที่ไม่เกิด Error ซึ่งผ่านทุกขั้นตอนแน่นอน

Debug Provider

```

--> 29/06/2018 11:58:09.69
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`, `sys_update_date`)
values
(?, ?, ?, ?, ?)
--> 29/06/2018 11:58:09.69
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`, `sys_update_date`)
values
(?, ?, ?, ?, ?)
--> 29/06/2018 11:58:09.69
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`, `sys_update_date`)
values
(?, ?, ?, ?, ?)

```

.....→

```

Param[4] = 29/6/2018 11:16:14
--> 29/06/2018 11:58:09.69
Param[0] = X
Param[1] = XX
Param[2] = xxx
Param[3] = 29/6/2018 11:57:07
Param[4] = 29/6/2018 11:57:20
--> 29/06/2018 11:58:09.69
Param[0] = Y
Param[1] = YY
Param[2] = yyy
Param[3] = 29/6/2018 11:57:07
Param[4] = 29/6/2018 11:57:21
--> 29/06/2018 11:58:09.69
Param[0] = Z
Param[1] = ZZ
Param[2] = zzz
Param[3] = 29/6/2018 11:57:07
Param[4] = 29/6/2018 11:57:22

```

ไม่มี Change cache หรือ Cds.ChangeCount มีค่าเท่ากับ 0

Data in ClientDataset    Change cache : 0

	cust_code	cust_name
▶		

ข้อมูลลง Database แนนอน (ตรวจสอบใน MySQL Database)



cust_code	cust_name	cust_tel	sys_create
1	Cust name = 1	Tel 1	2018-06-2
2	Cust name = 2	Tel 2	2018-06-2
3	Cust name = 3	Tel 3	2018-06-2
4	New Customer 4	Tel 2	2018-06-2
5	ลูกค้า 5	Tel 5	2018-06-2
6	ลูกค้า 6	Tel 6	2018-06-2
7	ลูกค้า 7	7	2018-06-2
8	ลูกค้า 8	8	2018-06-2
9	ลูกค้า 9	9	2018-06-2
X	XX	xxx	2018-06-2
Y	YY	yyy	2018-06-2
Z	ZZ	zzz	2018-06-2

**Case 7 Insert 4 Row โดยกำหนดให้มี Error 1 ครั้ง ใน รายการที่ 3 จากนั้นสั่ง**

### **Cds.ApplyUpdates(1)**

รายการที่ 3 จะเกิด Error เรื่อง PRIMARY KEY

โดย Test case นี้จะ insert ผ่าน และข้อมูลจะลงไปฐานข้อมูลจริง 3 record

คือ cust\_code = 10,11 ,12

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
10	ลูกค้า 10	लगरानने 10	29/6/2018 12:01:10	29/6/2018 12:02:04
11	ลูกค้า 11	लगरानने 10	29/6/2018 12:01:10	29/6/2018 12:02:07
3	ทดสอบ ERROR 3	ไม่ลง DB	29/6/2018 12:01:10	29/6/2018 12:01:56
12	ลูกค้า 12	लगरानने 12	29/6/2018 12:01:20	29/6/2018 12:02:09

## Debug provider

```

--> 29/06/2018 12:02:52.652
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`, `sys_update_date`)
values
(?, ?, ?, ?, ?)
--> 29/06/2018 12:02:52.652
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`, `sys_update_date`)
values
(?, ?, ?, ?, ?)
--> 29/06/2018 12:02:53.653
insert into `customers`
(`cust_code`, `cust_name`, `cust_tel`, `sys_create_date`, `sys_update_date`)
values
(?, ?, ?, ?, ?)

```

```

Param[1] = ZZ
Param[2] = zzz
Param[3] = 29/6/2018 11:57:07
Param[4] = 29/6/2018 11:57:22

--> 29/06/2018 12:02:52.652
Param[0] = 10
Param[1] = ลูกค้า 10
Param[2] = ลอรานนท์ 10
Param[3] = 29/6/2018 12:01:10
Param[4] = 29/6/2018 12:02:04

--> 29/06/2018 12:02:52.652
Param[0] = 11
Param[1] = ลูกค้า 11
Param[2] = ลอรานนท์ 10
Param[3] = 29/6/2018 12:01:10
Param[4] = 29/6/2018 12:02:07

--> 29/06/2018 12:02:53.653
Param[0] = 12
Param[1] = ลูกค้า 12
Param[2] = ลอรานนท์ 12
Param[3] = 29/6/2018 12:01:20
Param[4] = 29/6/2018 12:02:09

```

## โปรแกรมจะพบ Error 1 รายการ

```

Setting properties (For developer only) << Error log >> Provider Monitor
1: SQL Server Error: Duplicate entry '3' for key 'PRIMARY'

```

และจะมี Change cache = 1 (Cds.ChangeCount = 1) กด CancelUpdates เพื่อยกเลิกรายการที่ updates ไม่ผ่านทิ้งไปก่อน เพื่อรอทำ Test case ต่อไป

```

Detect change cache ApplyUpdates CancelUpdates
Data in ClientDataset Change cache :1

```

cust_code	cust_name	cust_tel
▶ 3	ทดสอบ ERROR 3	ไม่ลง DB

เข้าไปดูข้อมูล ใน Database ข้อมูล ถูก Commit ลงฐานจริง ๆ

cust_code	cust_name	cust_tel	sys_create_date
1	Cust name = 1	Tel 1	2018-06-29 10:22:16
10	ลูกค้า 10	สงฐานแน 10	2018-06-29 12:01:10
11	ลูกค้า 11	สงฐานแน 10	2018-06-29 12:01:10
12	ลูกค้า 12	สงฐานแน 12	2018-06-29 12:01:20
2	Cust name = 2	Tel 2	2018-06-29 10:22:16

**Case 8 Insert 5 Row โดยกำหนดให้มี Error 2 ครั้ง ใน รายการที่ 3 และ 4 จากนั้นสั่ง**

### Cds.ApplyUpdates(1)

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
13	ลูกค้า 13	ลูกค้า 13	29/6/2018 12:06:20	29/6/2018 12:07:11
14	ลูกค้า 14	ลูกค้า 14	29/6/2018 12:06:20	29/6/2018 12:07:15
1	จะเกิด Error และค้าง Change cache	ลูกค้า Error	29/6/2018 12:06:20	29/6/2018 12:07:36
2	จะเกิด Error และค้าง Change cache	ลูกค้า Error	29/6/2018 12:06:27	29/6/2018 12:07:48
17	ลูกค้า 17	ลูกค้า 17	29/6/2018 12:06:30	29/6/2018 12:07:07

เข้าไปดูใน Debug provider

```

--> 29/06/2018 12:28:17.617
insert into `customers`
(`cust_code`,`cust_name`,`cust_tel`,`sys_create_date`,`sys_update_date`)
values
(?, ?, ?, ?, ?)

--> 29/06/2018 12:28:17.617
insert into `customers`
(`cust_code`,`cust_name`,`cust_tel`,`sys_create_date`,`sys_update_date`)
values
(?, ?, ?, ?, ?)

```

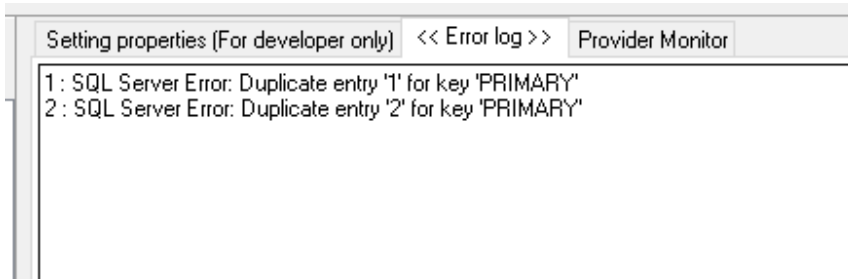
```

Param[0] = 13
Param[1] = ลูกค้า 13
Param[2] = ลูกค้า 13
Param[3] = 29/6/2018 12:06:20
Param[4] = 29/6/2018 12:07:11

--> 29/06/2018 12:28:17.617
Param[0] = 14
Param[1] = ลูกค้า 14
Param[2] = ลูกค้า 14
Param[3] = 29/6/2018 12:06:20
Param[4] = 29/6/2018 12:07:15

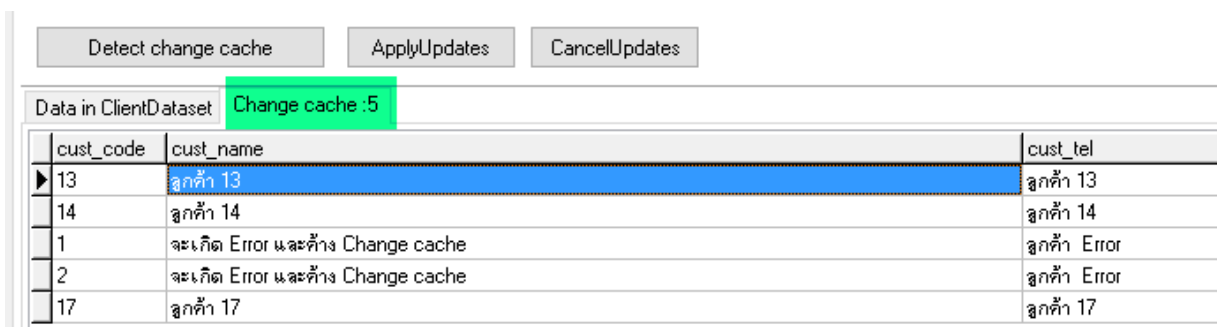
```

จะเห็น 2 รายการที่ DataSetProvider สร้าง INSERT SQL statement ให้  
คือรายการที่ 1 และ รายการที่ 2 จากนั้นโปรแกรมพบ Error สองครั้ง ติดกัน ดังภาพ



Case นี้ ได้กำหนด MaxErrors = 1

จะเห็นว่า มี Error = 2 ดังนั้น โปรแกรมจะทำการ Rollback และล้าง Change cache ไว้ทั้งหมด 5 รายการ (Cds.ChangeCount = 5) ไม่มีการ Commit รายการใด ๆ ลงไปใน Database  
ดังภาพ



ตรวจสอบใน MySQL DB

จะไม่พบรายการ ที่มี cust\_code = 13, 14 และ 17

cust_code	cust_name	cust_tel
1	Cust name = 1	Tel 1
10	ลูกค้า 10	สงฐานแน 1
11	ลูกค้า 11	สงฐานแน 1
12	ลูกค้า 12	สงฐานแน 1.
2	Cust name = 2	Tel 2
3	Cust name = 3	Tel 3
4	New Customer 4	Tel 2
5	ลูกค้า 5	Tel 5
6	ลูกค้า 6	Tel 6
7	ลูกค้า 7	7
8	ลูกค้า 8	8
9	ลูกค้า 9	9
X	XX	xxx
Y	YY	yyy
Z	ZZ	zzz

กลับไปปิด CancelUpdates ที่โปรแกรม เพื่อ Test case ต่อไป

**Case 9 Insert 6 Row โดยกำหนดให้มี Error 3 ครั้ง ใน รายการที่ 2, 5 และ 6 จากนั้นสั่ง**

**Cds.ApplyUpdates(1)**

กำหนดให้ รายการที่ 2, 5, และ 6 มีเลขที่ cust\_code ที่ซ้ำกับที่มีอยู่ในฐานข้อมูล

จะเกิด Error PRIMARY KEY

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
A	A	A	29/6/2018 12:35:02	29/6/2018 12:35:16
1	B ERR	B	29/6/2018 12:35:02	29/6/2018 12:36:08
C	C	C	29/6/2018 12:35:02	29/6/2018 12:35:18
D	D	D	29/6/2018 12:35:09	29/6/2018 12:35:19
2	E ERR	E	29/6/2018 12:35:09	29/6/2018 12:36:10
▶	F ERR	F	29/6/2018 12:35:10	29/6/2018 12:36:12

## Debug provider

ตรงนี้ DatasetProvider จะทำการสร้าง INSERT SQL statement ขึ้น คือรายการที่มี

cust\_code = A

cust\_code = B

เกิด Error ครั้งที่ 1 (SQL Server Error: Duplicate entry '1' for key 'PRIMARY')

cust\_code = D

เกิด Error ครั้งที่ 1 (SQL Server Error: Duplicate entry '2' for key 'PRIMARY')

ไม่เข้าไปทำในรายการที่ 6 ที่ cust\_code = 3 (เนื่องจาก กำหนด MaxErrors = 1 เพื่อพบ Error มากกว่า 1 DatasetProvider ทำการยกเลิก การ Updates และ Rollback)

จะมี Change cache ค้างอยู่ ทั้งหมด 6 รายการ

Test ApplyUpdates(MaxErrors), While MaxErrors = 1

Append 3 records    Save

	cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
	A	A	A	29/6/2018 12:35:02	29/6/2018 12:35:16
1	B	ERR	B	29/6/2018 12:35:02	29/6/2018 12:36:08
	C	C	C	29/6/2018 12:35:02	29/6/2018 12:35:18
	D	D	D	29/6/2018 12:35:09	29/6/2018 12:35:19
2	E	ERR	E	29/6/2018 12:35:09	29/6/2018 12:36:10
▶ 3	F	ERR	F	29/6/2018 12:35:10	29/6/2018 12:36:12

```

--> 29/06/2018 12:28:17.617
insert into 'customers'
('cust_code', 'cust_name', 'cust_tel', 'sys_create_date', 'sys_update_date')
values
(?, ?, ?, ?, ?)

--> 29/06/2018 12:38:08.68
insert into 'customers'
('cust_code', 'cust_name', 'cust_tel', 'sys_create_date', 'sys_update_date')
values
(?, ?, ?, ?, ?)

--> 29/06/2018 12:38:09.69
insert into 'customers'
('cust_code', 'cust_name', 'cust_tel', 'sys_create_date', 'sys_update_date')
values
(?, ?, ?, ?, ?)

--> 29/06/2018 12:38:09.69
insert into 'customers'
('cust_code', 'cust_name', 'cust_tel', 'sys_create_date', 'sys_update_date')
values
(?, ?, ?, ?, ?)

```

```

Param[2] = ลูกค้า 14
Param[3] = 29/6/2018 12:06:20
Param[4] = 29/6/2018 12:07:15

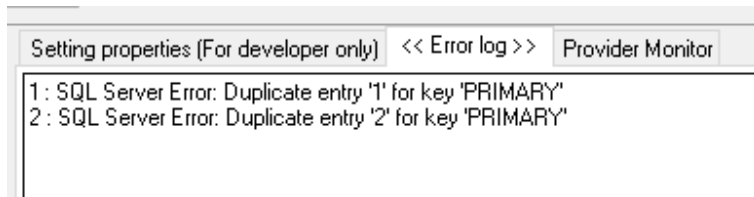
--> 29/06/2018 12:38:08.68
Param[0] = A
Param[1] = A
Param[2] = A
Param[3] = 29/6/2018 12:35:02
Param[4] = 29/6/2018 12:35:16

--> 29/06/2018 12:38:09.69
Param[0] = C
Param[1] = C
Param[2] = C
Param[3] = 29/6/2018 12:35:02
Param[4] = 29/6/2018 12:35:18

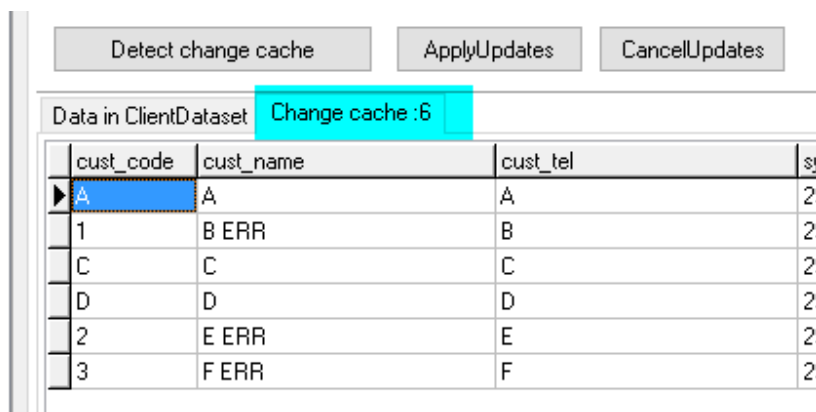
--> 29/06/2018 12:38:09.69
Param[0] = D
Param[1] = D
Param[2] = D
Param[3] = 29/6/2018 12:35:09
Param[4] = 29/6/2018 12:35:19

```

โปรแกรมจะแสดง Error ออกมาแค่ 2 รายการดังภาพ



ภาพด้านล่าง แสดง 6 รายการค้างอยู่ใน Change cache



6 รายการใน Change cache รอให้แก้ไขให้ถูกต้อง แล้ว Updates อีกรอบ หรือ CancelUpdates ในที่นี้ให้กด CancelUpdates ทิ้งไป

### สรุป การใช้ ApplyUpdates(-1)

การใช้งานคำสั่ง Cds.ApplyUpdates(-1); หากพบ Error โปรแกรมจะข้ามไปทำรายการถัดไปเรื่อย ๆ จนกว่าจะหมดข้อมูลที่รอการบันทึก (Change cache) หลังจากนั้นจะทำการ Commit ข้อมูลที่ไม่เกิด Error ลงไปใน Database จริง และจะมีรายการค้างอยู่ใน Change cache ที่จะรอให้แก้ไข หรือรอให้ยกเลิก

**MaxErrors = 0**

เมื่อ กำหนด MaxErrors เท่ากับ 0

เป็นการบอกโปรแกรมของเราว่า เมื่อเราสั่ง บันทึก (Updates) เมื่อ โปรแกรมพบ Error ครั้งที่ 1 โปรแกรมจะยกเลิกการทำงานทันที แม้จะมีรายการที่ค้างอยู่ใน Change cache โปรแกรมก็จะไม่ทำต่ออีกเลย และที่สำคัญโปรแกรมจะไม่ทำการบันทึกข้อมูลใดๆ ลงไปในฐานข้อมูลเลย

**MaxErrors = N**

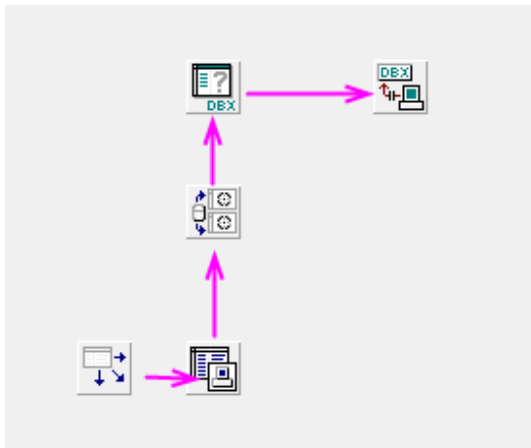
เมื่อกำหนด MaxErrors เท่ากับ N ซึ่งเป็นเป็นตัวเลข ตั้งแต่ 1 ขึ้นไป

เป็นการบอกโปรแกรมของเราว่า เมื่อเราสั่ง บันทึก (Updates) เมื่อ โปรแกรมพบ จำนวน Error มากกว่า N ครั้ง โปรแกรมจะยกเลิกการทำงานทันที แม้จะมีรายการที่ค้างอยู่ใน Change cache โปรแกรมก็จะไม่ทำต่ออีกเลย

ในกรณีที่มี จำนวน Errors มากกว่าจำนวน N โปรแกรมจะไม่ทำการบันทึกข้อมูลใดๆ ลงไปในฐานข้อมูลเลย

เมื่อมี Error แต่ Error ที่เกิดขึ้นน้อยกว่า หรือเท่ากับ N โปรแกรมจะทำการ Commit รายการที่ Updates ผ่านไปลง Database



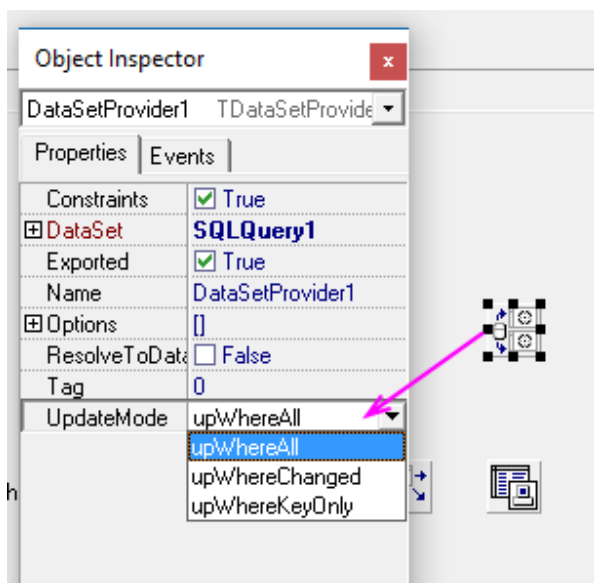


รู้จัก TDateMode ใน TDataSetProvider

TUpdateMode indicates how to locate records so that they can be updated.

ออกแบบมาเพื่อระบุถึงวิธีที่จะค้นหา Record เพื่อที่จะ Updates

TUpdateMode is the type of the UpdateMode property. UpdateMode specifies how the records are located when the ApplyUpdates method is called.



\*คำว่า Updates ในที่นี้หมายถึง


INSERT,

MODIFY, หรือ UPDATE

DELETE

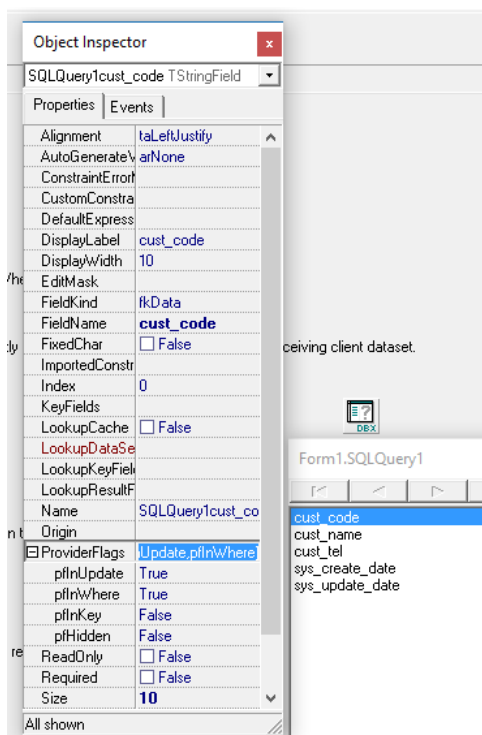
Value	Description
upWhereAll	All columns (fields) are used to locate the record. (This is the Delphi default.)
upWhereChanged	Only key field values and the original value of fields that have changed are used to find the record.
upWhereKeyOnly	Only key fields are used to find the record.

เมื่อเรียกใช้ ApplyUpdates เรื่องที่ต้องรู้ให้ลึกอีกตัว คือ

รู้จัก TDataSetProvider (  ) มี Properties ที่ทำคัญที่จะพูดถึงในบทความนี้ คือ TUpdateMode ที่มันมี 3 ตัวเลือก ในตารางข้างบน

ซึ่งออกแบบมาเพื่อใช้ ตอนที่สั่ง ApplyUpdates หลังจากส่งไปแล้ว DataSetProvider จะ GEN. SQL statement เพื่อที่จะ Updates ข้อมูลไปยัง Database

และ เจ้า DataSetProvider.UpdateMode จะใช้ร่วมกับ TFields.ProviderFlags



ความหมายของ Flag แต่ละตัวใน TFields

Value	Description
pfInUpdate	The field can be included in UPDATE statements. (That is, it can be modified) กำหนดให้ Field นี้อยู่ใน UPDATE SQL ด้วย (หรืออีกนัยหนึ่งคือแก้ไขข้อมูลใน Field นี้ได้)
pfInWhere	The field is included in the WHERE clause when the provider's UpdateMode is upWhereAll or upWhereChanged. กำหนดให้ Field นี้อยู่ใน WHERE SQL ด้วย เมื่อ DatasetProvider มี UpdateModes เป็น upWhereAll หรือ upWhereChanged.
pfInKey	The field is included in the WHERE clause when the provider's UpdateMode is upWhereKeyOnly. กำหนดให้ Field นี้อยู่ใน WHERE SQL ด้วย เมื่อ DatasetProvider มี UpdateModes เป็น upWhereKeyOnly
pfHidden	The field is included in data packets to ensure the uniqueness of records so that they can be correctly updated. It can't be seen or used by the receiving client dataset.

ต่อไปจะทำการทดสอบ โดยในบทความตั้งแต่นี้ไปจะใช้ MaxErrors = 0

TEST 1

DatasProvider.UpdateMode = upWhereAll

**DataSetProvider.UpdateMode**

\*Provider : UpdateMode

upWhereAll

upWhereChanged

upWhereKeyOnly

กำหนดให้ Demo เรียกใช้ UpdateMode : upWhereAll

ภาพในหน้าถัดไป ซึ่งหลังจากที่สั่ง Applyupdate(0) DataSetProvider ทำอะไรหลายอย่างในเบื้องหลัง เช่น

- Gen. INSERT SQL statements
- Mapping Parameters value
- Execute SQL และ ตรวจสอบ Errors
- เมื่อไม่พบ Error ก็ Commit เข้าไปใน Database

Setting properties (For developer only) << Error log >> Provider Monitor

Clear all log

### Debug Provider Generate SQL Command

-> 29/06/2018 15:59:03.63  
 insert into 'customers'  
 ('cust\_code', 'cust\_name', 'cust\_tel', 'sys\_create\_date')  
 values  
 (?, ?, ?, ?)

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

### Debug Provider Generate Params

-> 29/06/2018 15:59:03.63  
 Param[0] = TEST1  
 Param[1] = 测试 INS with upWhereAll  
 Param[2] = Test INS with upWhereAll  
 Param[3] = 29/6/2018 15:59:03

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

Setting properties (For developer only) << Error log >> Provider Monitor

Clear all log

### Debug Provider Generate SQL Command

-> 29/06/2018 15:59:03.63  
 insert into 'customers'  
 ('cust\_code', 'cust\_name', 'cust\_tel', 'sys\_create\_date')  
 values  
 (?, ?, ?, ?)

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

### Debug Provider Generate Params

-> 29/06/2018 15:59:03.63  
 Param[0] = TEST1  
 Param[1] = 测试 INS with upWhereAll  
 Param[2] = Test INS with upWhereAll  
 Param[3] = 29/6/2018 15:59:03

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

Add / Edit Customer.

Cust-code: TEST1

Cust-name: 测试 INS with upWhereAll

Tel: Test INS with upWhereAll

sys create date: Auto.

sys update date:

Save Cancel

Setting properties (For developer only) << Error log >> Provider Monitor

Clear all log

**Debug Provider Generate SQL Command**

29/06/2018 15:59:03.63  
 insert into 'customers'  
 values (cust\_code, cust\_name, cust\_tel, sys\_create\_date)  
 Values  
 (?,?,?,?)

**SQLQuery1.Fields.**

- cust\_code
- cust\_name
- cust\_tel
- sys\_create\_date
- sys\_update\_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**Debug Provider Generate Params**

29/06/2018 15:59:03.63  
 Param(0) = TEST1  
 Param(1) = ทดสอบ INS with upWhereAll  
 Param(2) = Test INS with upWhereAll  
 Param(3) = 29/6/2018 15:59:03

**SQLQuery1.Fields.**

- cust\_code
- cust\_name
- cust\_tel
- sys\_create\_date
- sys\_update\_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

- cust\_code
- cust\_name
- cust\_tel
- sys\_create\_date
- sys\_update\_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

- cust\_code
- cust\_name
- cust\_tel
- sys\_create\_date
- sys\_update\_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

- cust\_code
- cust\_name
- cust\_tel
- sys\_create\_date
- sys\_update\_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

- cust\_code
- cust\_name
- cust\_tel
- sys\_create\_date
- sys\_update\_date

\*ProviderFlags setting.

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

ตรวจสอบใน ฐานข้อมูล MySQL Database

6	ลูกค้า 6	Tel 6	20'
7	ลูกค้า 7	7	20'
8	ลูกค้า 8	8	20'
9	ลูกค้า 9	9	20'
TEST1	ทดสอบ INS with upWhereAll	Test INS with	20'
X	XX	xxx	20'
Y	YY	yyy	20'
Z	ZZ	zzz9	20'

TEST 2 แก้ไขข้อมูลด้วย upWhereAll เปิดข้อมูล cust\_code : TEST 1 ขึ้นมา และแก้ไขข้อมูล จากนั้นกด Save

ตัว DatasetProvider GEN. UPDATE SQL statement และ Commit ข้อมูลได้

**ApplyUpdates(0) Work with upWhereAll**

```

--> 29/06/2018 16:18:22.622
update 'customers' set
'cust_name' = ?,
'cust_tel' = ?,
'sys_update_date' = ?
where
'cust_code' = ? and
'cust_name' = ? and
'cust_tel' = ?
    
```

**DatasetProvider Gen. Update SQL statements ด้วย 3 Where**

```

--> 29/06/2018 16:18:22.622
Param[0] = ลูกค้าทดสอบครั้งที่ 2
Param[1] = เบอร์โทรลูกค้าทดสอบ 2
Param[2] = 29/6/2018 16:18:22
Param[3] = TEST1
Param[4] = ทดสอบ INS with upWhereAll
Param[5] = Test INS with upWhereAll
    
```

**SQLQuery1.Fields.**

- Field: **cust\_code**
- Field: cust\_name
- Field: cust\_tel
- Field: sys\_create\_date
- Field: sys\_update\_date

**\*ProviderFlags setting.**

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

- Field: cust\_code
- Field: **cust\_name**
- Field: cust\_tel
- Field: sys\_create\_date
- Field: sys\_update\_date

**\*ProviderFlags setting.**

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

- Field: cust\_code
- Field: cust\_name
- Field: **cust\_tel**
- Field: sys\_create\_date
- Field: sys\_update\_date

**\*ProviderFlags setting.**

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

**SQLQuery1.Fields.**

- Field: cust\_code
- Field: cust\_name
- Field: cust\_tel
- Field: sys\_create\_date
- Field: **sys\_update\_date**

**\*ProviderFlags setting.**

- pfInUpdate
- pfInWhere
- pfInKey
- pfHidden

TEST 3 ทดสอบลบ ข้อมูลเมื่อ cust\_code : TEST 1

DELETE เมื่อ DatasetProvider.UpdateMode = upWhereAll

**Debug Provider Generate SQL Command** (Clear all log)

```
--> 29/06/2018 16:46:29.629
delete from `customers`
where
`cust_code` = ? and
`cust_name` = ? and
`cust_tel` = ?
```

**Debug Provider Generate Params**

```
--> 29/06/2018 16:46:29.629
Param[0] = TEST1
Param[1] = ลูกค้าทดสอบครั้งที่2
Param[2] = เบอร์โทรศัพท์ทดสอบ 2
```

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

**\*ProviderFlags setting.**

- pflnUpdate
- pflnWhere
- pflnKey
- pflnHidden

TEST 4 DELETE ข้อมูลเมื่อ Cust\_code : X

DELETE เมื่อ DatasetProvider.UpdateMode = upWhereKeyOnly

DELETE เมื่อ cust\_code = X (DatasetProvider GEN SQL statement ค้างภาพ) และ Commit ผ่าน

**Debug Provider Generate SQL Command** (Clear all log)

```
--> 29/06/2018 16:50:16.616
delete from `customers`
where
`cust_code` = ?
```

**Debug Provider Generate Params**

```
--> 29/06/2018 16:50:16.616
Param[0] = X
```

**SQLQuery1.Fields.**

cust_code
cust_name
cust_tel
sys_create_date
sys_update_date

**\*ProviderFlags setting.**

- pflnUpdate
- pflnWhere
- pflnKey
- pflnHidden



## TEST 5

UPATE Cust\_code : 1

โดยเมื่อกำหนดให้ DataSetProvider.UpdateMode = upWhereKeyOnly

จุดสังเกต เมื่อเลือกใช้ upWhereKeyOnly จะต้องระบุ flag pfInKey ด้วยเสมอ

หากไม่กำหนดจะเกิด Error : Unable to find record. No key specified

## TEST 6

UPDATE cust\_code : 9 โดยกำหนด DataSetProvider.UpdateMode = upWhereAll  
และในจังหวะที่เกิด Save (*ApplyUpdates*) มี User อีกคนได้ทำการแก้ไขรายการนี้และทำการ Update ไปก่อนแล้ว (ก่อนหน้าสัก 30 วินาที)

ลองดูว่าจะเกิด Error ว่าอะไรขึ้น

Machine A  
แก้ไขข้อมูล 2 จุด  
- Cust-name  
- Tel

Machine B  
แก้ไข 2 จุด  
- Cust-name  
- Tel

Machine B เปิด Cust\_code : 9 ขึ้นมาเพื่อทำการแก้ไข และกด Save ทุกอย่างผ่านไปด้วยดี หลังจากนั้น Machine A ซึ่งเปิด Cust\_code : 9 ขึ้นมาคั้งไว้ที่หน้าจอ ได้ทำการแก้ไขและกด Save บ้าง เกิดและเกิด ERROR : Record not found or changed by another user

Setting properties (For developer only) << Error log >> Provider.Monitor After updates Provider.Monitor On update error

Debug Provider Generate SQL Command	Machine B	Debug Provider Generate Params
<pre>--&gt; 30/06/2018 10:53:45.645 update `customers` set `cust_name` = ?, `cust_tel` = ?, `sys_update_date` = ? where `cust_code` = ? and `cust_name` = ? and `cust_tel` = ?</pre>	<p>กด Save (ApplyUpdates) ทุกอย่าง OK (Save ได้)</p>	<pre>--&gt; 30/06/2018 10:53:45.645 Param[0] = MESA Param[1] = 085-2815399 Param[2] = 30/6/2018 10:53:45 Param[3] = 9 Param[4] = Customer 9 Param[5] = Tel 9</pre>

stomer Machine A Clear all log

Setting properties (For developer only) << Error log >> Provider.Monitor After updates

1: Record not found or changed by another user Error !!

อธิบายกันยาวหน่อย ทำไมมันถึง Error

“**Record not found** or **changed by another user**”

ข้อหนึ่ง กลับไปมองก่อน เรื่องแรก เราได้เลือก DataSetProvider.UpdateMode = upWhereAll ก็คือจะเอาทุก Field มี Flag pfInWhere ทุกตัวเอามาใส่ใน SQL Statement ในส่วนของ WHERE หากย้อนกลับไปอ่านดูจะเห็นว่า ได้ตั้งให้ field : cust\_code, cust\_name, cust\_tel มี flag เป็น pfInWhere ดังนั้น เมื่อสั่ง ApplyUpdates ตัว DataSetProvider จะ GEN. UPDATE SQL statement

```
update `customers` set
```

```
`cust_name` = ?,
```

```
`cust_tel` = ?,
```

```
`sys_update_date` = ?
```

```
where
```

```
`cust_code` = ? and
```

```
`cust_name` = ? and
```

```
`cust_tel` = ?
```

สังเกตที่ WHERE จะมี 3 Field เข้าไปเองโดยอัตโนมัติ แล้วทำไมมันถึง Error ละ ?

Machine A

Setting properties (For developer only) << Error log >> Provider.Monitor After updates Provider.Monitor On update error

**Debug Provider Generate SQL Command**

```
--> 30/06/2018 10:54:58.658
update `customers` set
`cust_name` = ?,
`cust_tel` = ?,
`sys_update_date` = ?
where
`cust_code` = ? and
`cust_name` = ? and
`cust_tel` = ?
```

**Debug Provider Generate Params**

```
--> 30/06/2018 10:54:58.658
Param[0] = Cheetah
Param[1] = 089-8815388
Param[2] = 30/6/2018 10:54:58
Param[3] = 9
Param[4] = Customer 9
Param[5] = Tel 9
```

ลองแกะ UPDATE SQL statement ออกมาจะได้

```
update `customers` set
`cust_name` = Cheetah,
`cust_tel` = 089-8815388,
`sys_update_date` = 30/6/2018 10:54:58
where
`cust_code` = 9 and
`cust_name` = Customer 9 and
`cust_tel` = Tel 9
```

สังเกตที่ WHERE

มันบอกว่า กำลังจะแก้ไขรายการนะ

โดยไปแก้ไขรายที่มี cust\_code = 9 and cust\_name = Customer 9 and cust\_tel = Tel 9

แต่ใน Database ไม่มีรายการใด ๆ เลยที่ตรงกับ where ข้างต้น

นั่นเพราะว่า Machine B ได้ทำการแก้ไขลูกค้าที่มี cust\_code = 9 and cust\_name = Customer 9 and cust\_tel = Tel 9 ไปก่อนหน้าแล้ว ดังนั้นในฐานะข้อมูลจึงไม่มี ลูกค้าที่มี เลขไขดังกล่าว

cust_code	cust_name	cust_tel	sys_cre
1	cheetah ทดสอบแก้ไข	แก้ไขแล้วนะ	2018-01
10	ซีตาคูสสารfdsvioกำลังแก้ไขนะละ	ซีตาคูแก้ไขfsdf	2018-01
11	ลูกค้า 11	สงฐานแน 10	2018-01
12	ลูกค้า 12	สงฐานแน 12	2018-01
3	Cust name = 3	Tel 3	2018-01
4	New Customer 4	Tel 2	2018-01
5	ลูกค้า 5	Tel 5	2018-01
6	ลูกค้า 6	Tel 6	2018-01
7	ลูกค้า 7	7	2018-01
8	ลูกค้า 8	8	2018-01
9	MESA	085-2815399	2018-01
99	99	99	2018-01

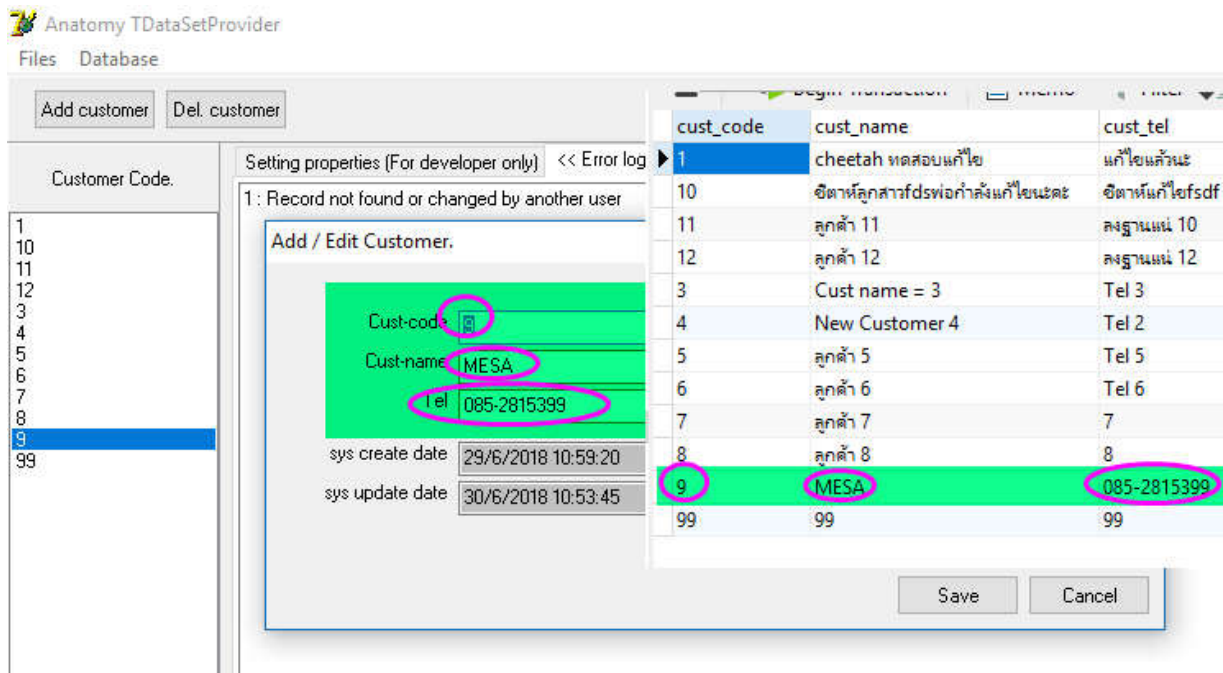
ภาพประกอบ

ลองเปิดดูในฐานข้อมูล ลูกค้าที่มี Cust\_code = 9 ถูกแก้ไขแล้ว และเมื่อ Machine A สั่ง Update โปรแกรมจึงฟ้อง Error ว่า

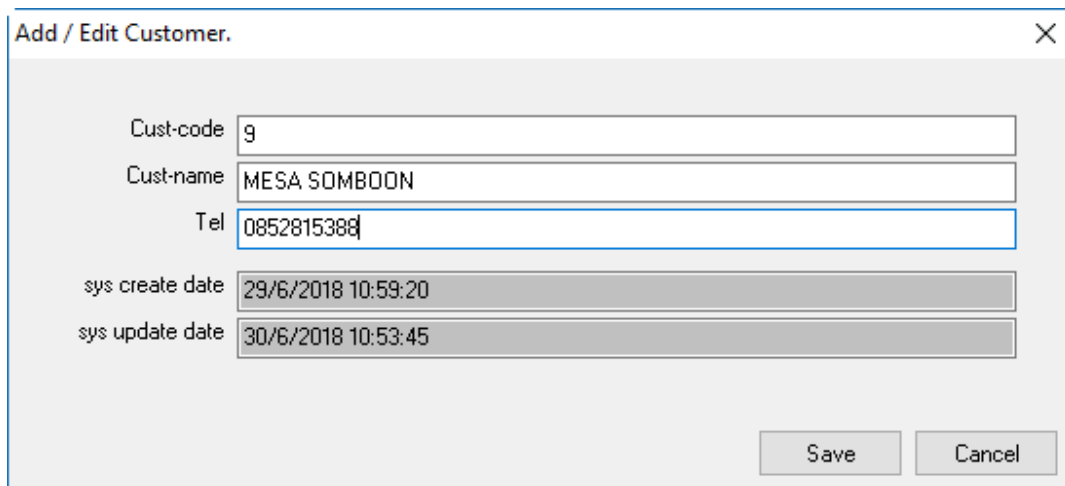
“**Record not found** or **changed by another user**”

หา record ดังกล่าวไม่เจอ ! หรือ Record ดังกล่าวโดนแก้ไขโดยผู้ใช้อื่น

กลับไปโปรแกรม กด CancelUpdates ไปก่อน และจะลองเปิด cust\_code : 9 มาแก้อีกครั้ง คุณก็จะผ่านหรือเปล่า



หลังจาก เปิด Cust\_code : 9 ขึ้นมาอีกรอบ โปรแกรม ก็แสดงข้อมูลชุดใหม่ที่ตรงกันกับใน Database แล้ว ลองแก้ไขดูและ Save แล้วดูผล ? โดยแก้ไขข้อมูลไปเป็นดังภาพ จากนั้นกด Save



หลังกด Save DatasetProvider GEN. UPDATE SQL statement เป็นดังภาพ และไม่มี Error ขึ้น  
มาแล้ว

Debug Provider Generate SQL Command	Debug Provider Generate Params
<pre>--&gt; 30/06/2018 11:30:09.69 update `customers` set `cust_name` = ?, `cust_tel` = ?, `sys_update_date` = ? where `cust_code` = ? and `cust_name` = ? and `cust_tel` = ?</pre>	<pre>--&gt; 30/06/2018 11:30:09.69 Param[0] = MESA SOMBOON Param[1] = 0852815388 Param[2] = 30/6/2018 11:30:09 Param[3] = 9 Param[4] = MESA Param[5] = 085-2815399</pre>

ตรวจสอบเพื่อความชัวร์ ด้วยการเปิดดูในฐานข้อมูล

cust_code	cust_name	cust_tel	sys_create_date	sys_update_date
1	cheetah ทดสอบแก้ไข	แก้ไขแล้วนะ	2018-06-29 10:22:16	2018-06-30 10:38:14
10	ชิตาห์ลูกสาวfdsพอกำลังแก้ไขนะคะ	ชิตาห์แก้ไขfsdf	2018-06-29 12:01:10	2018-06-30 08:15:41
11	ลูกค้า 11	สงฐานแน 10	2018-06-29 12:01:10	2018-06-29 12:02:07
12	ลูกค้า 12	สงฐานแน 12	2018-06-29 12:01:20	2018-06-29 12:02:09
3	Cust name = 3	Tel 3	2018-06-29 10:22:16	(Null)
4	New Customer 4	Tel 2	2018-06-29 10:29:08	2018-06-29 10:29:20
5	ลูกค้า 5	Tel 5	2018-06-29 10:29:08	2018-06-29 10:39:16
6	ลูกค้า 6	Tel 6	2018-06-29 10:29:08	2018-06-29 10:39:21
7	ลูกค้า 7	7	2018-06-29 10:59:20	2018-06-29 10:59:35
8	ลูกค้า 8	8	2018-06-29 10:59:20	2018-06-29 10:59:36
9	MESA SOMBOON	0852815388	2018-06-29 10:59:20	2018-06-30 11:30:09
99	99	99	2018-06-30 09:32:34	(Null)

หากเข้าใจถึงแก่นแท้ของมันแล้ว คิดว่า Delphi Developer คงแก้ไขปัญหา Error ที่ว่า

“**Record not found** or **changed by another user**”

ได้แบบไม่ยากอย่างแน่นอน

สำหรับการทดสอบในรูปแบบอื่น ๆ อยากให้ลองทดสอบด้วยตัวเอง  
สำหรับทุกท่านที่อ่านมาถึงตรงนี้ และสนใจที่จะทดสอบเพื่อให้เกิดความเข้าใจเพิ่มขึ้นท่าน  
สามารถ Download โปรแกรมเพื่อไปทดสอบด้วยตนเองได้ที่ [Anatomy TDataSetProvider](#) สำหรับ  
ใครที่ต้องการ Source code : ให้ Email to : samrids@gmail.com

เอกสารอ้างอิง บางส่วนจากหนังสือ Delphi in Dept: ClientDataset 2<sup>nd</sup> Edition

Applyupdates and MaxErrors

The MaxErrors parameter controls what will happened if errors are encountered during the update process. When MaxErrors is -1 the DataSetProvider attempt to apply all changes in the change cache. After all updates have been attempted, the applied updates are committed by committing the transaction. Any changes that cannot be applied remain in the change cache following the call to ApplyUpdates.

In addition, if you have write and OnUpdateError event handler, it is executed one for each record that could not be applied. If you do not have and OnUpdateError event handler, the DataSetProvider raise and exception for each error encountered. Any executions of OnUpdateError, or raising of exceptions, occur before the transaction is committed.

*Note: if you do not want the user to be confronted with exceptions each time and update fails, create OnUpdateError event handler for the DataSetProvider and place a comment line in the body of that event handler (to prevent Delphi From deleting this empty event hander). Its*



*execution will be a no op (no operation), but nothing will happen, other than the suppression of exception.*

When you call `ApplyUpdates` with a `MaxErrors` value of 0, a single failure causes the `DataSetProvider` to abort the update process and to roll back the transaction, In this case, the change cache is unchanged. Otherwise, all changes were necessarily successfully applied, the transaction was committed, and the change cache is empty.

Again, if `OnUpdateError` event handler has been assigned, it is executed. Otherwise, the `DataSetProvider` raises an exception.

When `MaxErrors` is a positive integer, and the number of errors encountered do not exceed `MaxErrors`, the transaction is committed (all of successful updates are committed), and the unsuccessful changes remain in the change cache. If the number of errors exceeds `MaxErrors` the transaction is rolled back and the change cache remains unchanged.

To some, setting `MaxErrors` to a positive integer, seems counter intuitive. Why would you accept errors? Actually, you are not accepting any errors. Only valid data is being applied to the underlying database. Those errors that are encountered remain in the change cache, where they can be reviewed, discarded, or corrected. If the invalid records can be fixed, those changes can then be applied by calling `ApplyUpdates` again.

The idea behind a positive MaxErrors value is that some number of failures can be tolerated, especially if the change cache includes a large number of updates. At some point, however, too many errors (this is a relative value) is a sign of a more significant problem, in which case the whole update process should be aborted and that problem addressed.

Author: Samrid Somboon

Email: [Samrids@gmail.com](mailto:Samrids@gmail.com)

Line: mesa.d