



PANORAMA X

DEVELOPER CERTIFICATION TRAINING

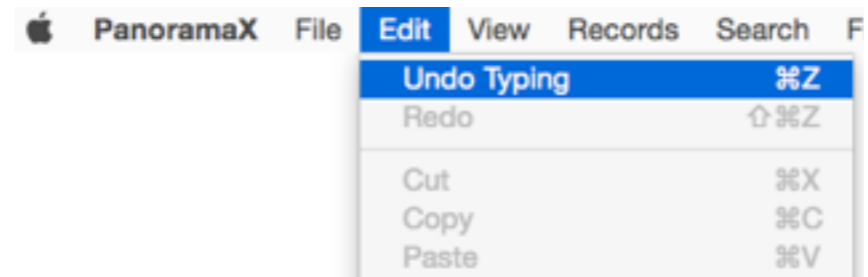
Copyright © 2016 ProVUE Development



Programmed Undo

Copyright © 2015 ProVUE Development

Undo



This is a short, but important, topic.

Undo is a key feature of Panorama X

You can make your procedures "undo-able", and you should.

Undo is how Panorama knows that a database has been modified.

So if you don't set up undo, Panorama might not save your changes (if you don't make any other changes)

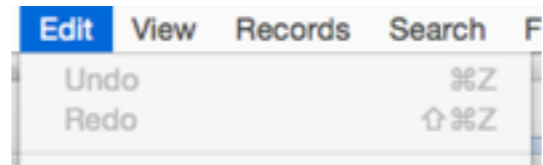
Undo

`Quantity = Quantity + 1`

Suppose your database has a field named `Quantity`.
This simple one line procedure increments the quantity.

Undo

Quantity = Quantity + 1



When you run the procedure, the quantity increases.

But Undo isn't enabled in the Edit menu.

Worse, if you haven't made any other changes to the database, and you close the database -- Panorama won't save the change to the Quantity!!

Panorama isn't aware that the database has changed, so it doesn't automatically save.

Undo

```
Quantity = Quantity + 1  
save
```

One workaround for this is to explicitly save at the end of the procedure.
This is perfectly fine if you don't want the opportunity to Undo.

Undo

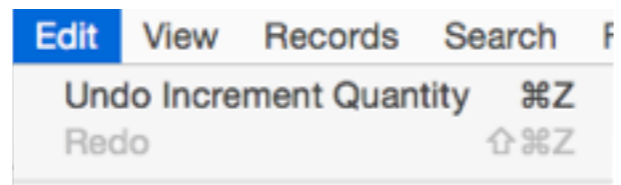
```
startdatabasechange "CurrentRecord","Increment Quantity"  
Quantity = Quantity + 1
```

Usually a better fix is to add Undo support to the procedure.

I've done that here.

Undo

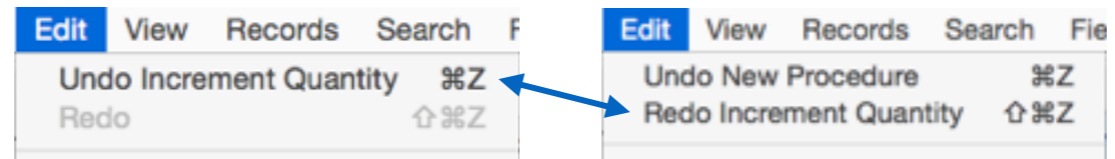
```
startdatabasechange "CurrentRecord","Increment Quantity"  
Quantity = Quantity + 1
```



Now when I run this procedure, the option to Undo appears in the View menu!

Undo

```
startdatabasechange "CurrentRecord","Increment Quantity"  
Quantity = Quantity + 1
```



If I undo, not only is the quantity change undone, but I now have the opportunity to redo!

Pretty good for one line of code!!

DEMO

Undo

startdatabasechange Modification,Title



- **CurrentRecord** – Modifications will be made only to the current record (changing the value of one or more fields).
- **AllRecords** – Modifications will be made to multiple records, and/or the number of records will change, for example adding or removing records, selecting records, filling a field, expanding/collapsing an outline, etc.
- **FieldConfiguration** – Modifications will be made to one or more field definitions, and to data. For example changing a field data type, inserting or deleting fields, changing a fields choice options, etc.
- **FieldProperty** – Modifications will be made to a field definition, but not the data itself, for example changing a field name, an input or output pattern, an input range, formula, etc.
- **FormConfiguration** – A form will be added or removed.
- **ProcedureConfiguration** – A procedure will be added or removed.

The first parameter tells Panorama what we intend to change.

The second parameter tells Panorama what to put in the Undo menu. Don't include the word Undo, Panorama will add Undo/Redo as necessary.

Undo

```
startdatabasechange "CurrentRecord", "Increment Quantity"  
Quantity = Quantity + 1
```

```
startdatabasechange "CurrentRecord", "Reset Quantity"  
Quantity = 1
```

If you use the CurrentRecord option, the procedure shouldn't change anything outside of the current record.

The way this works is that Panorama saves a copy of the current record before you change it. If you later decide to Undo, Panorama restores the record from the copy it saved. ONLY the current record is saved, so if the procedure modifies any other records, the Undo won't work correctly.

Undo

```
startdatabasechange "AllRecords", "New Invoice"  
addrecord  
Quantity = 1
```

```
startdatabasechange "AllRecords", "Sort by City & State"  
field State sortup  
field City sortupwithin
```

```
startdatabasechange "AllRecords", "Recalculate Balance"  
field Balance  
formulafill Credit-Debit  
runningtotal
```

AllRecords is the most common option.

Use this for anything that modifies more than the current record (as long as it doesn't change the field configuration)

Undo

```
startdatabasechange "FieldConfiguration", "Insert Numeric Field"  
insertfield automaticfieldname()  
setfieldproperties initializedictionary("TYPE", "Integer")
```

```
local startField startField = info("fieldname")  
right  
if info("stopped") rtn endif // no more fields  
startdatabasechange "FieldConfiguration", "Delete Fields to Right"  
loop  
  deletefield  
  stoploopif startField = info("fieldname")  
endloop
```

Use this option if you are adding or removing fields, or changing the data type of a field (or changing the choices of a choice field).

If you are also sorting, filling, etc., that's ok, you still use this option.

Undo

```
startdatabasechange "FieldProperty", "Upper Case Field Name"  
setfieldproperties initializedictionary(  
  "NAME", upper(info("fieldname")))
```

```
startdatabasechange "FieldProperty", "Set Numeric Input Only"  
setfieldproperties initializedictionary(  
  "RANGE", "09")
```

Use this option if you are changing a fields properties, but not in a way that will actually change the data itself.

If you are also changing the data, use the FieldConfiguration option.

Undo

```
startdatabasechange "FormConfiguration", "New Mailing Label"  
makenewform "Mailing Label"  
constructmailinglabel  
"Attn: «First» «Last»  
«Address»  
«Suite»  
«City», «State» «Zip»"
```

Use this option if you add or remove an entire form.

Undo

```
addrecord  
Name = "Herb"  
opendatabase "Payroll"  
addrecord  
Name = "Herb"
```



Undo only applies to one database.

If you have a procedure that modifies more than one database, it cannot be undone.

Undo

Graphics

```
startgraphicschange "New Red Star"  
newformobject "StarShapeObject",  
  "rectangle",rectanglesize(100, 396, 72, 72),  
  "color",htmlrgb("FF0000"),  
  "$PolygonCorners","5",  
  "$StarAngle","0°",  
  "$StarDepth","40%",  
  "$Stroke","Solid",  
  "$StarCorners","5",  
  "$Fill","Solid",  
  "LineWidth","1 px"
```

Later you'll learn how to do graphics changes in a program.

When you do, you'll want to know how to undo those changes, so here's how.



Programmed Undo

Copyright © 2015 ProVUE Development