Using Column History to store historical data in memo fields

Last updated: 30 Jan 2019 ***Difficulty level : Moderate***

Starting with **Access 2007**, **ACCDB** files include an **AppendOnly** property for **Long Text/Memo** fields.
This allows you to store a **history of the changes** made to the field.

The history of the **Memo/Long Text** field can later be retrieved using the **ColumnHistory** method, as explained below:

1. *Using Column History*

Create a table with a **long text (memo)** field.
Set its '**Append Only**' property to **Yes** to store the history of all changes to this memo field

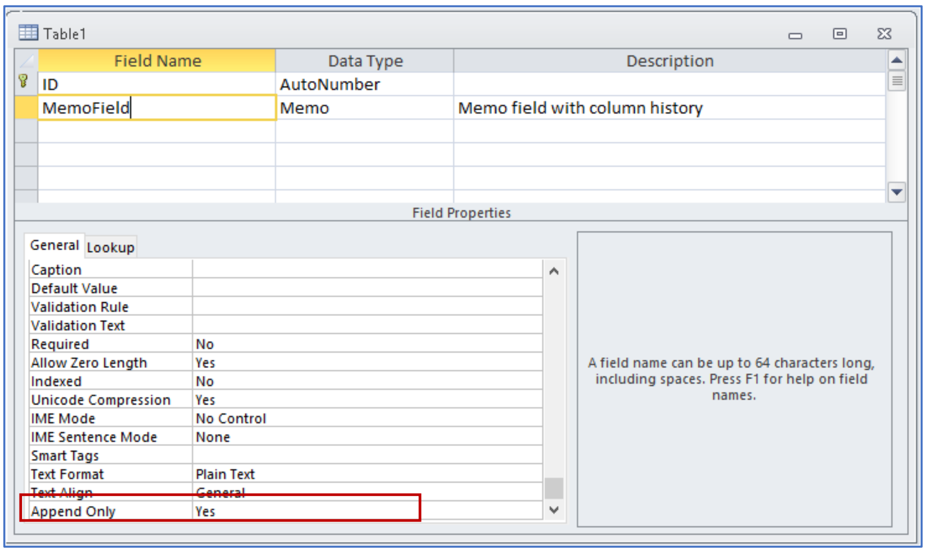


Table1

Field Name	Data Type	Description
ID	AutoNumber	
MemoField	Memo	Memo field with column history

Field Properties

General | Lookup

Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	
Text Format	Plain Text
Text Align	General
Append Only	Yes

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Enter some data in the field then edit one or more of the records

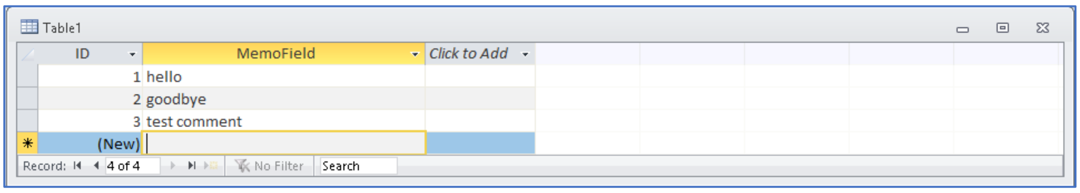


Table1

ID	MemoField	Click to Add
1	hello	
2	goodbye	
3	test comment	
(New)		

Record: 4 of 4 No Filter Search

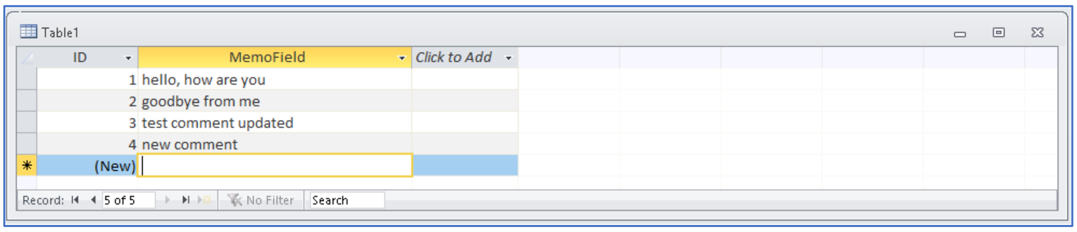
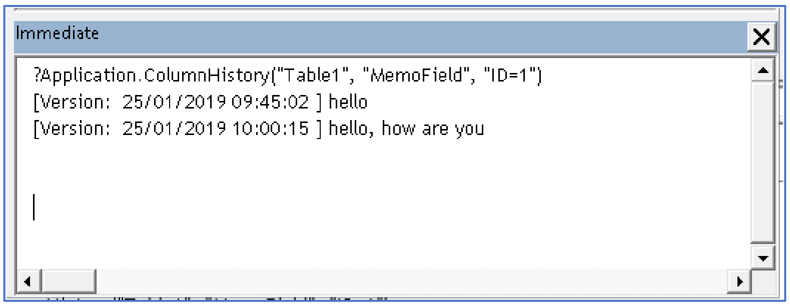


Table1

ID	MemoField	Click to Add
1	hello, how are you	
2	goodbye from me	
3	test comment updated	
4	new comment	
(New)		

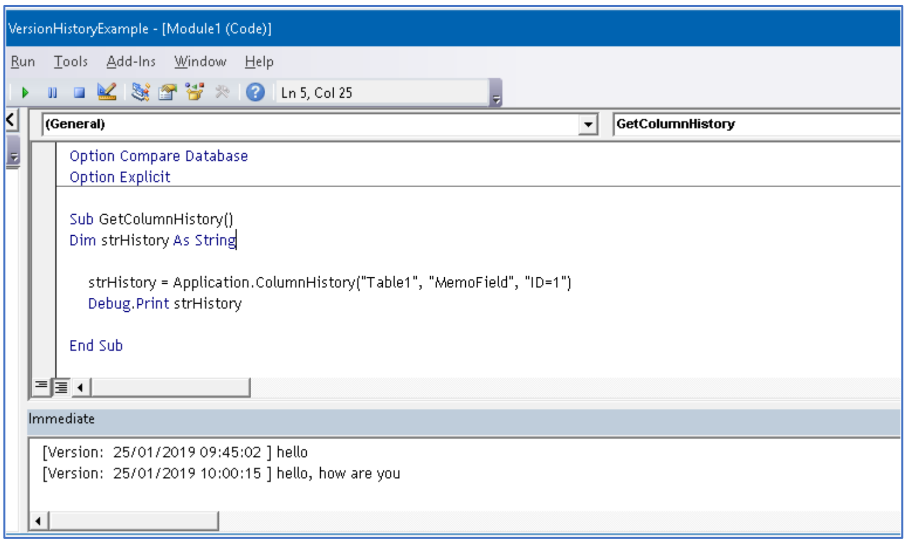
Record: 5 of 5 No Filter Search

You can view a history of an individual record in various ways
For example, by typing this in the **VBE Immediate** window:
?Application.ColumnHistory("Table1", "MemoField", "ID=1")



```
?Application.ColumnHistory("Table1", "MemoField", "ID=1")
[Version:  25/01/2019 09:45:02 ] hello
[Version:  25/01/2019 10:00:15 ] hello, how are you
```

Or you can create a procedure to do this:

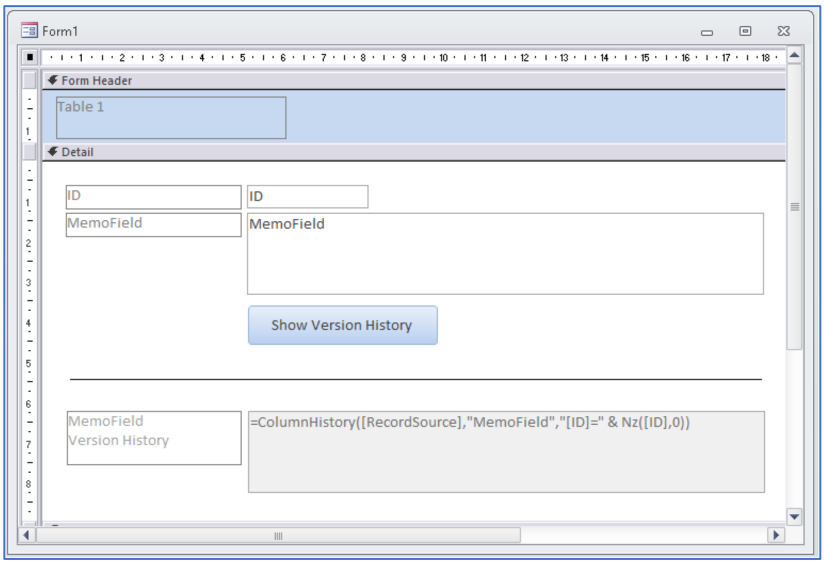

VersionHistoryExample - [Module1 (Code)]
Run Tools Add-Ins Window Help
Ln 5, Col 25
(General)
GetColumnHistory
Immediate

```
Option Compare Database
Option Explicit

Sub GetColumnHistory()
Dim strHistory As String

    strHistory = Application.ColumnHistory("Table1", "MemoField", "ID=1")
    Debug.Print strHistory

End Sub
```

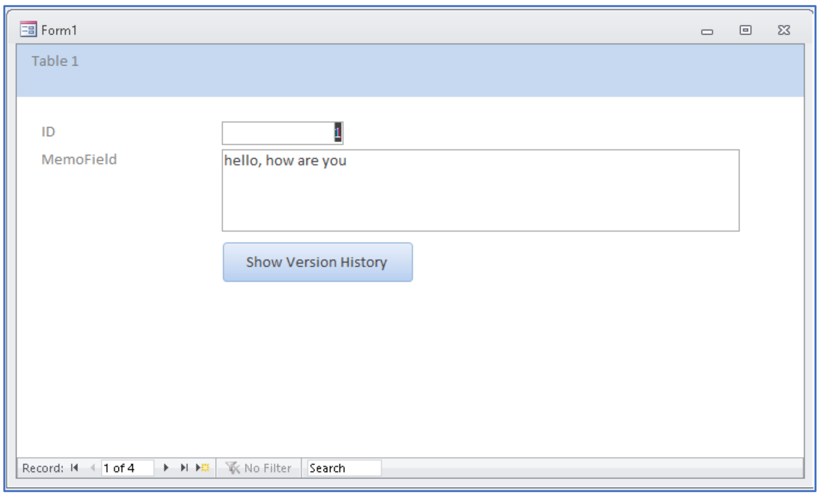
```
[Version:  25/01/2019 09:45:02 ] hello
[Version:  25/01/2019 10:00:15 ] hello, how are you
```

However, to make the feature have any real value, you can include the column history on a form.
In this case I have added an **extra form control** with **control source**:
=ColumnHistory([RecordSource],"MemoField","[ID]=" & Nz([ID],0))

The control is hidden and disabled (as it cannot be edited by end users)


Form1
Form Header
Table 1
Detail
ID
ID
MemoField
MemoField
Show Version History
MemoField Version History
=ColumnHistory([RecordSource],"MemoField","[ID]=" & Nz([ID],0))

The **default view** is:

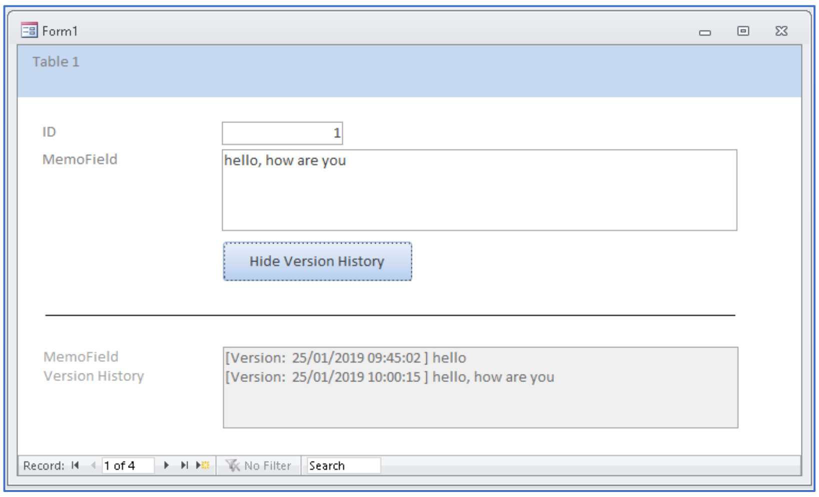


Form1

Table 1

ID	
MemoField	hello, how are you

After clicking the button, the **column history** is shown . . . **but it CANNOT be edited by end users**



Form1

Table 1

ID	1
MemoField	hello, how are you

MemoField Version History	[Version: 25/01/2019 09:45:02] hello [Version: 25/01/2019 10:00:15] hello, how are you

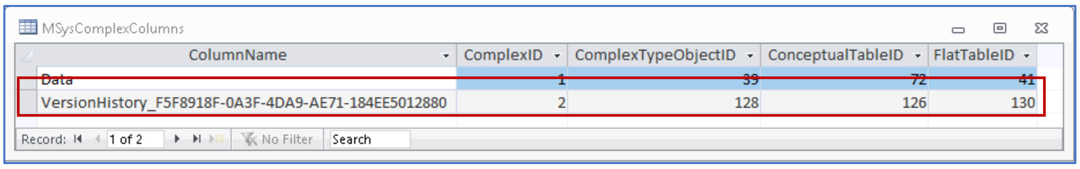
2. *How the ColumnHistory property works*

IMPORTANT:
The information in this section related to various system tables which are used by Access to make databases function correctly

Some system tables can be viewed & a few can be edited
But that doesn't mean you should do so ....UNLESS YOU ARE ABSOLUTELY SURE WHAT YOU ARE DOING
Altering one table may have 'knock on' effects on other tables

Incorrectly editing system tables may corrupt your database or prevent you opening it

When the **column history** property was specified **(by setting 'Append Only' to Yes)**, a new record was added to the system table ***MSysComplexColumns***.
NOTE: Set ***Show System Objects = Yes*** in **Navigation Options** to view this table



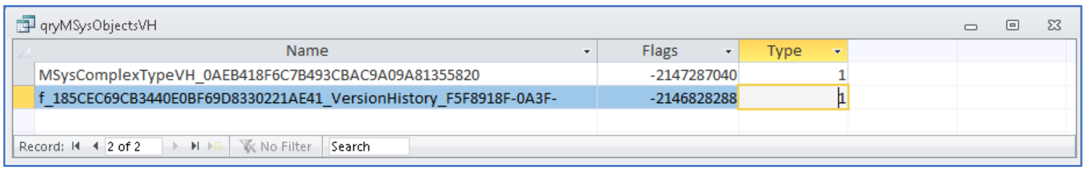
MSysComplexColumns

ColumnName	ComplexID	ComplexTypeObjectID	ConceptualTableID	FlatTableID
Data	1	39	72	41
VersionHistory_F5F8918F-0A3F-4DA9-AE71-184EE5012880	2	128	126	130

This record indicates new system tables have been created.

Two new ***system tables*** are created for each table with a memo field where the ‘Append Only’ property = Yes
These tables are ‘***deep hidden’*** and do not appear in the ***navigation pane*** list
However, the new system table names can be identified by creating a query on the ***MSysObjects*** table

SELECT MSysObjects.Name, MSysObjects.Flags, MSysObjects.Type
FROM MSysObjects
*WHERE (((MSysObjects.Name) Like "*VH*" Or (MSysObjects.Name) Like "*Version*") AND ((MSysObjects.Flags)<>1) AND ((MSysObjects.Type)=1));*


qryMSysObjectsVH

Name	Flags	Type
MSysComplexTypeVH_0AEB418F6C7B493CBAC9A09A81355820	-2147287040	1
f_185CEC69CB3440E0BF69D8330221AE41_VersionHistory_F5F8918F-0A3F-	-2146828288	1

Record: 2 of 2 No Filter Search

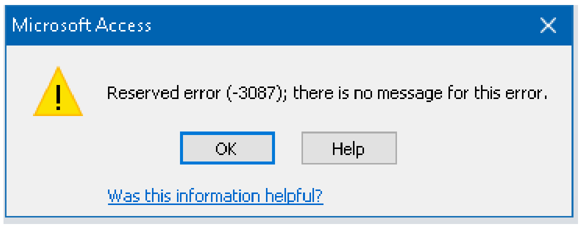
NOTE:
There is some unfortunate confusion in nomenclature for this feature.
To enable it, you set ***Append Only = Yes***.
However, the property is called ***ColumnHistory*** and the system tables refer to ***VersionHistory***

As previously stated, system tables are required to ensure Access works correctly.
All are hidden. Most of them cannot be edited for security reasons.

However, by using **deep hidden tables**, Access makes it difficult for us to view the contents of these tables

If we create this query on the first table listed above, a reserved error occurs

*SELECT * FROM MSysComplexTypeVH_0AEB418F6C7B493CBAC9A09A81355820;*

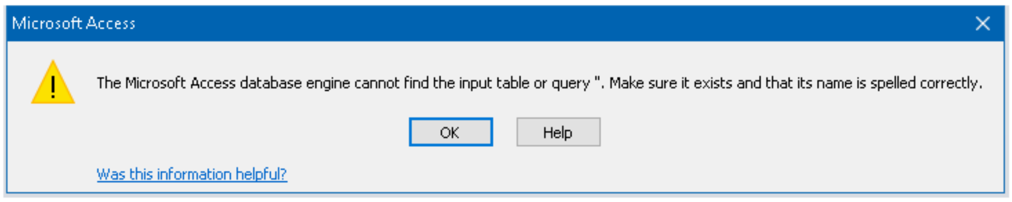

Microsoft Access

Reserved error (-3087); there is no message for this error.

OK Help

Was this information helpful?

Trying to view the other table gives a different error.
Square [] brackets are needed due to the table name ending in ‘-‘
*SELECT * FROM [f_185CEC69CB3440E0BF69D8330221AE41_VersionHistory_F5F8918F-0A3F-];*


Microsoft Access

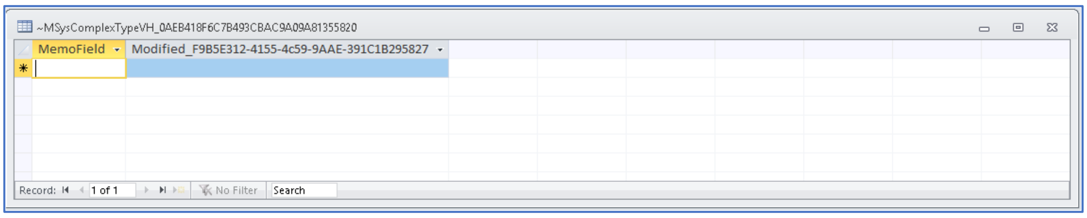
The Microsoft Access database engine cannot find the input table or query ''. Make sure it exists and that its name is spelled correctly.

OK Help

Was this information helpful?

To view the contents of these tables, we need to use a bit of trickery.
I am deliberately not going to explain how I achieve this in this article

Surprisingly the first of these tables is empty:

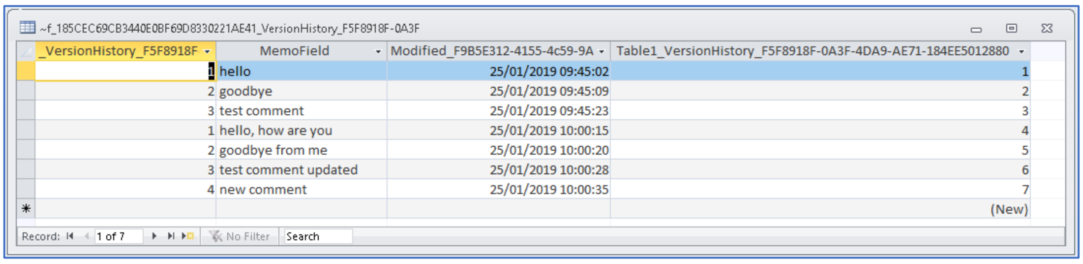


~MSysComplexTypeVH_0AEB418F6C7B493CBAC9A09A81355820

MemoField	Modified_F9B5E312-4155-4c59-9AAE-391C1B295827
*	

Record: 1 of 1 No Filter Search

The **version history data** is stored in the second table:

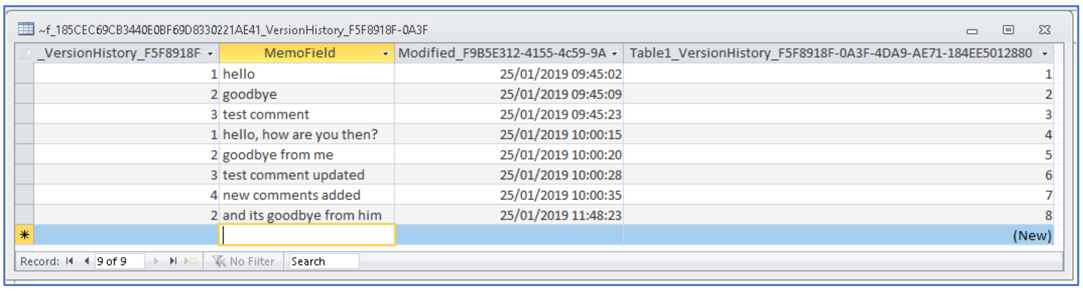


~f_185CEC69CB3440E0BF69D8330221AE41_VersionHistory_F5F8918F-0A3F

_VersionHistory_F5F8918F	MemoField	Modified_F9B5E312-4155-4c59-9A	Table1_VersionHistory_F5F8918F-0A3F-4DA9-AE71-184EE5012880
1	hello	25/01/2019 09:45:02	1
2	goodbye	25/01/2019 09:45:09	2
3	test comment	25/01/2019 09:45:23	3
1	hello, how are you	25/01/2019 10:00:15	4
2	goodbye from me	25/01/2019 10:00:20	5
3	test comment updated	25/01/2019 10:00:28	6
4	new comment	25/01/2019 10:00:35	7
			(New)

Record: 1 of 7 No Filter Search

Although Access has made it very difficult to view this table, once it is visible, it can in fact be edited.
For example, I have edited 2 records and added a new record.
The screenshot shows the table being edited

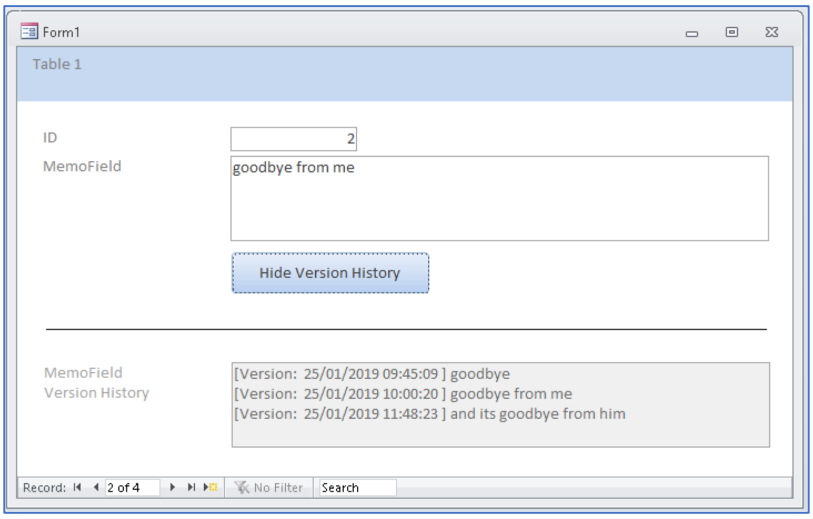


~f_185CEC69CB3440E0BF69D8330221AE41_VersionHistory_F5F8918F-0A3F

_VersionHistory_F5F8918F	MemoField	Modified_F9B5E312-4155-4c59-9A	Table1_VersionHistory_F5F8918F-0A3F-4DA9-AE71-184EE5012880
1	hello	25/01/2019 09:45:02	1
2	goodbye	25/01/2019 09:45:09	2
3	test comment	25/01/2019 09:45:23	3
1	hello, how are you then?	25/01/2019 10:00:15	4
2	goodbye from me	25/01/2019 10:00:20	5
3	test comment updated	25/01/2019 10:00:28	6
4	new comments added	25/01/2019 10:00:35	7
2	and its goodbye from him	25/01/2019 11:48:23	8
			(New)

Record: 9 of 9 No Filter Search

Any changes made to the column history are then shown in the form we created earlier



Form1

Table 1

ID	2
MemoField	goodbye from me

MemoField Version History

[Version: 25/01/2019 09:45:09] goodbye
[Version: 25/01/2019 10:00:20] goodbye from me
[Version: 25/01/2019 11:48:23] and its goodbye from him

Record: 2 of 4 No Filter Search

It should be emphasised that **editing the column history** directly in the system table does **NOT update** the **original table**.

Even so, in this case, the ability to edit the table does make sense, as it means someone who knows how to do so can delete inappropriate entries from the column history.

However, none of the above is documented anywhere.

As far as I am aware, very few people know how to view the contents of the deep hidden system tables. I found out how to do so mainly by trial and error.

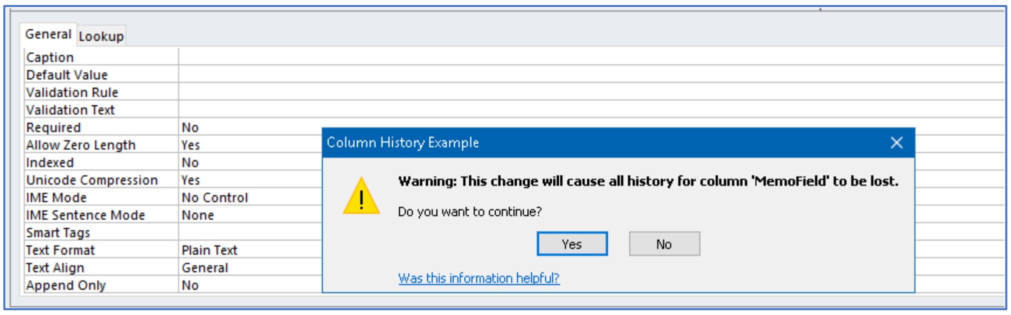
3. *Editing Column History*

The **ColumnHistory** property is only intended for situations where **historical data should be retained without changes.**
If it is **ABSOLUTELY necessary** to **edit the column history**, there are **3 possible approaches/workrounds**:

a) Remove all Column History for an individual record
Copy the record then delete the original record. The **ColumnHistory** is NOT transferred

Add 3 images here

b) Remove/replace Column History for all records
Set the **Append Only** property to **No**. You will be warned that the **ColumnHistory** will be deleted.
If no other field depends on the associated system table, it will be deleted automatically



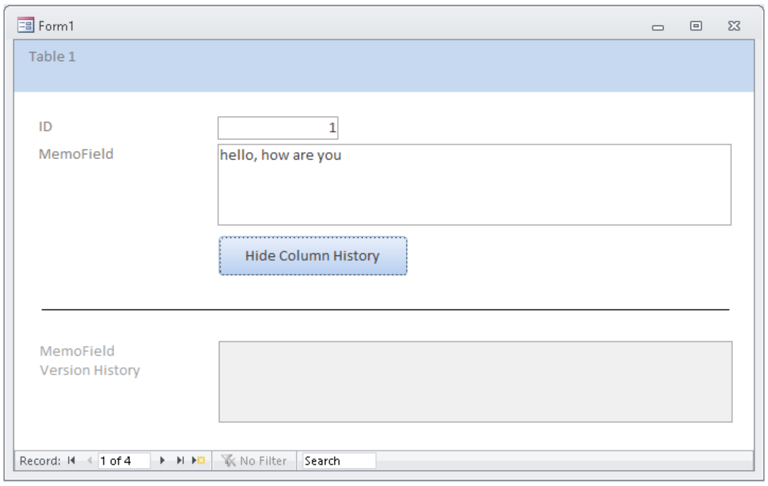
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	
Text Format	Plain Text
Text Align	General
Append Only	No

Column History Example

Warning: This change will cause all history for column 'MemoField' to be lost.

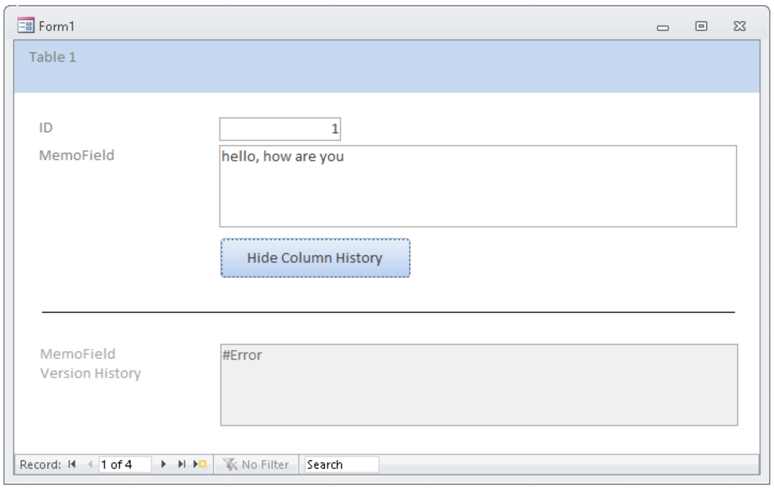
Do you want to continue?

If you wish, you can then reset the **Append Only** property to **Yes**.
The **system table will be re-created automatically with no data**



ID	1
MemoField	hello, how are you
MemoField Version History	

Alternatively, leave **Append Only = No** to switch the feature off **permanently** for that field
The form control shows **#Error** as the system table no longer exists



c) Delete an individual item in the column history for one or more records
This is the most flexible and powerful solution and was described in **section 2** above.
However, to do this requires knowledge of how to view the **deep hidden system table** used to store the **ColumnHistory**.

How this is done is not documented by Microsoft and has deliberately not been explained in this article

As has been demonstrated, each of these methods of editing the **ColumnHistory** are far from ideal.
My strong advice would be to avoid using this method to retain historical data if editing is ever anticipated.

Instead create a **separate table** with the memo field and link it to the main table using a **one to many** join.

If required, editing of previous records by end users can easily be prevented by locking the memo field control at form level
However, the field could still be edited by system admins in the table itself when necessary

4. Upsizing to SQL Server

If, at any stage, you decide to upsize your datafile to **SQL Server**, you need to be aware that **the column history data** cannot be migrated.
If you no longer need the historical data, you should just switch off the feature before upsizing.

However, if you wish to **retain that historical data**, we can recover it for you as a **standard Access table**.
Please note that this is a chargeable service charged at £60 GBP per hour.
If you only have one column history memo field to convert, it is unlikely to be more than 1 hour's work.

Similar conditions apply as for the database conversion feature but, in this case, **ACCDB/ACCDE/MDB/MDE** file types are all acceptable .

For further details of this **recovery service**, please email info@mendipdatasystems.co.uk with details of your file(s)

5. Downloads

Click to download:

The example database:	Column History Example	(ACCDB file - zipped)
PDF version of this article:	Using Column History in memo fields	(PDF)

6. *Further Reading*

For further information on this topic, see:

http://www.fmsinc.com/MicrosoftAccess/2007/ColumnHistory/Index.asp

https://sourcedaddy.com/ms-access/append-only-fields.html

I would be grateful for any feedback on this article including details of any errors or omissions

Colin Riddington	***Mendip Data Systems***	***Last Updated 30 Jan 2019***