SQL-proofing* Your Access Application

Peter Bryant

Corylus Business Systems

pbryant@corylus-business.co.uk

*Well some issues that are worth considering. This is not a definite list!

My background

- Started in mainframe/mini development (punched cards and green screens) – COBOL, Algol68R, Fortran
- Mid 80's - came the PC into business
- Used Windows since v2.x and Access developer since the 1st beta (this does not, necessarily, make me an expert!)
- 20 years of IT Management dabbling in code for internal needs
- Now self-employed, I have three development areas:
    - Turn data into information - long term (pseudo-)agile/prototyping projects
    - Build new systems to replace rubbish or outmoded system
    - Take over maintenance
- That's my excuse list finished ☺

Agenda

- Hello My name's Peter and I'm an Accessoholic.
- I haven't touched Access since 8:45 this morning...
- Agenda proper
- Part 1 – rather random lessons learned from some recent SQL migration projects
- Part 2 – DSN-Less connection of Access FE’s to SQL
- Part 3 – open forum: further ideas & issues

Part 1 – Random issues

- This is not a definitive list, these are lessons learned from some recent SQL migration projects
- It's all (mostly) about good enough, not the best possible at all costs… so don’t judge!

Indexes

- Don't use index as identity - ever.
  - Because (AFAIK) of partitioned SQL design options SQL now assigns group of identity values and can miss out whole chunks of consecutive numbers. It’s a bit like RID’s in Active Directory*
  - This can make your VAT Invoice Number illegal.
- Roll your own
  - But make it a unique key as well

*Did I just lose the room?

Garbage collection

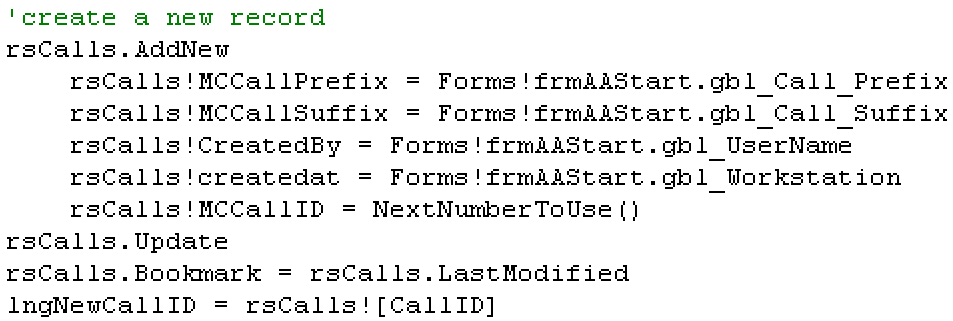
- Whilst I have no definitive evidence… it just feels good to do it this way. Do your own garbage collection properly
- rst.close followed by db.close; followed by
- Set rst = nothing and Set db= nothing
- Generally I declare objects within the lowest level (ie at sub or procedure level) and create/destroy as I go.
- Only very occasionally will I have module level or higher objects
- We had one project which looped round opening other accdb's, a dev (who shall remain nameless!) forgot to destroy the object created. After a few 100 objects, the inevitable happened ☺

Record locking

- Stop people getting into a record if someone else is using it.
- Have a couple of lock fields – user and workstation.
- Have 3 functions
  - Is<tbl>RecordLocked – check, and tell the user who is using the record
  - Lock<tbl>Record – fill in the fields
  - Unlock<tbl>Record – clear the fields
- With SQL Server you can use system level values for the values – or you can use standard windows API call to get the user and workstation

Bookmarks

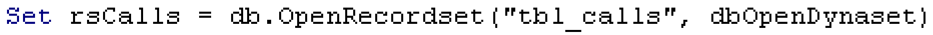
- In access after adding a record you can inspect the new record id straight after .AddNew before .Update, just by referencing the index column
- In SQL Server, the key to the record is not available until after the .Update write, so use the LastModified/Bookmark method



```
'create a new record
rsCalls.AddNew
    rsCalls!MCCallPrefix = Forms!frmAAStart.gbl_Call_Prefix
    rsCalls!MCCallSuffix = Forms!frmAAStart.gbl_Call_Suffix
    rsCalls!CreatedBy = Forms!frmAAStart.gbl_UserName
    rsCalls!createdat = Forms!frmAAStart.gbl_Workstation
    rsCalls!MCCallID = NextNumberToUse()
rsCalls.Update
rsCalls.Bookmark = rsCalls.LastModified
lngNewCallID = rsCalls![CallID]
```

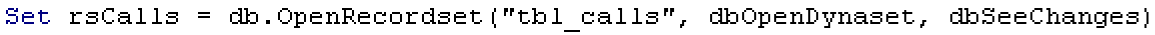
dbSeeChanges

- In Access you can be very lazy and just open a recordset by naming it.

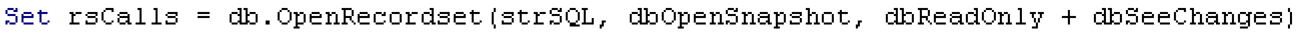


```
Set rsCalls = db.OpenRecordset("tbl_calls", dbOpenDynaset)
```

- With a SQL Backend you must use dbSeeChanges for any table with an identity column (which should be all your tables!)



```
Set rsCalls = db.OpenRecordset("tbl_calls", dbOpenDynaset, dbSeeChanges)
```



```
Set rsCalls = db.OpenRecordset(strSQL, dbOpenSnapshot, dbReadOnly + dbSeeChanges)
```

T-SQL

- This is a personal preference.
- I don’t like SQL code in VBA, it’s harder to read, harder to change, and you cannot see a diagram and it may have unexpected consequences.
- For any SQL statement that is used in code, my preference is for an Access Query, or SQL View which does the same thing, and can be easily linked.
- It means your Access Query will be upsized to SQL
- It becomes really useful if you use parameters – push the parameters to the backend, run the query from there and get better performance.

SELECT INTO

- Corollary
- If you like to use SELECT…INTO code to select records and then insert them into the database, you will get a lot of I/O
- Instead, write a Stored Procedure in SQL to do this for you, it will all run in the backend and perform much better.

Views and tracking their indexes

- For views to be updateable, they must have an identified index. This can be set when linking to the back end the first time.
- MUST handles this rather wonderfully.
- However, on relinking – you must preserve this information or else the views will become read only.
- Check out Andy’s code at http://www.ascassociates.biz/relinking-tables-and-views-between-access-and-sql-server/ - it just works.

Continuous Forms

- When you create a continuous form in the designer and link it to a table or view the recordset is set to be a dynaset by default.
- When the record set is large, this performs very slowly.
- Recommendation:
  - Create the continuous form
  - Change the form to snapshot
  - Add a refresh button (if needed – IME not normally necessary)
  - If updates are required to a record – do that as a popup form
  - Refresh the form on closing the popup
- This moved one form from half a minute or more to load and scroll, to nano-seconds.

Filtering views at the backend

- Filtering can be done in Access or SQL.
- If you do in Access then the data I/O over the LAN can impact performance badly
- Andy’s MUST – Parameters table has a per person view of parameters
- In use – push your parameters to the backend
- Modify the SQL View to use the parameters view (the personal one) to select the records
- Use this as your record source.

Custom database security 1

- Permissions to the database are generally user has full read write, it is rare for an SME to get into very granular security
- Users can use an ODBC connection to read AND write.
- This is, officially, bad.
- Telling people they will be sacked for it, will not recover your data
- With DPA/GDPR sensitive data, this could be serious
- Ah, but SQL has sp_setapprole to connect as a role, this means users can have no access, but the role has everything
- But it doesn’t work from VBA ☹

Custom database security 2

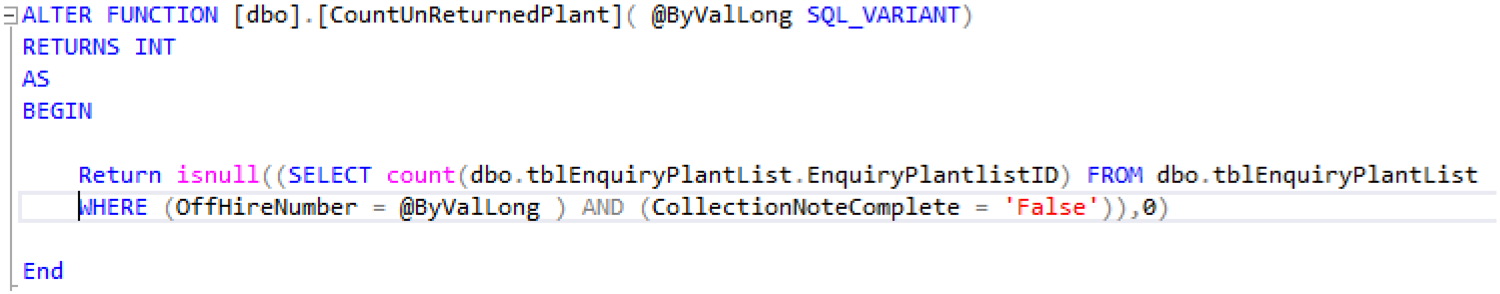
- Programmatically connect to the database with a SQL user/pwd combination (which is a programmed Role) instead of using the user credentials in a trusted connection.
- This leaves the password in code (or potentially in a front end table), but that is more manageable and secure
- Revoke domain users from db permissions (other than being enough to “see” and then start a connection to the database).

Custom database security 3

- If you use the backend parameters table based on the connection user id, then this will fail.
- So instead rework the filtering on the parameter table from the SQL setup:
  - (substring(suser_sname(),charindex('\',suser_sname())+(1),len(suser_sname())))
- To
  - (host_name())
  - NB Terminal Server will break this.

Functions

- If you write functions in Access that are used in a query, these will not be available when the query is converted to a SQL View
- So don’t do it.
- If you do, then work out how to do them as Scalar Functions in T-SQL, they look like this:



```
ALTER FUNCTION [dbo].[CountUnReturnedPlant]( @ByValLong SQL_VARIANT)
RETURNS INT
AS
BEGIN

    Return isnull((SELECT count(dbo.tblEnquiryPlantList.EnquiryPlantlistID) FROM dbo.tblEnquiryPlantList
    WHERE (OffHireNumber = @ByValLong ) AND (CollectionNoteComplete = 'False')),0)

End
```

MUST

- Beg ~~borrow~~** or ~~steal~~* a copy of MUST!
- Or
- Use SMSS migration techniques to get your tables and views into the SQL server

- MUST is amazing, and I wish Andy would keep it available. Whilst I do a lot of post-MUST work, it does so much heavy lifting for you

*This is not allowed
**This may not be allowed

dbo.format

- If you use the format function in Access Queries it does not translate well to SQL views, as the format function there is for dates
- We had a lot of Format function calls that put an integer into a 6 digit string
- This had to be turned into something like:
    - right('000000' + cast(TC.MCCallID as varchar),6)
- So watch out, and learn CAST

Booleans

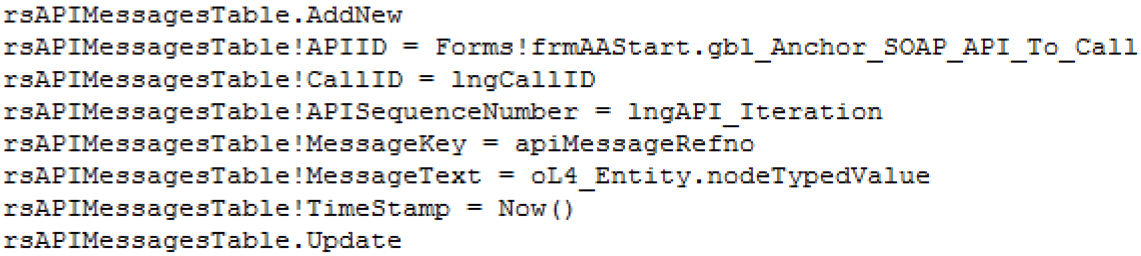
- You should never use explicit 0 and 1 for Booleans anyway

- Make sure you always do logical tests with = True or = False, ***or***
- (after some discussion) Make sure all tests result in valid Boolean values. N.B. True does not equal Not False*

*What did you do in the Boolean Wars Daddy?

v2016, DAO and .AddNew

- This is currently unconfirmed as a 2016 bug, but… heads up
- .AddNew reported as running slow in code like this:



```
rsAPIMessagesTable.AddNew
rsAPIMessagesTable!APIID = Forms!frmAAStart.gbl_Anchor_SOAP_API_To_Call
rsAPIMessagesTable!CallID = lngCallID
rsAPIMessagesTable!APISequenceNumber = lngAPI_Iteration
rsAPIMessagesTable!MessageKey = apiMessageRefno
rsAPIMessagesTable!MessageText = oL4_Entity.nodeTypedValue
rsAPIMessagesTable!TimeStamp = Now()
rsAPIMessagesTable.Update
```

- On stepping through we are observing a 5-10 second delays on the .AddNew line
- Considering ADO and an INSERT SQL (I know, I don’t like them in code!) for diagnostic purposes.

Links

Check out the following

- Optimizing Microsoft Office Access Applications Linked to SQL Server: https://technet.microsoft.com/en-us/library/bb188204(v=sql.90).aspx
- Microsoft Access Performance Tips to Speed up Your Access Databases: http://www.fmsinc.com/MicrosoftAccess/Performance.html

Part 2 – Database connections

- DSN-Less connections Access FE’s to SQL with DSN-less code
  - Allowing the user the ability to pick the right backend database.
  - Allowing the application to respond to being run on different networks by automatically switching backend server at start-up.

Demo

- No presentation is worth much if there’s not a demo of running code that fails.
- So here goes…

Part 3 – Open Forum

Help for Heroes

- I’m doing my 7th bike ride – 350 miles (the long way) from Lille to Arnhem in September.
- If you would like to donate – http://bit.ly/pjb17!

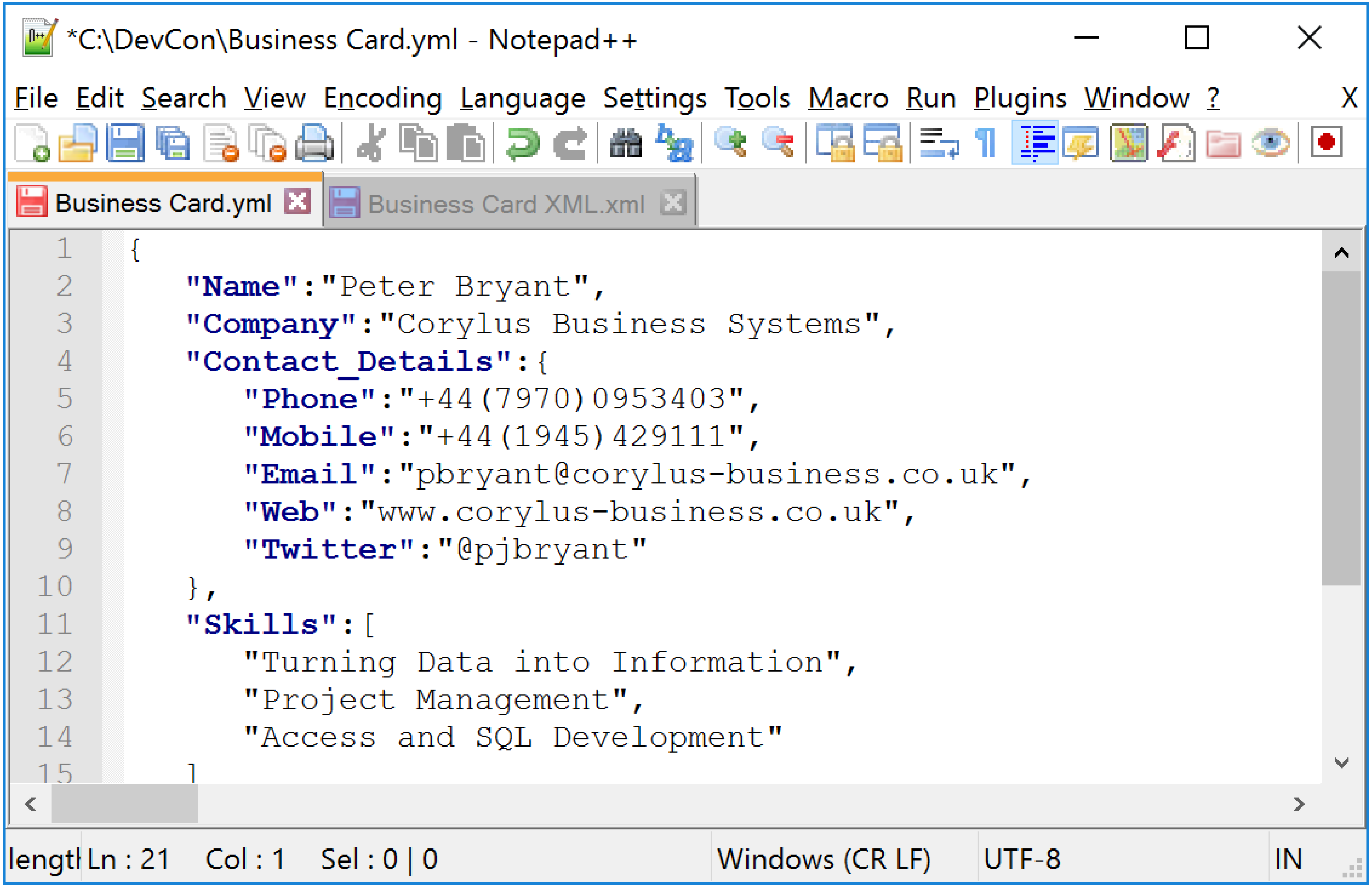
SQL-proofing* Your Access Application

Peter Bryant

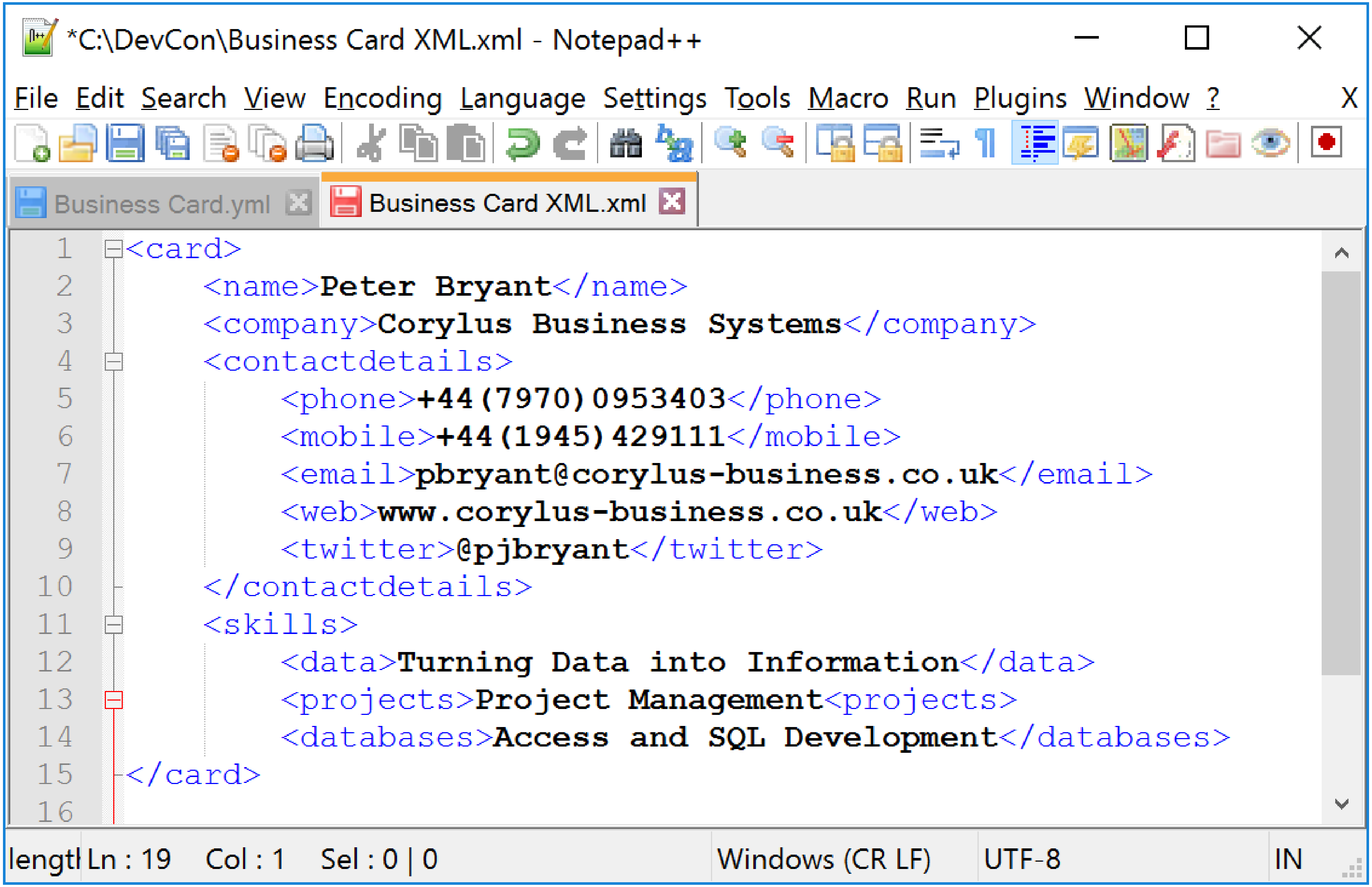
Corylus Business Systems

pbryant@corylus-business.co.uk

*Well some issues that I hope were worth considering.



```
{
    "Name":"Peter Bryant",
    "Company":"Corylus Business Systems",
    "Contact_Details":{
        "Phone":"+44(7970)0953403",
        "Mobile":"+44(1945)429111",
        "Email":"pbryant@corylus-business.co.uk",
        "Web":"www.corylus-business.co.uk",
        "Twitter":"@pjbryant"
    },
    "Skills":[
        "Turning Data into Information",
        "Project Management",
        "Access and SQL Development"
    ]
```


*C:\DevCon\Business Card XML.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X
Business Card.yml
Business Card XML.xml

```
<card>
    <name>Peter Bryant</name>
    <company>Corylus Business Systems</company>
    <contactdetails>
        <phone>+44(7970)0953403</phone>
        <mobile>+44(1945)429111</mobile>
        <email>pbryant@corylus-business.co.uk</email>
        <web>www.corylus-business.co.uk</web>
        <twitter>@pjbryant</twitter>
    </contactdetails>
    <skills>
        <data>Turning Data into Information</data>
        <projects>Project Management<projects>
        <databases>Access and SQL Development</databases>
</card>
```

lengtl Ln : 19 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
IN

OUT
TO LUNCH