

How to change the UnitronUC32 system Data Storage platform to MSSQL Server



By default Datalog and Alarm data from a UnitronUC32 site is stored in Microsoft Access format. However, it can be desirable to change this to Microsoft SQL server under some circumstances.

This document describes how to decide whether or not to change the platform, and how to carry out the change.

Choosing a Data Storage platform

There are two main platforms that can be user for Data Storage in the UnitronUC32 system:

- **MS Access Database** – this method of storage does not require maintenance of a database server, but only works with relatively small amounts of data. This is the default for newly-installed UnitronUC32 systems.

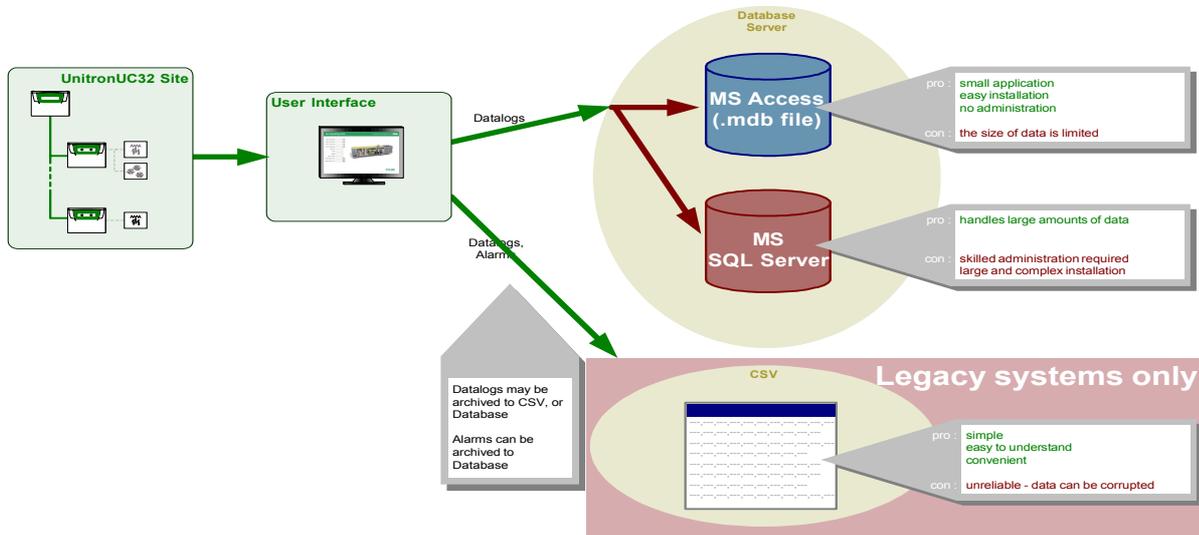
Note: The Datalog Database Archiving application (DDA) allows you to manage the size of large MSAccess datalog databases by archiving data to a separate MSAccess .mbd, or to CSV files. This can often be enough to overcome the data size restriction.

- **MSSQL Server Database** – very scalable and reliable, and can handle very large amounts of data. However, it is complex to set up and to administer.

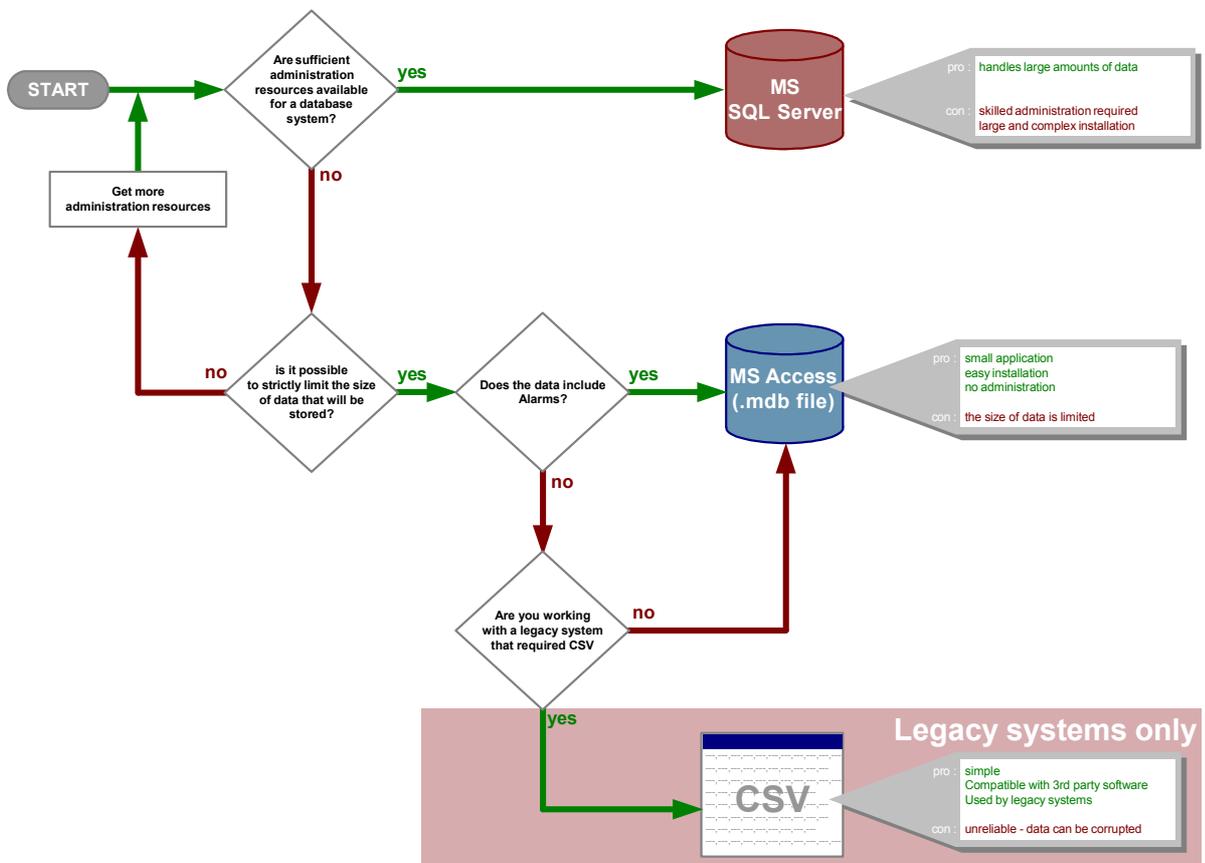
In some legacy systems Datalog data may be stored directly in **CSV** (Comma Separated Values text file). This storage platform is simple to use, and can be accessed directly by many 3rd party applications. However

1. CSV is available for Datalogs only.
2. CSV is **not** reliable for large-scale use.
3. Triggered Time-Stamped datalogs cannot be accurately represented in the legacy Cylon CSV format. Instead, a regular time interval is assumed and the values supplied are extrapolated to fit into this time interval. This reduces the accuracy of such datalogs.

Data Storage Platforms in the UnitronUC32 system

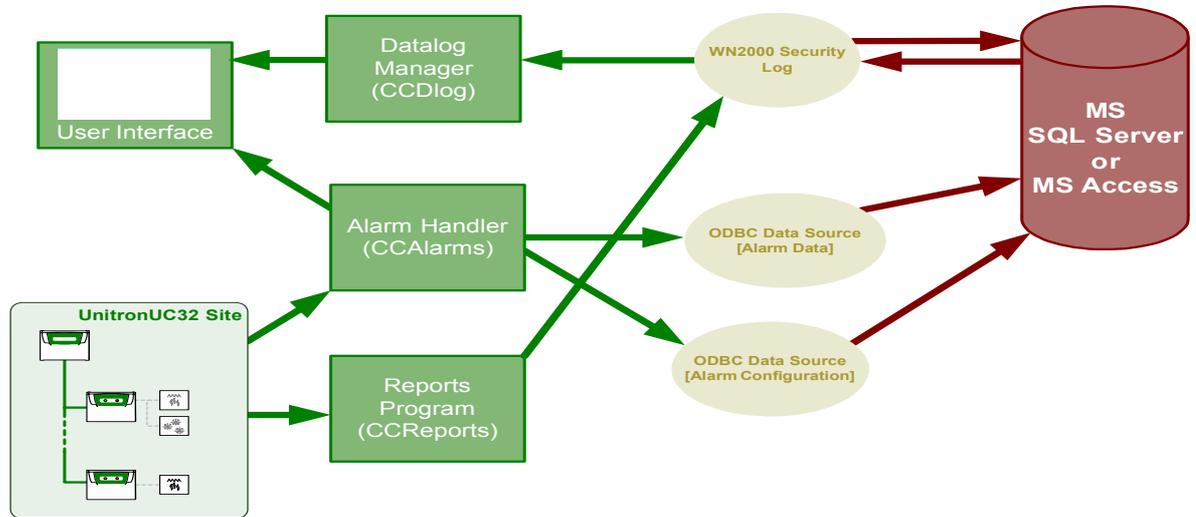


How to choose a Data Storage Platform



Database Information Flow

Database Information Flow in the UnitronUC32 System



How to Set up SQL Server (Express) in the UnitronUC32 system

In order to use SQL Server as the Data Storage platform for the UnitronUC32 system, you must first install SQL Server Express on your PC as follows:

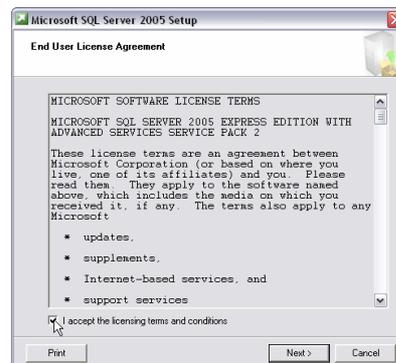
1. Run SQLEXPRESS_ADV.EXE

This file is available on the Unitron Engineering Centre CD, or as a 250 Mb download from the Microsoft website: <http://go.microsoft.com/fwlink/?LinkId=135973>

2. Click run

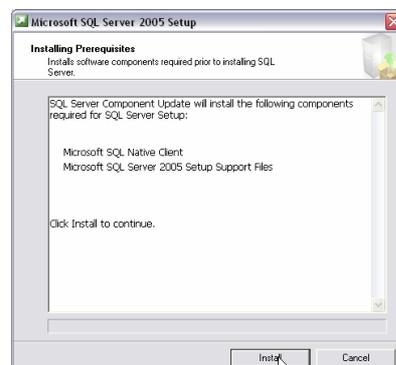


3. Read the terms carefully, then tick the "I accept the licensing terms and conditions" unless you have a specific reason not to proceed with the installation.

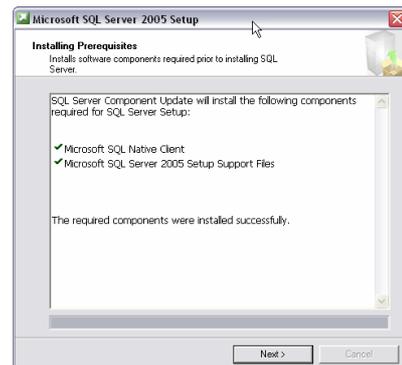


4. Click Next

5. Click Install



6. Click next



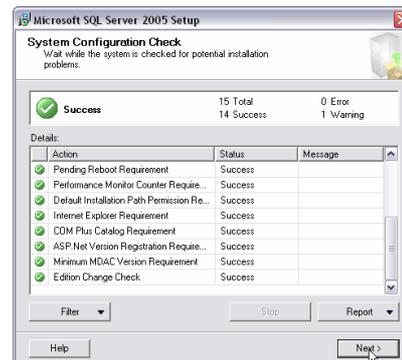
7. Click Next



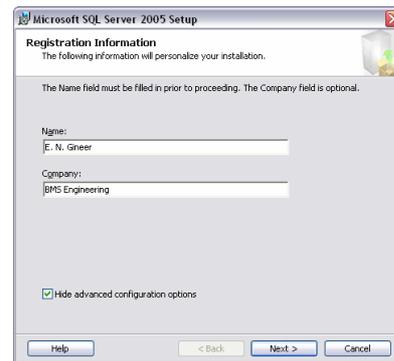
8. The System Configuration Check screen is displayed. View the action list, and correct any faults that are displayed .

Note: An error about IIS may be ignored.

9. Once all critical issues have been resolved, click Next

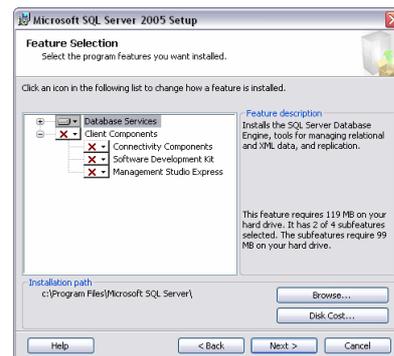


10. Enter a Name and Company for the installation, then click Next

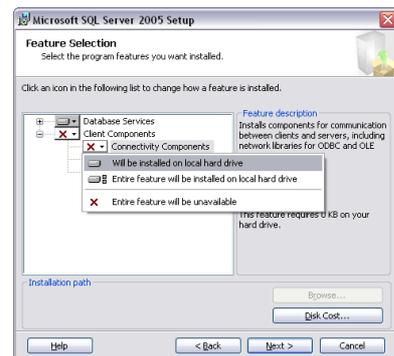


11. Select the optional features to be installed

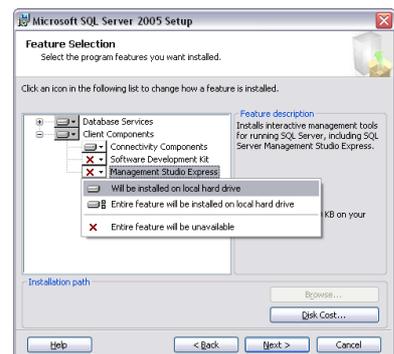
Note: some features that you will need for setting up the UnitronUC32 databases are NOT installed by default. You **must** add them here.



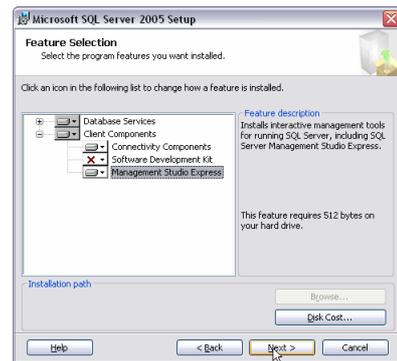
12. select "Connectivity Components" for installation.



13. select "Management Studio Express" for installation.



14. When Database Services, Connectivity Components and Management Studio Express are all selected for install, click Next



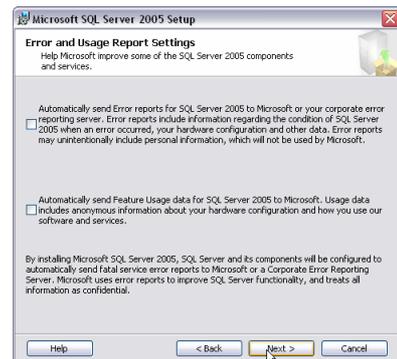
15. Select "Mixed Mode" as the Authentication mode .
Enter "cylonctl" as the password.



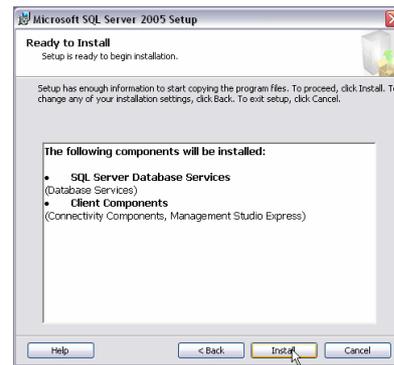
16. Select security options and click Next



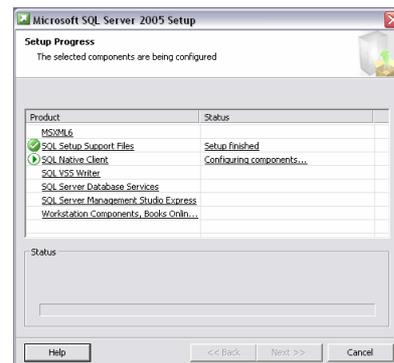
17. Click Next



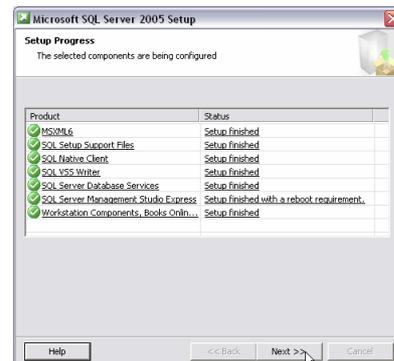
18. Click Install



19. Click Next



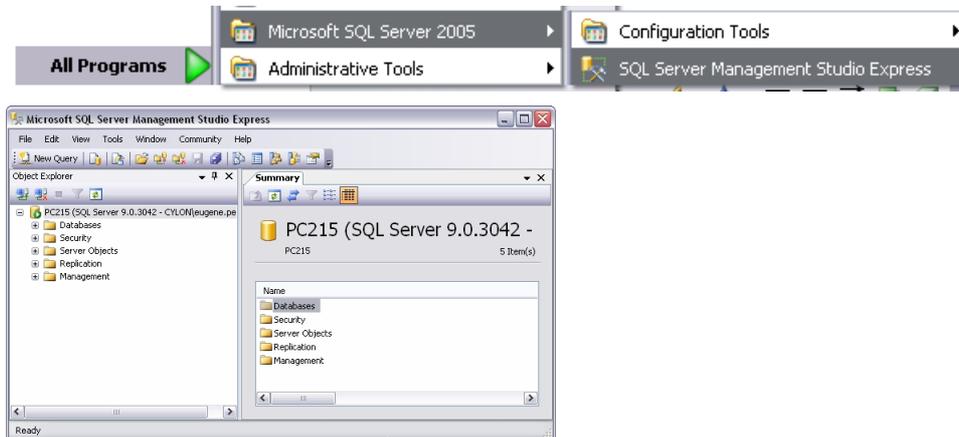
20. Click Finish



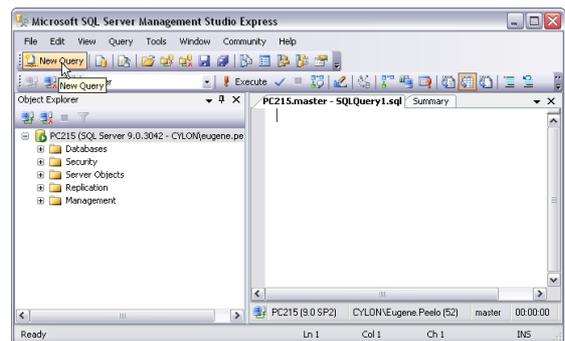
The SQL Server is now installed on your PC and running.

Next, set up the Cylon table Structure as follows:

21. Start the SQL Server Management Studio Express from the windows Start menu



22. Click on New Query



23. Paste in the following:

```

/*=====*
* Copyright(C), 2002 Cylon Controls, All rights reserved *
*=====*

FILE      : Datalogs.sql

DESCRIPTION : This SQL script is to generate a database for datalogs on a MS-SQL
              Server 2000 and later versions.
              The script creates the database, all the tables and initialises the version
              in the configuration table

*=====*/

USE master

CREATE DATABASE WN3000SL
go

USE WN3000SL
    
```

```
CREATE TABLE [Datalog Descriptors]
(
[Datalog ID] int IDENTITY(1,1) PRIMARY KEY CLUSTERED,
[Site Name] nvarchar (50) NULL,
[UCC4 Name] nvarchar (50) NULL,
[UCxx Name] nvarchar (50) NULL,
[Datalog Name] nvarchar (50) NULL,
[Interval] int NULL,
[Units] nvarchar (50) NULL
)

go
GRANT ALL ON [Datalog Descriptors] TO PUBLIC
go

CREATE TABLE [Datalog Values]
(
[Record ID] int IDENTITY(1,1) PRIMARY KEY CLUSTERED,
[Datalog ID] int NOT NULL REFERENCES [Datalog Descriptors] ([Datalog ID]) ON DELETE CASCADE,
[Time] datetime NOT NULL,
[Value] real NOT NULL
)

GRANT ALL ON [Datalog Values] TO PUBLIC

CREATE TABLE [Configuration]
(
[Upgrade ID] int IDENTITY(1,1) PRIMARY KEY CLUSTERED,
[Upgrade Time] datetime DEFAULT GETDATE(),
[High Version] int NULL,
[Low Version] int NULL,
[Comment] nvarchar (255) NULL
)
go

GRANT ALL ON [Configuration] TO PUBLIC
go

--Adding initial records:
BEGIN TRANSACTION
INSERT [Configuration] ([High Version], [Low Version], [Comment]) VALUES('1', '1', 'Initial
Version')
COMMIT TRANSACTION

go

CREATE TABLE [Alarm]
(
AlarmID int IDENTITY(1,1) PRIMARY KEY CLUSTERED,
ConnectedAlarmID int NULL,
SiteName varchar (50) NULL,
SiteNumber smallint NULL,
UCC4Name varchar (50) NULL,
UCC4Number smallint NULL,
UC16Name varchar (50) NULL,
UC16Number smallint NULL,
Priority smallint NULL,
StartedAt datetime NULL,
EndedAt datetime NULL,
AlarmType smallint NULL,
TriggerPointName varchar (50) NULL,
TriggerPointNumber smallint NULL,
TriggerPointType bit NULL,
TriggerPointValue real NULL,
TriggerPointUnit varchar (25) NULL,
AlarmNumber smallint NULL,
ProgramModuleNumber smallint NULL,
```

```
AlarmMessage varchar (300) NULL,
UCC4SysStatus smallint NULL,
UC16SysAlarms smallint NULL,
Note varchar (2048) NULL,
AcknowledgedBy varchar (50) NULL,
AcknowledgedAt datetime NULL,
Suppressed bit NULL,
StringNumber smallint NULL,
ExtraBits smallint NULL,
ExtraInteger int NULL,
ExtraString varchar (50) NULL
)
go

GRANT ALL ON [Alarm] TO PUBLIC

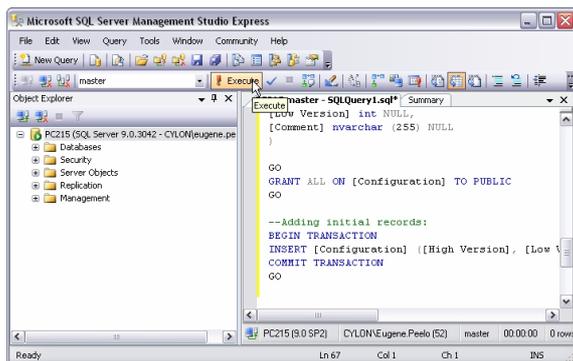
go

-- This function allows comparison by time ignoring date
-- It returns 1, if the time-parameter is less than the
-- composed time-parameter, or 0 otherwise.
CREATE FUNCTION TimeLessThan(@d AS DATETIME, @h AS INT, @m AS INT, @s AS INT) RETURNS BIT
BEGIN
IF DATEPART(hh, @d) < @h
RETURN 1
IF DATEPART(hh, @d) > @h
RETURN 0
IF DATEPART(mi, @d) < @m
RETURN 1
IF DATEPART(mi, @d) > @m
RETURN 0
IF DATEPART(ss, @d) <= @s
RETURN 1
RETURN 0
END
go

-- This function allows comparison by time ignoring date
-- It returns 1, if the time-parameter is greater than the
-- composed time-parameter, or 0 otherwise.
CREATE FUNCTION TimeGreaterThan(@d AS DATETIME, @h AS INT, @m AS INT, @s AS INT) RETURNS BIT
BEGIN
IF DATEPART(hh, @d) > @h
RETURN 1
IF DATEPART(hh, @d) < @h
RETURN 0
IF DATEPART(mi, @d) > @m
RETURN 1
IF DATEPART(mi, @d) < @m
RETURN 0
IF DATEPART(ss, @d) >= @s
RETURN 1
RETURN 0
END

go

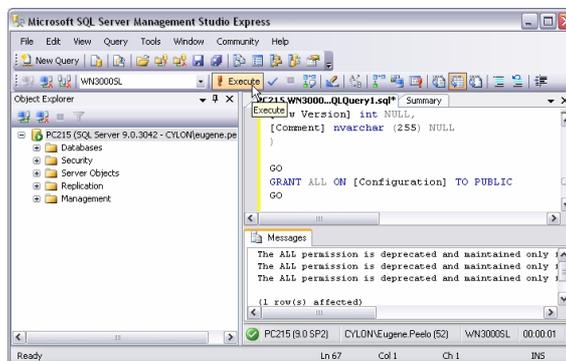
USE master
GO
```



Alternatively you could save the text above to a text file (with the extension '.sql') then use File > Open in the SQL Server Management Studio Express to load it.

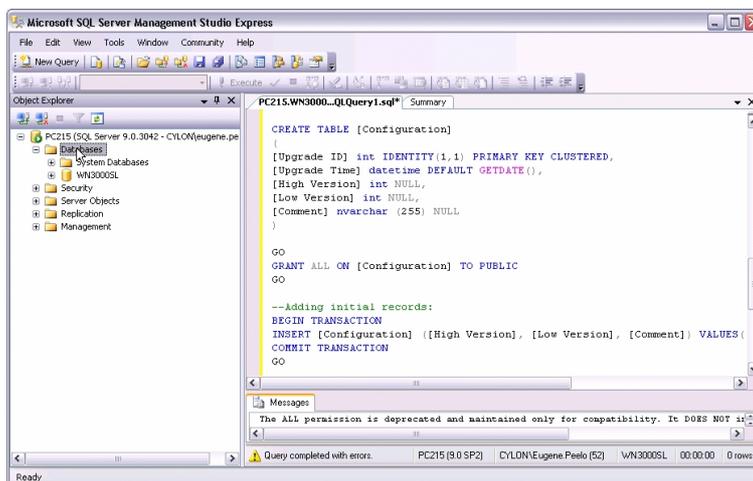
24. Click execute

The database should now be set up.



25. Click on 'databases' in the left-hand pane, and press the [F5] button to refresh it.

The WN3000SL database will then be visible:



Redirecting UnitronUC32 data to the SQL Server

Redirecting Datalog data

To direct data from UnitronUC32 datalogs to the SQL Server platform, use the WN3000 Security Log in the Windows Control panel:

26. Double-click on the WN3000 Security Log icon.



27. Select Connection Type: Microsoft SQL Server 2000 or later.
28. De-select Local Server, and enter the full name of the SQL Server instance including the PC name – e.g. pc165/SQLEXPRESS
29. Click OK

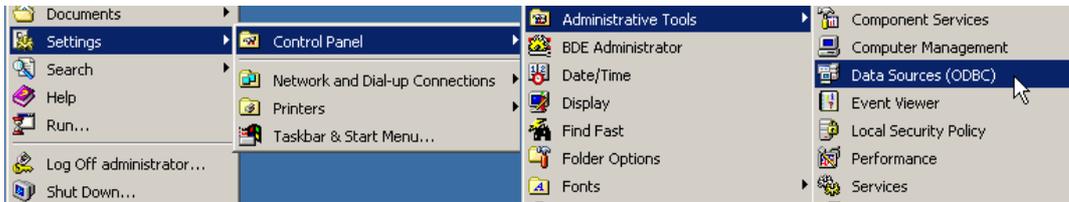


Note: After changing the datalog Data Storage platform to SQL Server, if problems are detected with the storage you can simply change back to MSAccess using the same WN3000 Security Log.

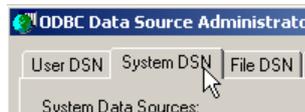
Redirecting Alarm messages to the SQL Server platform

Configure the Alarm Handler to log alarms to SQL Server

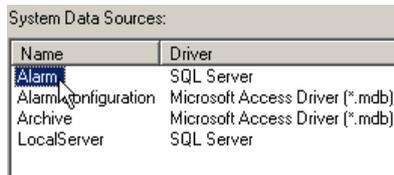
30. Go to start>settings>control panel>administrative tools. Double-click to open **Data sources (ODBC)** dialog.



31. Click the **System DSN** tab.



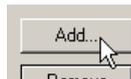
32. Click on the **Alarm DSN** name.



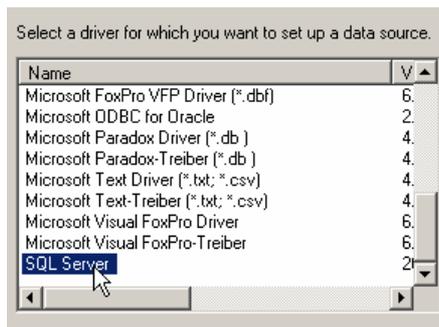
33. Click the **Remove** button to remove this Alarm DSN because it is pointing to Microsoft access and not SQL server.



34. Click the **Add** button.



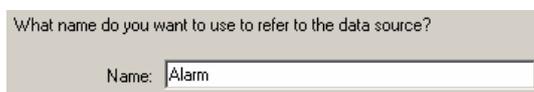
35. Scroll to the bottom of the list and select **SQL Server**.



36. Click **Finish**.



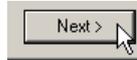
37. Enter **Alarm** in the **Name** field. Leave the **Description** field blank.



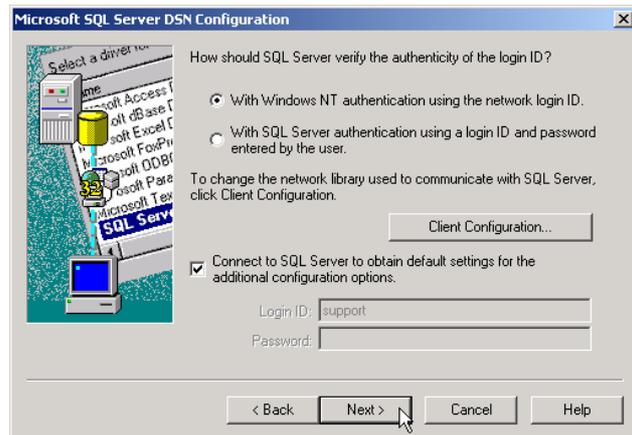
38. Select the appropriate server.



39. Click **Next**.



40. Click **Next**.



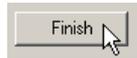
41. Tick the check box **Change the default database to**. Select from the database list - Alarms.



42. Click **Next**.



43. Click **Finish**.



44. Click **OK**.

45.

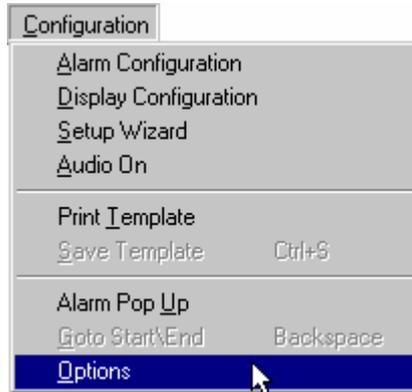


46. Click OK in the ODBC Data Source Administrator dialog.

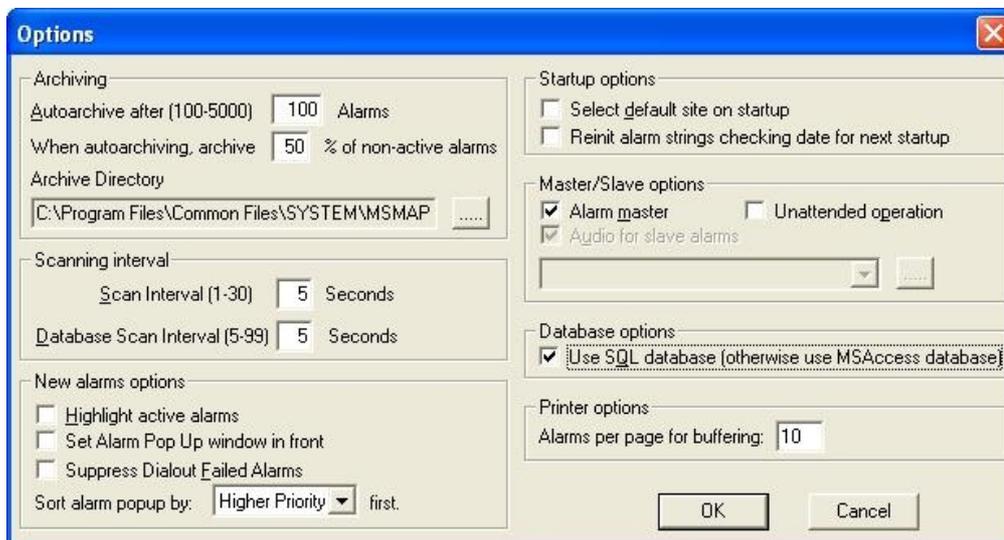
Configure the Alarm Handler to log alarms in SQL format

47. Open the Alarm Handler module in the Unitron Engineering Centre.

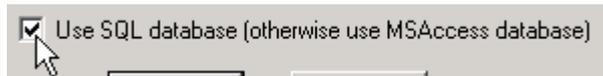
48. Select **Options** from the **Configuration** menu.



The **Options** dialog will open:



49. Select the **Use SQL database** option.



50. Then click the **OK** button to confirm the change.

Querying the SQLServer platform - Cylon Table Structure

The Data Storage Platform contains the following tables:

Datalog Descriptors	Alarm	Configuration
Datalog ID	AlarmID	Upgrade ID
Site Name	ConnectedAlarmID	Upgrade Time
UCC4 Name	SiteName	High Version
UCxx Name	SiteNumber	Low Version
Datalog Name	UCC4Name	Comment
Interval	UCC4Number	
Units	UC16Name	
	UC16Number	
	StartedAt	
	EndedAt	
	AlarmType	
	TriggerPointName	
	TriggerPointNumber	
	TriggerPointType	
	TriggerPointValue	
	TriggerPointUnit	
	AlarmNumber	
	ProgramModuleNumber	
	AlarmMessage	
	UCC4SysStatus	
	UC16SysAlarms	
	Note	
	AcknowledgedBy	
	AcknowledgedAt	
	Suppressed	
	StringNumber	
	ExtraBits	
	ExtraInteger	
	ExtraString	

Datalog Values
Record ID
Datalog ID
Time
Value

Alarms Structure

The Alarms part of the database consists of a single table.

AlarmID	Each alarm that is registered with the UCC software is stored in a row in this table, and is given a unique ID
ConnectedAlarmID	
SiteName SiteNumber UCC4Name UCC4Number UC16Name UC16Number	The address of the alarm is stored in these 6 fields.
StartedAt EndedAt	The start and stop times of the alarm
AlarmType	
TriggerPointName TriggerPointNumber TriggerPointType TriggerPointValue TriggerPointUnit	The point monitored by the Alarm block is described in these 5 fields
AlarmNumber ProgramModuleNumber AlarmMessage UCC4SysStatus UC16SysAlarms Note	
AcknowledgedBy AcknowledgedAt	
Suppressed StringNumber ExtraBits ExtraInteger ExtraString	

Datalog Structure

The datalogs part of the database consists of two tables:

- **Datalog Descriptors**

Every time the software gets a datalog it adds a line to this table (even if it is the same datalog).

The fields are Datalog ID, Site name, UCC4 name (we now call it comms controller name), UCxx Name (we call it field controller name), Datalog name, Interval between samples (ignored) and Units

- **Datalog Values**

This table has a row for every data sample. This consists of a unique ID for the sample (Record ID) pointer to the datalog header (Datalog ID), the timestamp of the data (Time) and the data itself as a Float (value).

Because the reports program collects a snapshot of the datalog and stores it in the database, there will be duplicate data in the database. However there is no requirement to store redundant data when archiving.

Migrating Data from MSAccess to MSSQL Server

It is not recommended that historical data is moved from the MSAccess platform to the SQL Server platform. Instead, it is recommended that the historical data is kept in MSAccess format as an archive, and future data is stored in SQL Server.

Datalog Database Archiving application (DDA)

DDA allows users to:

- manage the size of large MSAccess datalog databases by archiving data to a separate MSAccess .mdb, or to CSV files
- export subsets of a collection or set of Datalogs to MSAccess (mdb) or CSV files. For example, if you had a number of different sites, you could export the datalogs from only one site, and send them to someone interested in only that site. (bureau communicating with customer?)
- Archives from SQL, and MDB and CSV to either MDB or user-defined CSV
- Converts Cylon CSV to user-defined CSV

Note: for User-Defined CSV formats, 3 templates are shipped as examples with the DDA application, and users can easily create their own.