

Using XLeMs as a Collection Engine

Revision	Date	Change
Rev 1	22/03/12	Initial revision

This application note should be read in conjunction with the latest version of the XLeMs User's Manual, available under "Technical Support" on the SHM Website.

Introduction

While XLeMs software is a fully functional Energy Monitoring System, it can also function as a simple intermediary data collection engine to allow historic data from SHM loggers to be accessed by third-party EMS software that does not support the SHM Datalogger Protocol.

XLeMs collects data from the loggers at regular intervals and stores it in its Readings Database. At intervals (which should not be less than the collection interval) the historic data is transferred to the EMS software application.

Historic data can be transferred in two ways :-

1. 'Pushed' by generating CSV reports using the Scheduler and exporting them via FTP
2. 'Pulled' by enabling the oBIX Server and interrogating XLeMs via the network

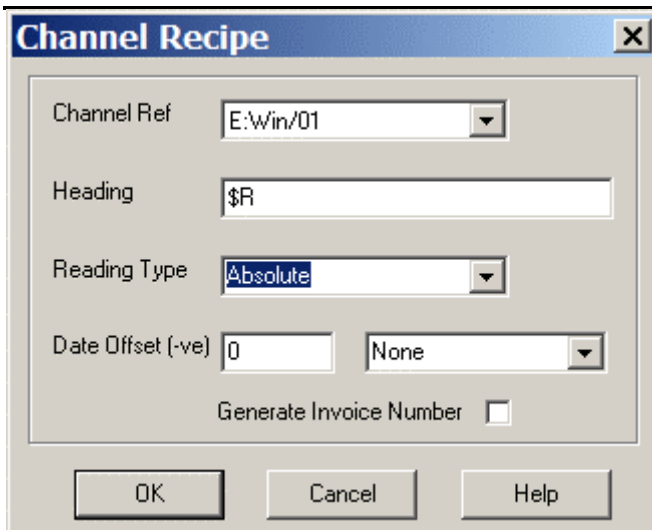
Data Collection

XLeMs must first be configured with details of all the Connections, Loggers and Channels in the system. Be sure to set the **Readings Multiplier** for each channel if you want the data scaled to appropriate units rather than raw logger data.

The XLeMs scheduler is then configured to collect the data at regular intervals, typically daily or hourly. Create a new Scheduler Item of type "Collect Data" to do this.

Pushing Data

First you must create a new XLeMs Template to produce a Report of all the channels in CSV format. If you have a lot of channels (and you must do this if you have more than 250 channels on your system) you will need to split the channels into several Templates. You should make the Interval in the Template the same as the logging interval of the logger (normally 30 minutes). For all Channels in the Template, the **Reading Type** should be set to **Absolute** to record the actual register values for energy meters :-



Channel Recipe

Channel Ref:


Heading:

Reading Type:

Date Offset (-ve):

Generate Invoice Number

You would normally run the Report as often as data is collected, so the Span of the Template should be this interval at least. If you are creating a daily Report, then you would set the **Time Constraint** of the Start Time to "**Start Day**", and for an Hourly report it would be set to "**Hour**" :-



Report Recipe Wizard

General | Start Date | Span | End Date | Channels

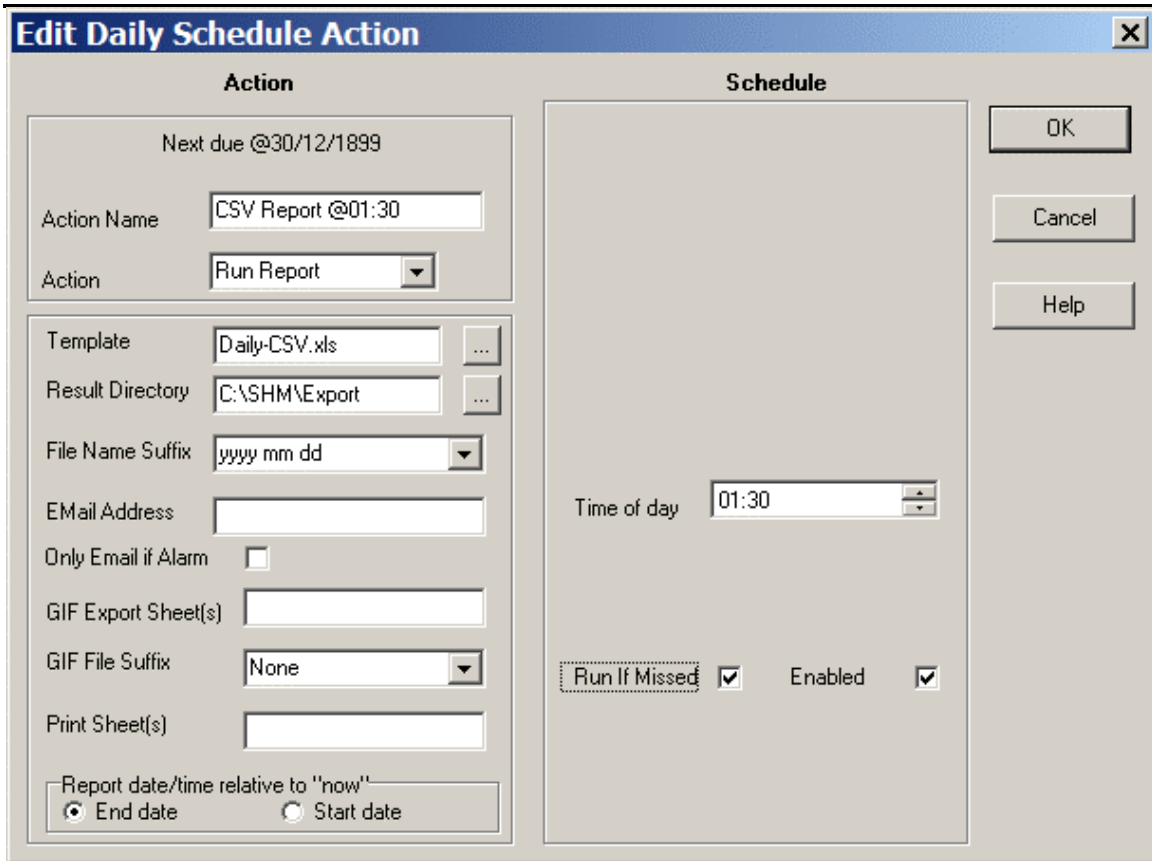
March 2012						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Date Constraint :-

Start Time / Date

Time Constraint :-

You then need to configure the Scheduler to run your Report(s) at regular intervals. If you are collecting data daily, then you might set this Action to run at say 01:00. In that case you could run your Report(s) some time later, at say 01:30 :-



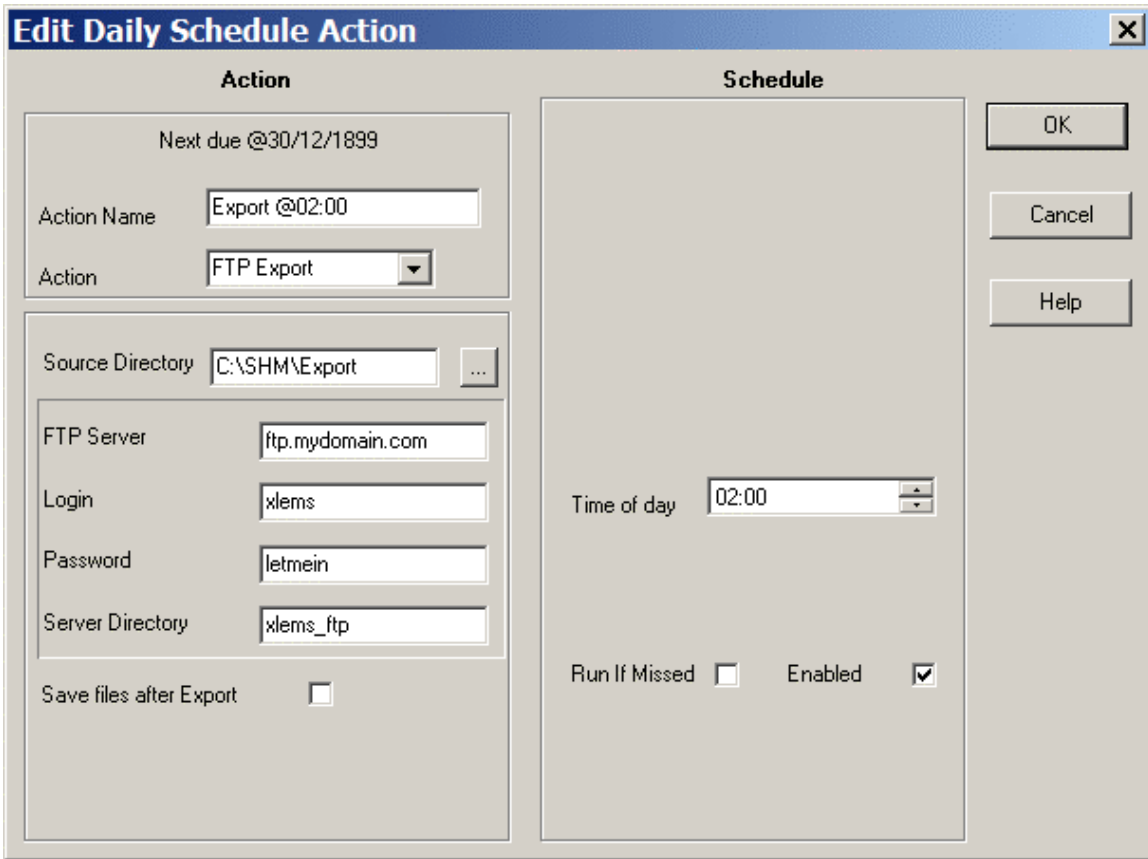
- In this example, the Template **Daily-CSV.xls** has already been set up to give 48 half-hour readings for all the Channels in the system.
- Set the **Result Directory** to be a special directory where all the CSV files will go.
- Set the **File Name Suffix** to be **yyyy mm dd** so that each day's CSV file will have a unique name.

When the Scheduler runs this Action at 01:30, the start date and time will snap to 00:00 the previous day, and will produce a CSV file containing all the previous day's readings. A typical file produced by the running above Action on 23/03/2012 would be called **Daily-CSV-2012-03-22.csv** with data :-

```
Date/Time,SQ:Ach/01,E:Win/01
22/03/2012 00:00,778363,5141.1
22/03/2012 00:30,778363,5141.2
22/03/2012 01:00,778363,5141.3
22/03/2012 01:30,778363,5141.5
... etc. etc.
```

If the **Results** directory where these files are stored is a shared folder on your network, then they can be read by your EMS application directly. However if this is not convenient or possible, you can arrange to export the files to an FTP Server. To

do this you need to create another Scheduler Action which you would run after the Reports, say at 02:00 :-



Edit Daily Schedule Action

Action

Next due @30/12/1899

Action Name:

Action:

Source Directory: ...

FTP Server:

Login:

Password:

Server Directory:

Save files after Export:

Schedule

Time of day:

Run If Missed: Enabled:

OK
Cancel
Help

This Action will run at 02:00 every day and export all files in the **C:\SHM\Export** directory to the FTP Server. Files that are successfully exported are deleted from the directory afterwards.

Collection Failure Problem

There is a slight problem with the above scheme if data collection from one or more channels fails. Data collection is normally very reliable, but when running over BT or GSM modem connections, occasional failures are inevitable. However any connection link may fail and be out of action for a time until it is repaired.

For example if the network connection to one of your loggers fails some time on Monday afternoon, when XLeMs collects the data at 01:00 on Tuesday, it won't get any data for those Channels affected. That means that the Report generated at 01:30 for Monday will have blanks in it for those Channels.

If the link is repaired later on Tuesday then when XLeMs collects at 01:00 on Wednesday it will pick up Monday's data as well as Tuesday's. The Report generated at 01:30 will be for Tuesday and will contain all the readings. However there is no way of automatically going back and re-running the Report for Monday (which would now contain valid readings as the data has now been collected). You would have to manually re-run the Report for Monday.

A workaround for this problem is to configure the Template to produce Reports for a longer period, say 7 days. Each Report would contain data for the previous 7 days, but it would be produced daily. This means that the EMS system that processes the CSV files will get a lot of duplicate data, but provided any collection failures are rectified within 7 days, any blanks will eventually get filled in.

oBIX Server

Enabling the oBIX Server feature in the XLeMs Scheduler offers a better way of accessing the historic data in the XLeMs database. It avoids the problem of collection failures, because the EMS system's oBIX Client will send a query asking for data with a timestamp later than the last good one it received. If data collection has failed, the oBIX Server will not return any readings. Once data collection resumes, it will return all the missing readings.

To use the oBIX Server, you only need to enable it and set the TCP Port (default 80) – refer to the manual. You do not need to define any Templates or run any Reports. You just need to configure the Scheduler to collect data at regular intervals.

An increasing number of EMS software applications are supporting oBIX, which makes this the preferred choice.

Scheduler Configuration

In addition to the above, it is useful to configure the Scheduler to carry out a few more Actions to make the system work better (refer to the manual for full details). These are :-

- Back up the database every week. Choose a file suffix such as **Week of Year** to produce a unique file for each week.
- Prune the database every month to make things more efficient. You don't need to store more than the previous 12 month's worth of data, since XLeMs is only acting as a buffer between the dataloggers and the EMS system's own database.
- Create an "Alarm" Template to check that data has been collected successfully and run it every day. There is a sample Template provided with XLeMs that can be used as a guide for this. If data collection fails for one or more Channels, an Email can be sent containing the details of the failure.

A typical list of Actions might be (for FTP Push) :-

XLeMs Scheduler [XLeMs.xsh]

File Scheduler Collect Tools Help

Start Stop XLeMs

- Hourly
 - Daily**
 - Collect Data @01:00
 - CSV Report @01:30
 - Export @02:00
 - Test Collection @03:00
 - Weekly
 - DB Backup @09:00 Mon
 - Monthly
 - Prune Database 1st
 - Yearly

```

Collected 4 readings
Done channel #160
Testing Chan #207
Tx :-#01NB0D
Rx :- #01NB2D
Done logger #2
Thread exiting...
Thread Disconnected
Thread done
Done connection #3
Finished Collection (Single Connection)
  
```

ChanID	ChanRef	ChanNum	LastStored	Descrip

23/03/2012 15:54:13 Comms (Inactive) Next Schedule at 24/03/2012 01:00:00 Version 2.0.0 beta.4