SIR REPORTER

The Newsletter for SIR UK Users

Contents

Editorial

It's over six months since the last issue of Reporter was published and a lot has been happening in the SIR world. SIR Pty have been particularly busy. They have fixed all the reported bugs and 'features' in SIR 2002. If you've ever had need to report a problem to SIR Pty you will know just how responsive they are. I was having problems with merging two databases, so I sent the details to Australia and within less than a week there was a new version of SIR 2002 available for download from SIR's web site. I doubt whether I would have had that sort of response from Microsoft!

But it's not just sorting out user problems that keeps SIR Pty occupied. They have also been very busy developing the next major release of SIR, SIR/XS. This will include, amongst other things, the ability to read and output XML, SQL in PQL and something some of us have been wanting for a long time, 32 character names. Full details of all the new features to be included in SIR XS can be found on subsequent pages.

The SIR UK User Group committee have also been busy. We had a user training and conference in Manchester at the beginning of July which I'm sure everyone who was there would agree, was very successful. Thanks go to all the contributors and to Adrian Hodgson and his SIR colleagues at ORC for organising the three days and ensuring that everything went down smoothly (including the Boddingtons!). A full report of the event is included in this issue.

As well as organising conferences and training the committee produces Reporter but perhaps our most important role is to keep in contact with SIR Pty, letting them know what users in the UK think about SIR and how they would like to see the product evolve. The committee is elected at the AGM and we are always looking for new blood. However, this year we failed to attract any new recruits and are looking to co-opt new committee members as our chairman points out in his column. So if you care about SIR and could manage a meeting once every two months or so, contact Dave Doulton – d.c.doulton@soton.ac.uk. We'll be very pleased to have you on board.

Fran Williams
Fwill@essex.ac.uk

SIR is at http://www.sir.com.au
UK SIR Users Group is at http://www.soton.ac.uk/~sug

Chairman's Chat

It seems that yet again I have been elected Chair of this august organization, so yet again it is up to me to fill this space. Firstly let me say that we are still short of committee members to help create Reporter and to organize our events. As they say many hands make light work but with just a few it becomes heavy work. Please contact me if you think you can help or would like further details.

Since the last Reporter we have had a very successful conference in the heart of Manchester reports of which are contained in this issue. Next year we are aiming to have just a one day conference in London or Edinburgh as there will, we hope, be an International Conference in New York not organized by this committee but being up to our usual high standards.

At this point there should be versions of SIR/XS the next release of SIR to see in action. It will contain some more useful features – see elsewhere in this issue. The SIR product just keeps on getting better, hopefully you all agree.

Hope to see you at one event or the other next year.

Dave Doulton

SIR Training and Users Conference July 2-4 2003

Jarvis Piccadilly Hotel, Manchester

Manchester was more than adequately compensated for the loss of David Beckham by the arrival of 25 delegates and tutors for the 2003 SIR International Training and Conference.

The event took the now usual format of 1½ days of parallel training sessions for novices and advanced users, followed by a half day's conference. The venue was the Piccadilly Hotel in the city centre, and the meeting was organised by the SIR Users Group (UK) Committee, with help from local SIR client, ORC International.

John Lemon of Aberdeen University led the ***SIR introductory training***, which covered a variety of topics, including where SIR came from ("A Brief History and Background of SIR"), schema design, data input in SIR2002, and an overview of the utilities such as UNLOAD etc. The trainees – one of whom had used SIR for six years but came along for fun and still learnt something – spent the major part of the course on PQL retrievals trying to find "all the blue eyed children of blue eyed parents" and answer questions like "whether the blood group of the mother affects the sex of the baby" (John uses two of his own databases rather than COMPANY for his training). All the way through he emphasised why SIR with PQL remains so good at data management and retrievals in the Windows XP age.

Meanwhile this year's advanced training elegantly brought together three quite distinct SIR subjects under the title of ***Client/Server Applications with SIR***. **Tony Reardon** (SIR Pty, Australia) and **Tom Shriver**, of Datavisor, USA gave us a step-by-step guide to mastering Master, SIR's tool for managing concurrent use of databases. In addition to carrying out its primary function, Master provides protection against data corruption when your system unexpectedly crashes.

Next morning, following the Manchester "Ale Trail" the night before, our minds were stimulated by Open Database Connectivity (ODBC), something I've actually been toying with myself as I try to get Microsoft Access to speak coherently to my SIR databases. Again, Tony and Tom were our tutors, explaining how to use the ODBC tools in PQL to read from other software packages (including Access, Excel and SPSS). Tom has developed an ODBC system in which he can program just about every piece of data manipulation in PQL.

Concluding the advanced training programme, **Dave Doulton** from Southampton University demonstrated how a little working knowledge of the web design language HTML can be combined with the SIR executable file SIRWEB.CGI to create sophisticated interactive web pages. You can, for example, enter a form-full of data on a web page, press the send button, and in seconds receive on your screen a nicely-formatted report from a remote SIR database. This method is freely available to anyone using SIR2000/2002. We can all try it – and probably should!

On the last morning, following the previous night's Course Dinner, came the nourishing ***SIR UK Users International Conference***.

The conference is a good place to hear what developments SIR have been working on in the last year and what they're likely to be doing in the next. And they've been very busy. **Tony Reardon**'s keynote talk reminded us that in SIR2002 we now have many more facilities than a few years ago, including multiple databases, Master, SIR SQL Server, long character strings, GUI screens and screen painters, and a web interface – to name but a few. And where is SIR going next? It will continue to support multiple operating platforms and retain compatibility with earlier versions of SIR, while improving ease of use and introducing new training modules. Visual PQL will be further strengthened as SIR's primary programming tool. In a recent survey, SIR found its users were keenest to have 32-character variable names (76% of replies) and improved recovery facilities (60%). Work on these and many other features is already well advanced. A beta release of the new version of SIR is expected next year. Tony asked users (not just those at the Conference) to give him their requests and ideas for new features – take him at his word by emailing tony@sir.com.au!

This year's users' papers – from a particularly experienced selection of users - were as follows:

Tom Shriver (Datavisor) – *"Using ODBC to Migrate an SQL Server database to SIR"*. In this case study from his own recent experience, Tom was able to automate much of the conversion of data from a major New York study of cancer patterns. He used PQL programs to pick up a new SIR schema from the output of the Access documenter facility, even coping with the reduction of long variable names to SIR's 8 characters (one of the delegates helpfully suggested he should have waited for the next version of SIR, which promises long variable names!). Unfortunately, Tom has found that the technical conversion to SIR was one of the easiest parts of the job, as there had been many flaws in the original database design, management and data input. But at least its future operation will be more dependable.

John Lemon (Aberdeen University) – *"How to avoid some of your SIR disasters: Ten Top Tips"*. John provided a reminder of all the good practices we once learned and should still apply every time we take responsibility for a SIR database.

(1) Make sure your backups are done often enough and reliably enough (and automate them in PQL programs).
(2) Verify your file regularly.
(3) If it's corrupted, you can still recover much of your data using a Journal File and/or the Dump utility.
(4) Label and document your schemas thoughtfully for the next person to understand.
(5) Ditto for your PQL programs, sprinkling helpful comments (C or |) within the code.
(6) Indent your program text for clarity.
(7) Organise your procedure file into meaningful families with sensible names.
(8) Be efficient in schema design – keep your record keys small, use long string and real number variables sparingly, leave free columns to insert variables later.
(9) Use subroutines and subprocedures in programs where possible.
(10) Think about your program logic to achieve faster operation. Well, how did you fare?

Randy Banks (Essex University) – *"Dynamic database information retrieval using SIR/CGI"*. In this colourful presentation, Randy gave us a live demonstration of extracting schema information from a SIR database via a website. And very well it worked too (even avoiding the usual unexpected glitches of live computer demonstrations!). Randy is an enthusiastic "LSD" user, which he eventually revealed to be a standardised, fits-all-databases suite of procedures giving him a web-based method replacement for SIR's List Schema Detailed utility (whence the name of the program – "LSD"). Perl "stubs" are used to pass CGI parameters through the SIR/Web interface to his databases. The basic method can be applied to all sorts of interrogations of SIR databases and, being on a website, can be accessed by anyone, anywhere, with the right passwords.

There followed a brief AGM of the SIR UK Users Group Committee, after which we adjourned to lunch and headed home to chew over the many things we'd learned.

Patrick Brown, I.O.M.

SIR XS Development

The next release of SIR, named SIR/XS (eXtended System), is planned to have a number of major new features. Work is already well underway and some features are already implemented.

32 Character Names

Perhaps the feature requiring the most extensive changes internally and that will be most noticeable is to allow **32 character names** and this is already working in DBMS. New long names can be used wherever a SIR name is required:

Database Schema
- Database Names
- Record Names
- Variable Names
- Index Names
- Passwords

Tabfiles & Tables
Families
Members
VisualPQL
- Variables
- Sub-Routines
- Sub-Procedures
- Labels
- Buffer Names
- Filenames

As well as extended standard names that fit the current SIR naming standards except that they are longer (start with alpha, capitalized, contain letters, numbers, four special characters $ # @), SIR/XS also allows you to specify non-standard names by enclosing them in curly braces { }. These can be from 1 to 30 characters (plus braces) and have no translation, can contain any characters and are stored without braces so sort normally.

XML Support

Another new feature is support for **XML.** XML is a generalized markup language (eXtensible Markup Language). An XML File is a text file that has a hierarchical structure where *tags* enclose content. It resembles HTML but uses a users own tags. Support is going to include both reading and writing XML. A new PQL Procedure, XML SAVE FILE that produces XML files is already implemented.

```
XML SAVE FILE FILENAME = filename /
 ROOT     = 'string' /
 BREAK = variable (TAG = 'string',
 ATTRIBUTES = (varname (format)),..)
 ELEMENTS   = (varname(format)),..)/
 BOOLEAN = (logical expression) /
 MISSCHAR = character /
 SAMPLE = fraction/
 SORT   = variable,....
 DTD    [= filename]
 SCHEMA [= filename]
```

The ROOT clause specifies the outermost tag. Data is written in a nested hierarchy determined by multiple BREAK variables. At each level data can be written either as attributes or as elements. The tags used can be variable names (default) or can be specified in the command. Schema information can be written to separate files as a Document Type Definition (DTD) and/or as an XML schema.

When creating an XML file in SIR, the application can restrict itself to specific parts of the standard and still produce compliant, standard XML. However, if SIR is to claim to read XML, it has to cope with any XML compliant file. While it is relatively straightforward to produce an interface that copes with files such as the simple XML produced by SIR, it is much more difficult to meet the standard under all circumstances. For example XML files can use different encodings to deal with other languages.

The proposal is to use the standard SIR READ command to process a file (defined in the OPEN as XML) and then provide a set of functions to return information. These PQL commands and functions would interface to a standard XML reader called Libxml2 XmlTextReader. Libxml2 is the XML C parser and toolkit developed for the Gnome project (but usable outside of the Gnome platform). It is free software available under the MIT License. Libxml2 is known to be very portable, the library should build and work without serious troubles on the variety of systems that we require for SIR (Linux, Unix, Windows). There is direct support for namespaces, xml:base, entity handling and DTD validation. This is close to the DOM Core specification.

In most cases libxml2 tries to implement the specifications in a relatively strictly compliant way. As of release 2.4.16, libxml2 passes all 1800+ tests from the OASIS XML Tests Suite.

To read XML in PQL the program identifies the file as XML on the OPEN and simply issues READ commands repeatedly to progress to each node in sequence in document order. If the XML references a DTD, it can be used to validate the parsed document progressively and/or to implement attribute defaulting and/or entity substitution. There are new clauses on the OPEN to specify how a DTD is used.

At each READ, the program is located on a given node and a set of new PQL functions can be used to query the node properties and return names and values.

For example, the NODETYPE function returns what type of node has been read (1 for start element, 15 for end of element, 2 for attributes, 3 for text nodes, 4 for CData sections, 5 for entity references, 6 for entity declarations, 7 for PIs, 8 for comments, 9 for the document nodes, 10 for DTD/Doctype nodes, 11 for document fragment and 12 for notation nodes).

We also intend to investigate how XML can be incorporated as an option into some standard utilities e.g SIR FILE DUMP …./XML .

SIR/XS contains additional features to assist in generic processing of tagged files, that is files where content is surrounded by start and finish tags. If an output file is identified as a tagged file on the OPEN, then subsequent WRITE commands surround each output element with a tag. The tag defaults to a variable name or can be specified on the WRITE.

Recovery

SIR/XS extends and improves **Recovery** and much of this is already implemented**.** Journaling has been completely re-written and SIR/XS journal files have a new structure which enables better checking and reporting plus some new features. There is a new journal utility, Journal Rollback that rolls back a database from given place. SIR/XS will also support **Transaction Processing** with PQL commands to start a transaction, end a transaction (Commit) or abort a transaction. This is journal based and does a rollback on abort.

Another new feature is **Incremental Unload** which creates a journal of updates added to the end of an existing unload.

There are new PQL commands to process journals enabling user written audit trails and other recovery strategies. These journal commands are:

```
PROCESS JOURNAL
JOURNAL RECORD IS record_name
    PQL access to record variables
END JOURNAL RECORD IS
END PROCESS JOURNAL
```

These commands provide a simple and straightforward way to select which sets of journal records to process and to retrieve data from specific journal records in exactly the same way that data is retrieved from a database record.

New GUI Features

There are a number of **new GUI controls** already implemented in SIR/XS as follows:

`COMBO`
This is a CHOICE control that lets you edit the text. All the EDIT and CHOICE functions and commands work here.

`SPIN`
This is a pair of arrows which alter a set of values up or down. e.g. to scroll through a set of allowed numbers. The shorter of height & width determines if it is horizontal or vertical spinner. The spinner can be used to update a value in the previously defined edit (or other) control.

`SLIDER`
This is a scale with a movable handle. It is similar to SPIN in that it uses the same messages, commands and functions.

`PROGRESS`
This is a progress bar under user control. The shorter of height & width determines if it is horizontal or vertical bar

`TREE`
A TREE control is a hierarchical multiple level display which the user can expand to show items to lower levels or contract to hide lower levels. A plus sign is displayed against any items that can be expanded. The control is fully built by the program before it is displayed and then the expansion/contraction is automatically handled by the control without further programming.

SQL in PQL

SIR/XS will have a new **PROCESS SELECT** command which is essentially SQL in PQL

```
PROCESS SELECT
  variable_list | * | ALL
  FROM     [ tabfile.   ] table_name
           [ database.] record_name
           [INDEXED BY index_name]
  [WHERE condition]
...
END PROCESS SELECT
```

This processes from multiple data sources and does a JOIN if more than one table/record. Data is returned in the sequence of first named data source (or index sequence if accessed through secondary index).

You can use WHERE to restrict which records are selected using any of the PQL conditions and expressions but cannot specify subqueries.

The command returns one set of named variables at a time and there is no aggregation, no GROUP BY, and no UNION. Work has started on the compilation phase of this.

Multiple Data Files

Another feature planned (but not yet implemented) is **Multiple Data Files** allowing you to split the .sr3 data file across different directories/disk drives. There is no overhead for existing single file databases. The split is based on serial key ranges which means case values if a case structured database and record type/key values if a caseless database.

There is a new schema command which can be issued before any data added otherwise will require restructure:

```
DATA FILES
FILENAME = ' ' /
FILENAME = ' ' FROM keyvalue /
FILENAME = ' ' FROM keyvalue
```

SIR Server

Another planned new feature is a **SIR Server** though work has not yet started on this.

Similar in concept to SQLServer this is started on network, waits for users to logon, processes requests, and returns results. Requests can be logon, run named procedure, logoff. For security and control reasons the server does not accept generic SIR commands or PQL programs from clients, it accepts only requests to run saved procedures which can be password protected. It accesses a single system procedure file as specified by the system administrator. If the procedure creates text output then all output to standard output goes back to client.

Procedures can contain any SIR commands and can run PQL.

A Client program can logon, send request to run a procedure, can check error codes, get output line at a time then logoff.

Other New Features

There are various minor new features that are already implemented including**:**

Allow 'filename' as synonym for 'dsn' in open
Cross record functions CNT CNTR MAX MAXR MIN MINR work with string values

PWRITE writes last amended date and time written on procedure command

PREAD has option to overwrite based on date/time

Facility to rename family

Variable Labels separate from variable documentation. New command VAR DOCUMENT

Tony Reardon
Technical Director, SIR Pty

SIR UK User Group Committee Members 2002/2003

Randy Banks (Treasurer)
ISER
University of Essex
Wivenhoe Park
Colchester CO4 3SQ
Tel: 01206 873067
Fax: 01206 873151
email: randy@essex.ac.uk

Kathy Brooks (Reporter)
Forvus
53 Clapham Common
South Side
London SW4 9BX
Tel: 020 7819 1012
Fax: 020 7819 1010
email: kathy@forvus.co.uk

Patrick Brown
Institute of Occupational Medicine
8 Roxburgh Place
Edinburgh EH8 9SU
Tel: 0131 667 5131
Fax: 0131 667 0136
email: Patrick.Brown@iomhq.org.uk

Dave Doulton (Chair)
University of Southampton
Computing Services
Highfield
Southampton SO17 1BJ
Tel: 023 8059 3541
Fax: 023 8059 3131
email: D.C.Doulton@soton.ac.uk

Adrian Hodgson
ORC International
5th Floor City Point
701 Chester Road
Stretford
Manchester M32 0RW
Tel: 0161 877 6781 (switchboard)
Fax: 0161 872 3997
Email: adrian.hodgson@orc.co.uk

John S. Lemon
Aberdeen University Computing Centre
Edward Wright Building
Dunbar Street
Aberdeen AB24 3QY
Tel: 01224 273350
Fax: 01224 273372
Email: j.s.lemon@abdn.ac.uk

Mo Reardon/Tony Reardon
SIR PTY LTD
312 Mona Vale Road
Terrey Hills
NSW 2084, Australia
Tel: 00612 9450 2354
Fax: 00612 9475 1430
email: mo@sir.com.au
email: tony@sir.com.au

Frances Williams (Secretary)
ISER
University of Essex
Wivenhoe Park
Colchester CO4 3SQ
Tel: 01206 873568
Fax: 01206 873151
email: fwill@essex.ac.uk

You can email the whole committee by addressing your message to **sug.comm@essex.ac.uk**