

Tree•times

A Publication of Treehouse Software, Inc.

This Issue

DPS X-Link: Instant XML Gateway for ADABAS	1
Editor's Notes	2
PAC Vs. N2O (The Gloves are Off)	2
Map It on Your PC	4
An Important Question About XML Gateway Technology	7



Download Your Free Trial of
DPS X-Link Today!

Just go to:
treehouse.com/dpsxlnk.shtml



Treehouse Software, Inc.

DPS X-Link: Instant XML Gateway for ADABAS

by Dan Vimont
and
Joseph Brady

SOA



DPS X-Link is "plug-n-play" middleware that provides client applications with fully-automated, real-time, read/write access to ADABAS data and metadata in XML and XML schema format.

What role can **DPS X-Link** play at a customer site?

CLIENT/SERVER

Development of Java and .NET applications that need remote, read/write access to ADABAS.

- >> "Old-school" client/server: Interface with **DPS X-Link** via direct invocation of X-Link's Java, .NET, or COM functions.
- >> "New-school" client/server: Interface with **DPS X-Link** as a Web Service via X-Link's SOAP/WSDL functionality.

(Note that X-Link's presentation of ADABAS data as XML and ADABAS metadata as XML Schema is **perfect for new Java/.NET development.**)

EDI MODERNIZATION

Replacing a legacy EDI configuration with a new B2B Web Service replacement? Using **DPS X-Link** (in conjunction with one of the many BPM/BPEL tools available) allows for plug-n-play replacement of the EDI configuration, while leaving the legacy batch procedures in place (i.e., unplug EDI, plug in B2B Web Services).

INSTANT INCORPORATION OF ADABAS INTO SOA

For sites serious about "going SOA", **DPS X-Link** provides instant incorporation of ADABAS into their new SOA environment as a fully-operational (SOAP/WSDL) Web Service.

RDBMS-to-ADABAS PROPAGATION

By configuring **DPS X-Link** with the offerings of IBM (Q Replication and Event Publishing), Oracle (Oracle Streaming), and others, we can deliver an RDBMS-to-ADABAS propagation solution.

INTEGRATION OF ADABAS AND SAP

Thanks to the new open-standards (XML-based) interfaces provided by SAP to their ERP offerings, **DPS X-Link** is the optimal middleware bridge between the ADABAS and SAP worlds.

(See page 7 for an important question about XML gateway technology)

Editor's Notes

by Joseph Brady

Tales From the Trails

Treehouse presented *"Simplifying the Management and Sharing of Data - ADABAS and More"* at four locations in Europe this month: Brussels, Belgium; Wiesbaden, Germany; Bonn, Germany; and Berlin, Germany.

The day-long events were met with much interest and enthusiasm by representatives from sites throughout Europe. If you would like to host a special Treehouse Event like this at your site, please contact us.

Over the past few months, all Treehouse sales and technical representatives have been very busy on various consulting jobs, site visits, and product demonstrations. They have visited sites in Alaska, Belgium, California, Canada, Colorado, Delaware, Florida, Germany, Hawaii, Illinois, Luxembourg, Massachusetts, Michigan, Missouri, The Netherlands, New York, North Carolina, Ohio, Texas, U.K., Utah, Venezuela, Virginia, Washington D.C., and Washington State.

Tree·times

Editing, Writing, and Design

Joseph Brady

Contributors

Mark Jones, Wayne Lashley,
Heather Snyder, Michael Szakach,
Dan Vimont

Production and Distribution

Terri Hammerschmitt

Back issues available upon request.
Documentation for all products
is available in hard copy or on
CD-ROM.

Hard Copy Circulation: 8,000

PAC Vs. N2O (The Gloves are Off) by Mark Jones

When I was asked a while ago to compare two software tools, one that I had been using extensively for the past eight years and one that I had used briefly about 12 years ago, I thought to myself, "this is not going to be easy". So the best place to start was to take a good look at the 'new' tool, and in this scenario, it was **N2O**.

But before looking at **N2O**, I recalled many of the basic functionalities that any CCM tool for the SAG world should have across the board:

- Control the change and movement of all programming components, NATURAL modules (source and object), PREDICT information, and 3GL (COBOL, JCL, etc.)
- Give it a version
- Keep all this code "where it belongs" (i.e., keep the NATURAL source and object right on the FUSER files where it currently exists)
- Store prior versions someplace in case we "mess up"
- Allow the ability to look at audit information
- Handle and control concurrent development of the same objects
- Archiving
- Some nice tools and utilities

To my pleasant surprise, N2O fulfilled my list—and more!

Throughout my career, I have noted that increasingly, companies are realizing that their application development processes need to be controlled and standardized. This not only simplifies software production, but also makes it more consistent, enhancing quality while saving time and costs. In fact, controlled development means significant benefits for both software engineers and end-users.

Treehouse Software's **N2O** offers tight control over all stages of software development and deployment. **N2O** is not just a powerful tool for managing various versions of an application, but also an effective means of monitoring development, managing migration, and protecting the (base) production environment.

Now that we have dispensed with the formalities, let's get down to the 'technical' or 'good' stuff.

One of the biggest advantages that **N2O** has over PAC is its ease of use and that, in anybody's language, is a bonus. Once the installation has done the basic CCM 'life cycle' analysis, it then takes literally a few minutes to bring an application under the control of **N2O** -- unlike PAC, which takes days or sometimes weeks to get an application integrated or to achieve the same results. This is primarily because **N2O** does not force users to 'compile' components at the migration stage. In fact, **N2O** does not *force* users to do much of anything, except keep things under control.

In my initial tests (not forgetting that I last used **N2O** about 12 years ago), it took me no longer than a few minutes to define and process an application through **N2O**. That included environment and master event definitions, writing some test NATURAL programs, doing some basic migrations, and 'checking-out' and 'checking-in' of its components. It was, without doubt, easy.

I particularly liked the fact that you "drop **N2O** over your base" (normally Production) environment and work backwards, as it were. This is a huge time saver in comparison to PAC, where you may have to change your base/development/test environments just to make PAC work. With PAC, all of your code has to reside inside PAC, where you cannot edit, compile, or execute it. So, you end up jumping back and forth between the PAC file and you're FUSER(s). Ugh!

(continued on page 3)

PAC Vs. N2O (The Gloves are Off) (continued from page 2)

Are you satisfied with PAC? We'd like to hear from you

Way back in March of 1991, in issue #8 of the Treehouse newsletter (then called TREETIPS), we published an **N2O** vs. PAC comparison, written by a customer who was trialing both products. The customer purchased **N2O** over PAC, citing, "*There is a big difference in philosophy... N2O lets you define the environment you will be using... PAC controls the environment...*" as one of the primary deciding factors.

During the course of that year, the **N2O** vs. PAC competition was heating up, mainly because of some special "promotional pricing" that was being offered for early adopters of PAC. **N2O** was released in the Summer of 1989 and quickly gained a foothold in the NATURAL change management market. When PAC came out in Winter of 1991, there was major catch-up work that needed to be done, thus incredible pricing incentives. So later on, in November of 1991, we published issue #10 of TREETIMES, and out of fairness (in case there were some real advantages in using PAC over **N2O**), we offered to publish testimonials from satisfied PAC users. Now it's 2005, and we are *still* looking for that satisfied PAC customer. And the offer to print that testimonial still stands.

Don't forget about the 'slick' way that **N2O** handles 3GL components. Again, it's a case of defining where your Cobol or JCL components are residing, and then away you go, migrating these back and forth with ease.

I hear someone saying, "What about PREDICT?" And the answer is, "Yes, what about PREDICT?" Would it be pretentious for me to say, "As above -- with ease, and no great setting up or worrying about"? I guess not, as that's the way it is. **N2O** is able to handle PREDICT components as it does with all other types of components.

Another solid feature of **N2O** is its 'Toolbox Subsystem'. The **N2O** Toolbox Subsystem provides tools that can assist programmers in their development activities, including NATURAL object comparison, source code comparison, and source display. The Toolbox also offers a 'Documentation Toolbox' function that provides utilities to display/print NATURAL Objects, File Layouts, Descriptor X-Ref (Cross-Reference) Information, Object Flow Analysis, and Object X-Ref in local **N2O** environments. And finally, the Toolbox gives us the

N2OSCAN utility, which goes beyond the capabilities of the NATURAL SCAN utility, providing programmers and analysts with extensions and flexibility, making searches for specific strings within NATURAL objects incredibly easy. How developers live without these very clever features, I will never know. Yes, PAC does have some similar capabilities, but have you ever tried to use the PAC Compare utility? I won't even go into it.

UnPACing...

Let's touch briefly on the fact that if you ever decided to remove your CCM tool from your NATURAL or 3GL application, **N2O** comes out as easily as it went in. PAC, on the other hand (as I am sure a number of you have first hand experience in this), does not un-install, and literally has to be 'strong-armed' into being removed. Again, this is due to the fact that PAC, in some cases, actually adds data to the NATURAL Object record (naughty), so when PAC is removed these records have to be "cleaned". Fortunately, Treehouse has strong arms and the tools to help make this "cleaning" process easy. Please contact me for more information about our special offer for an "UnPACing" package.

Understanding One's Own Product

If I was in the market to buy some new CCM software, I would have to ask myself one question "Would I buy a product (PAC) from a company that no longer has any technical expertise in that said product, or would I buy a product (**N2O**) from a company that has vast technical expertise in its product?" Not a difficult question to answer.

Sewickley, Pennsylvania or ??????

You know that **N2O** is developed and supported out of Treehouse headquarters in Sewickley, and you can receive support from our people around the world. You might expect PAC support out of Germany, but what about in North America? And, is the product still being developed, and where?

Another interesting side note is that when **N2O**'s first release was announced in March of 1989, a product claiming to do the same thing as **N2O** from Software AG was announced the same day. It was named PACS and "would do everything **N2O** does". Well, they've had 16 years...

If you would like to speak to me directly about **N2O**, or how we can help you in your CCM project, I would be glad to answer your questions. My contact information is:

Mr. Mark Jones

Senior Technical Representative

Phone: +44-(0)-1283-220-601

Fax: +44-(0)-7876-813-945

E-mail: mjones@treehouse.com •

Map it on Your PC

Treehouse Software's **tRelational** and **Data Propagation System (DPS)** deliver an extremely robust toolset to support the mapping and transformation of ADABAS data to RDBMS schema.

tRelational auto-generates complete RDBMS schema from existing ADABAS file structures and, alternatively, allows easy mapping of ADABAS fields to pre-existing data warehouse or ERP schemata. **tRelational** generates RDBMS DDL and the **DPS** Parameters. After **tRelational** does the mapping, **DPS** then delivers an efficient Extract, Transform, and Load (ETL) and Change Data Capture (CDC) engine.

tRelational is a menu-driven NATURAL application consisting of 4 subsystems:

1. ADABAS File Implementation and Analysis
2. RDBMS Modeling and Mapping
3. DPS Parameters Subsystem
4. Administrative Subsystem

tRelationalPC offers an alternative modeling environment to the Modeling and Mapping subsystem.



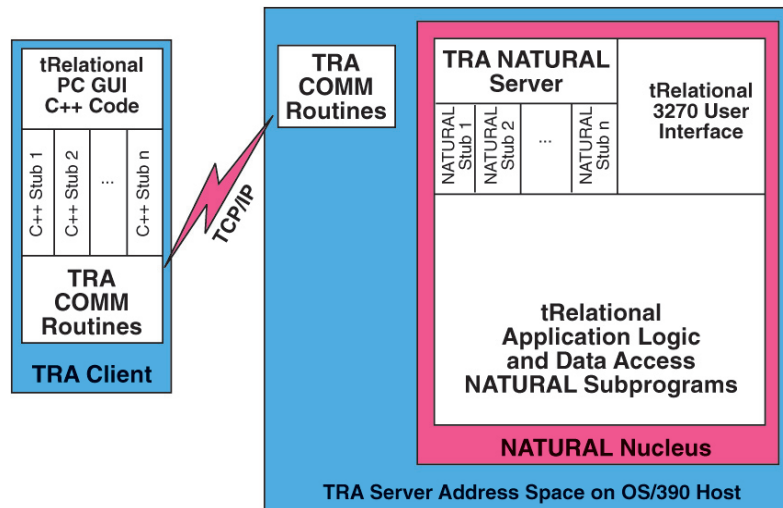
Use tRelational on your PC

Customers are very pleased to learn that **tRelational's** data modeling and mapping functions are also provided in a user-friendly Graphical User Interface (GUI), called **tRelationalPC**. This remote Windows-based client offers an instantly familiar ERD-like layout for creation and maintenance of RDBMS data models and mappings of ADABAS fields to RDBMS columns. Customers needing to map to a pre-existing schema or model may import the schema in **tRelationalPC** and easily perform the click, drag, and drop mapping.

tRelationalPC complements the **tRelational** mainframe processing in added functionality and ease of use. Additionally, customers may assign the modeling and mapping responsibilities to staff members that are not conversant in mainframe ADABAS/NATURAL skills.

How does tRelationalPC talk to tRelational on the mainframe?

TSI has developed middleware called **Treehouse Remote Access (TRA)**. **TRA** is a z/OS- (also OS/390-) based server that allows TSI clients (such as **tRelationalPC**) to invoke NATURAL subprograms that reside on the mainframe. Clients may be on any platform that supports TCP/IP communication and has network connectivity to the host. **TRA** utilizes two static TCP/IP ports and runs as a started task or batch job. **tRelationalPC** and **TRA** are included free with the **tRelational** product.



tRelationalPC communicates with **tRelational** on the mainframe via transparently-invoked Remote Procedure Calls (RPCs). When the **tRelationalPC** user selects certain remote-access functions, **tRelationalPC** transparently initiates RPC-based communications with the mainframe-based **TRA** server to execute **tRelational** mainframe-based procedures. These procedures extract or refresh data contained in the mainframe-based meta-data repository. The remote-access functions, including "Load Implemented Files", "Remote Open", and "Remote Save", can be easily initiated by using **tRelationalPC's** GUI interface.

Note: Customers (e.g., VSE sites) that cannot, or choose not to, install **TRA** can still take

advantage of **tRelationalPC**. **tRelational's** batch functions **TREUNLD/TRELOAD** and **tRelationalPC's Tools/Import-Export TRELOAD** functions support an asynchronous alternative to **TRA**. For example, the customer may execute **tRelationalPC's Tools/Import TRELOAD** (via mainframe-based batch processing), transfer the resulting file to the PC, and then execute **tRelationalPC's Tools/Import TRELOAD** function.

(continued on page 5)

Map it on Your PC *(continued from page 4)*

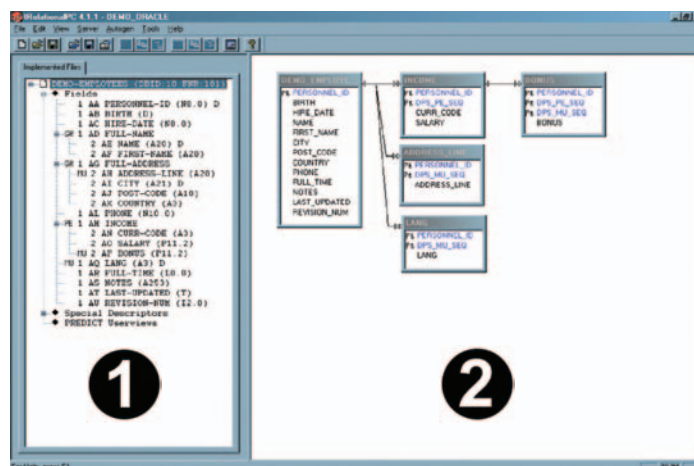
tRelationalPC Functions and Features

- Windows applications frequently offer the user alternative ways to execute a given task or function, including menus, toolbars, and the mouse “right-click”. For example, Server functions may be selected from the menu or by selecting a toolbar function.
- tRelationalPC** supports all of the tRelational mainframe modeling and mapping features. **tRelationalPC** also supports some added functionality (e.g., copy/paste a column and ODBC Import of Schema).

Note: The DEMO_ORACLE model is an auto-generated model from the DEMO_EMPLOYEES file. Notice the generated child tables derived from the ADABAS MU and PE fields, the generated Primary Keys, and the Foreign Key constraints.

As you will see on the following pages, the **tRelationalPC** interface includes standard menu/toolbar and status bar, and the workspace is divided into two parts:

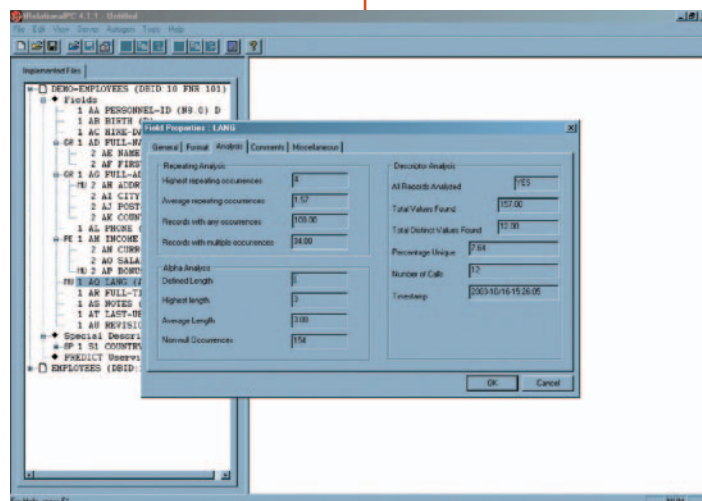
- 1 Implemented Files (left pane):** The Implemented files are retrieved from **tRelational**'s mainframe-based repository (TREDICT file) by executing the Server/Load Implemented Files request. An Implemented file (derived from PREDICT and ADABAS file/userview metadata) provides the basis for modeling and mapping to an RDBMS schema and must be retrieved prior to modeling in **tRelationalPC**. Following retrieval from the mainframe, the Implemented files are stored locally on the PC for ease of usage in subsequent tRelationalPC sessions.
- 2 Model (right pane):** A **tRelational** Model is a physical RDBMS schema containing tables, columns, and constraints with explicit ADABAS field to RDBMS column mappings. An auto-generated model may be retrieved from **tRelational**'s repository by executing the Server/Remote Open request. The model may then be modified and saved to a local drive or network file server as a .tre file. After the modifications are completed, the model must be written to the **tRelational** mainframe-based repository by executing the Server/Remote Save request.



Implemented File – Analysis Statistics

tRelational captures statistical analysis and summaries of the ADABAS data content for the purpose of understanding the data for modeling decision-making. The statistical analysis information stored in the **tRelational** mainframe-based meta-data repository is loaded to the client environment during the Server/Load Implemented Files processing. This information is available in **tRelationalPC** and is displayed by right-clicking an ADABAS field, selecting Properties, and then selecting the Analysis tab.

The Repeating Analysis depicts the MU/PE repeating statistics to aid in denormalization decisions, the Alpha Analysis shows the Alphanumeric field length statistics to aid in assignment of CHAR and VARCHAR datatypes to target Columns, and the Descriptor Analysis offers Descriptor/Superdescriptor statistics to help identify Primary Key and Alternate Index choices.



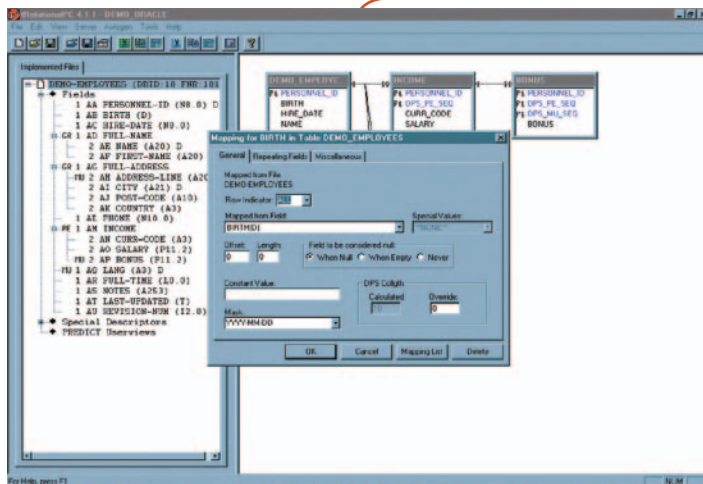
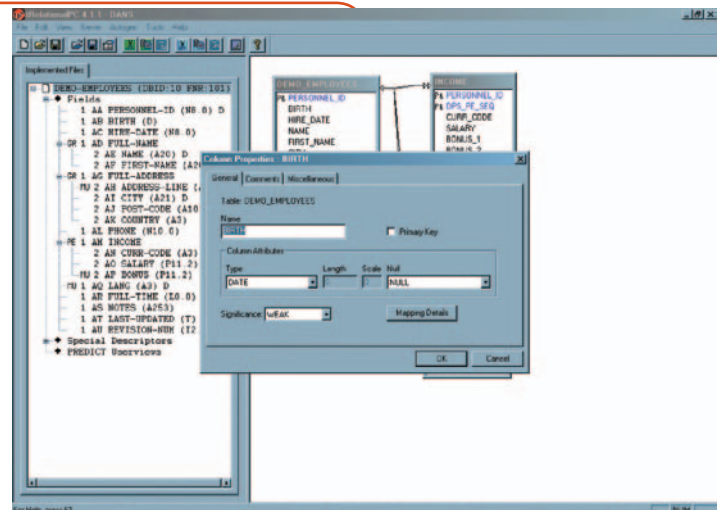
(continued on page 6)

Map it on Your PC *(continued from page 5)*

tRelationalPC Column Properties

The Column properties fully describe a Column within a target RDBMS Table. The properties are accessed when:

- The user double-clicks on the Column name.
- The user right-clicks on the Column name and chooses the “Properties” item.



tRelationalPC Mapping Details

A Mapping associates an ADABAS source field or constant values with a target Column in an RDBMS Table. The information is available by selecting a column, right-clicking, and selecting the Mapping Details. Through usage of multiple Mappings, a single column may be mapped from a number of fields, field components, or constant values (e.g., FIRST-NAME, the first character of MIDDLE-NAME, a period ('.'), and LAST-NAME may be mapped in concatenation to a column FULL_NAME). The Mapping List button displays the list of mapping detail records. Note this is equivalent to selecting the mainframe Mapping Summary option.

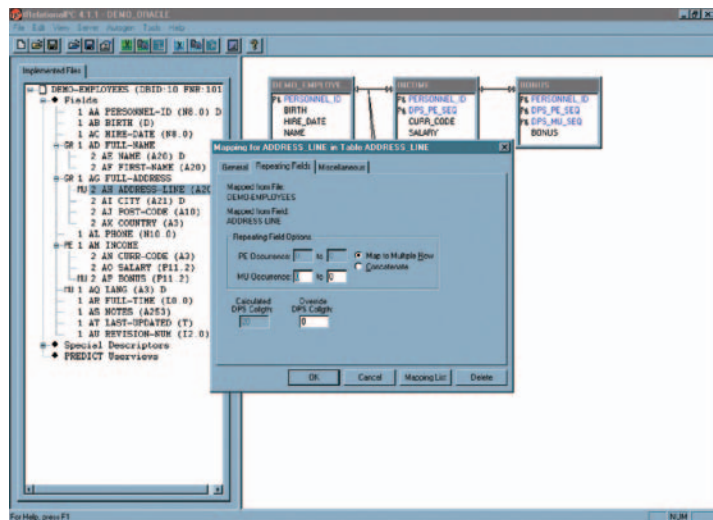
tRelationalPC Mapping Details – Repeating Fields

The Repeating Fields tab provides access to additional mapping options for ADABAS MUs and PEs. The user identifies the MU or PE occurrence or range of occurrences to be propagated from the ADABAS record to the RDBMS Column during the DPS Transformation process.

If Map to Multiple Rows is set and a range of occurrences is specified, DPS will automatically generate a new row for each non-null occurrence found in an ADABAS record.

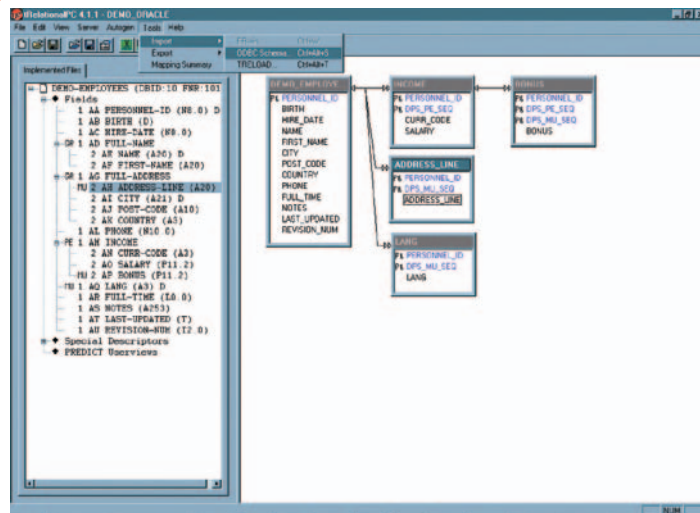
Alternatively, this feature also supports the explicit denormalization of repeating values. Repeating fields with positional significance may be mapped to distinct columns and/or tables. For example, an MU field, “ADDRESS-LINE”, could be optionally mapped (in a denormalized fashion) to the distinct columns “address_line_01”, “address_line_02”, etc.

Concatenation of repeating field values is also a standard option. For example, all occurrences of the MU field, “ADDRESS-LINE”, may be concatenated into a single column, “address”.



(continued on page 7)

Map it on Your PC *(continued from page 6)*



to specify an ADABAS file from which fields are to be mapped, the user simply drags-and-drops a Column name onto a source ADABAS field name, and the mapping detail window pops up to display the mapping record created. It's that easy!

Summary

tRelationalPC was designed to complement **tRelational** mainframe processing and to provide a Windows-based solution to model and map ADABAS file structures to RDBMS schemata. **tRelational** is not intended to compete with popular modeling products, such as ERWin, ER/Studio, Oracle Designer, PowerDesigner and Data Architect. However, the DDL generated by **tRelational** may be reverse-engineered into such modeling products and, as mentioned above, schema generated from these modeling solutions may be imported into **tRelational** for mapping purposes. The key functionality for which **tRelational** is designed is the explicit ADABAS to RDBMS mapping.

Contact Treehouse Software Inc. today to find out how **tRelationalPC**, **tRelational**, **DPS**, and **DPSync** can help your organization effectively deliver ADABAS data to your RDBMS platforms. There is no faster or more efficient solution on the market to meet your ADABAS-to-RDBMS transfer requirements. •

Import ODBC Schema

The Tools/Import/ODBC Schema feature allows existing RDBMS tables/views, columns, foreign keys, primary keys, and indexes, all with their appropriate attributes, to be converted into **tRelationalPC** Data Models. Use of this feature requires an appropriate ODBC (Open Database Connectivity) driver and definition of the ODBC Data Source on the working PC.

After requesting ODBC Schema import, the user may select any set of Tables from any accessible database, which immediately appear in the Data Model area. The mapping of the Tables and Columns to ADABAS files and fields is then a convenient drag-and-drop process. After each Table's properties are edited

*"We used **tRelationalPC** extensively to perform our data modeling for several 'canned' Oracle tables created by a third-party vendor. In particular, we used the IMPORT ODBC SCHEMA feature, which saved us a tremendous amount of time. As a result we were able to deploy a **DPS** solution in a rather short timeframe after making some minor modifications and some data transformation with the data model (also using **tRelationalPC**). Another nice feature is once you are all done you do a 'REMOTE SAVE' and all of your changes are automatically transferred to **tRelational** on the mainframe."*

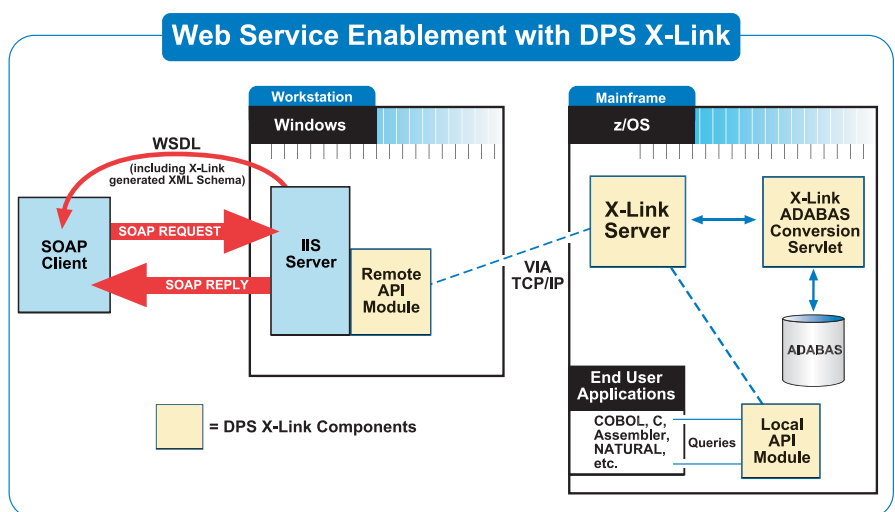
Rusty Sodegren
DBA at Penn State University

An Important Question About XML Gateway Technology...

An important question for a company that is just now investing in Gateway technology:

Do you want to invest in 90's ODBC technology or in 21st Century Web Services technology (DPS X-Link)?

Customers tell us that early-adopter companies replacing their SQL-based access with SOAP-based (i.e., XML-based) access realize returns-on-investment in short order (probably due in part to the extremely "XML friendly" nature of the Java and .NET environments). So, here's the choice: more-expensive 90's gateway technology or less-expensive state-of-the-art XML gateway technology. •



All product names mentioned in TREETIMES are trademarks and/or products of their respective holders. The mention of any non-TSI product or information provided by outside sources in TREETIMES should not be considered to imply support or endorsement by Treehouse Software, Inc., its employees, or affiliates.

Treehouse Software Products

ADABAS-to-RDBMS Data Transfer:

- DPS** - ADABAS-to-RDBMS data materialization (ETL), replication, and propagation (CDC) software
- DPS X-Link** - Instant XML-based read/write access to ADABAS
- DPSync** - Near-real-time ADABAS-to-RDBMS data propagation (CDC) software product set
- tRelational** - ADABAS modeling, mapping, and data analysis tool; **DPS** parameter generator
- tRelationalPC** - Windows-based graphical interface to make the tasks of modeling and mapping even simpler
- Treehouse Remote Access (TRA)** - Middleware that allows **tRelationalPC** to communicate with **tRelational** on the mainframe.
- NatQuery** - GUI-based tool that intelligently generates NATURAL code to handle all of the complexities of data extraction from ADABAS
- NatCDC** - Add-on to NatQuery designed to create immediately-usable data out of the ADABAS PLOG

UNIX:

- SEDIT** - XEDIT and ISPF/PDF compatible editor for UNIX and Windows
- S/REXX** - REXX-compatible language for UNIX and Windows
- S/REXX Debugger** - Optional graphical debugger for **S/REXX** programs

Software AG Related:

- ADAMAGIC** - Tool for converting mainframe ADABAS files into ADABAS for UNIX/Linux/Windows, flat file, or comma-delimited formats
- ADAREORG** - File reorganization tool for ADABAS
- ADASTRIP** - Data extraction utility for ADABAS
- AUDITRE** - Generalized ADABAS auditing facility
- AUTOLOADER** - ADABAS file automatic unload/reload/dump utility
- CHART for NATURAL** - NATURAL application analysis and documentation tool
- N2O** - NATURAL application change management system
- N2O/3GL** - 3GL support within **N2O** for PANVALET, LIBRARIAN, ENDEVOR, and PDSs
- PROFILER for NATURAL** - NATURAL quality assurance and testing tool
- SECURITRE** - ADABAS and NATURAL security interface to RACF, ACF2, and TOP SECRET
- TRIM** - ADABAS and NATURAL performance monitor

Phone: (412) 741•1677 **Fax:** (412) 741•7245 **E-mail:** tsi@treehouse.com **Web:** <http://www.treehouse.com>

TREEHOUSE SOFTWARE, INC.

409 Broad Street, Suite 140
Sewickley, PA 15143 USA

PRSRT STD
US POSTAGE
PAID
PITTSBURGH PA
PERMIT No 1715