# Tree times

A Publication of Treehouse Software, Inc.

#### This Issue

Near-Real-Time ADABAS Data Propagation with DPSync 1
Choosing the right Approach to Data Integration1
Editor's Notes 2
Instant, XML-based Access to ADABAS7

"I just wanted to let you know what a great job **Dan Sycalik** did here last week.

Everyone involved from the Tax Commission was impressed with his knowledge of the product and his teaching skills. I was extremely pleased with the amount we were able to accomplish in just a three day training session, and consider the visit a tremendous success."

#### Steve Coons

IT Manager Utah State Tax Commission (tRelational/DPS Customer)



#### *Near-Real-Time ADABAS Data Propagation with DPSync* by Joseph Brady and Wayne Lashley

Many ADABAS sites have addressed the need for transfer of data from ADABAS to their various RDBMS-based applications by means of Extract, Transform, Load (ETL) and/or Change Data Capture (CDC) products. Treehouse's **tRelational/DPS** has become the industry-leading solution by delivering the most mature,



#### Choosing the Right Approach to Data Integration by Joseph Brady

Today's enterprise must leverage existing applications in order to quickly achieve corporate strategic objectives. Many organizations have made significant investments in newer application technologies for eBusiness and other critical applications. Yet, most still have a legacy application portfolio that must be adapted to share data with the newer systems. To resolve this, the modern enterprise is often looking for data integration technology that provides:

- Rapid integration of new applications with its existing portfolio
- Integration of new business processes (e.g., eBusiness applications)
- Managed migration of legacy systems
- Business process optimization after re-structuring, take over, mergers, etc.
- Automated loading of data marts.

Generally speaking, integration technology extracts data from one or more sources,

(continued on page 2)

#### Beefing up the Treehouse Sales Presence in Germany

Treehouse welcomes VersaTec IT Services, operating out of Wörrstadt, Germany (near Frankfurt) as our new affiliate. VersaTec owner. Viktor Hubenow was with Software AG from 1987 -2003, where his position included Project Leader of various development projects, consulting to customers. and Sales Representative for medium to large accounts.

Viktor and company work closely with our European Technical Representative, **Hans-Peter Will**, who handles pre/post-sales support, forming a highly experienced ADABAS/NATURAL sales and support team. We look forward to much success in Germany!

#### Treehouse Welcomes our New Marketing Analyst

Michael Szakach has joined the Treehouse team as a Marketing Analyst. Mike is not a stranger to the Treehouse. He's been working here for the past few years during his college breaks. Now that he has graduated, earning his Bachelor's degree, he's ready to dive into all of our new initiatives, contributing in variety of areas including technical writing, research into new technologies, as well as sales and marketing support. As we ramp up our efforts in the new product areas, these types of activities are becoming more and more important. Welcome aboard Michael!

# Tree otimes

#### Editing, Writing, Design, Joseph Brady

Contributors

Wayne Lashley, Michael Ripple, Chris Rudolph, Heather Snyder, and George Szakach

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

#### Choosing the Right Approach to Data Integration

#### (continued from page 1)

manipulates it, and writes it to targets. Data integration involves the movement of data from one system to another based on commands, schedules and events. An integration product, such as Treehouse Software's **iBahn**, is able to accept data in different forms and provides a rules engine for mapping and transformation of that data.

#### **Data Integration Project Structure**

Data integration is driven by business needs that most often result in the commencement of a project. One of the fundamental purposes of such a project is to convert business needs into a technical solution. Yet most integration products are not project oriented. They simply provide technical integration without reference to the original business drivers. This can be a fatal flaw that results in IT projects taking on a life of their own, and that may not serve the best interests of the business as time goes on.

**iBahn** allows you to structure solutions that can be viewed from a business or technical perspective down to the lowest level of detail. Integration Pipes are the main unit of work in **iBahn**, and they can be defined to have a direct correlation with the business processes they support. **iBahn** allows you to specify data integration requirements in varying degrees of detail, consistent with the way that most projects progress. It is designed to provide value to a project at the onset, not just during the final development and implementation stages.

Most integration projects follow a logical progression:

#### Project Assessment:

Working with the customer's key stakeholders, both business and technical, to determine the scope of the engagement, roles of the team members, project schedules, and critical success factors.	<b>iBahn</b> provides immediate access to information sources to assess the nature and complexity of the information sources.
Project Definition: In conjunction with IT and user project team members, mapping out the desired operational environment, the required functionality, and the detailed project plan.	<b>iBahn</b> provides both the Business Workflow and the Application View. The Business Workflow captures the business transactions as a foundation of the data integration solution. The Application View defines the information sources.
Solution Design: Designing a solution to meet the specified goals, approach, and technical requirements, as well as planning its implementation.	<b>iBahn</b> provides a Network View that identifies the location, data structure, and access method for each information sources. The Integration Steps (technical integration) are defined for each of the Integration Pipes. The specifics of the Integration Steps (e.g., StepMap) do not need to be configured at this time.
Solution Implementation: Continuing to work in concert with the customer's project team, developing the solution using proven methods that ensure feedback and flexibility.	<b>iBahn</b> 's Integration View provides a GUI interface to define the details of the Integration Steps. Drop down menus and drag-and-drop configuration make the integration process simple to configure and change as required.

(continued on page 3)



December 2003

-2-

#### Choosing the Right Approach to Data Integration (continued from page 2)

Today's enterprise is built upon a "Business Workflow" or a set of business processes that define the interaction between business applications or information sources. These business processes vary in size and context based on the integration requirements (e.g., claim reporting vs. a division breakdown report, order processing vs. item inventory, or payroll vs. weekly commission update).

A business workflow generally includes:

- 1. Business Processes: comprised of one or more related Business Transactions.
- Business Transactions: atomic actions that can be described logically or independent of the IT infrastructure.
- Integration (Unit) Pipes: a set or series of Integration Steps that define the physical movement or transformation of data. In iBahn, Business Transactions

Tree **otimes** 

define logical integration and Unit Pipes define physical integration.

Typically, a business process contains a logically related group of business transactions. These transactions require access among independent systems, applications, or information sources, and they can be described independently of the IT infrastructure.

Business transactions do not lend themselves to partial completion. For example, when updating a customer record, partial completion will not usually make sense. However, submitting an order may include a series of steps such as credit checks, shipping selection, and tax calculations that allow it to be defined as a set of logical steps. In this example "Update Customer Record" might be a Business Transaction whereas Submit Order might be a Business Transaction or a Business Process.

December 2003



#### (continued on page 7)

-3-

#### Near-Real-Time ADABAS Data Propagation with DPSync (continued from page 1)

To meet these customer's needs, Treehouse is now delivering a high-performance, scalable, near-real-time Change Data Capture system that is comprehensive and robust, uses minimal CPU resources and can be implemented in hours. We are pleased to introduce **DPSync**.

#### The Technical Challenge



Some shops have turned to middleware and/or ADABAS Stored Procedures and Triggers (SPaTs) in an attempt to implement nearreal-time CDC. These approaches are programmingand intensive place an unacceptable demand on ADABAS and other system resources, driving up the cost

and complexity of the technology architecture. Issues of transaction backouts, two-phase commit and rollback, and platform-to-platform transaction semaphoring are difficult to resolve, and these detract from solution stability and data integrity.

There can be no question that the use of PLOG data is the safest, most reliable, and most efficient basis for ADABAS CDC. The PLOG is a complete, reliable record of all changes that have occurred in ADABAS, and ADABAS writes data to the PLOG as part of its normal housekeeping.

To avoid contention with ADABAS itself (a hallmark of **tRelational/DPS** product architecture), processing of the PLOG data must await an automatic or manually-invoked PLOG dataset switch and subsequent ADARES PLCOPY. Many organizations that would like to take advantage of the efficiency of this approach view the wait time as an inhibitor to implementation.

One option for capturing updates quicker is to switch the PLOG very frequently. This imposes a small but measurable workload on ADABAS, but it also can create an operational issue. For most shops, PLOG copies form the basis for recovering ADABAS databases to a recent point in the event of a catastrophic system failure. Therefore, PLOG copies are often maintained on tape devices and the tapes are periodically transported offsite. More frequent switching of the PLOG drives increased numbers of tapes, which is not acceptable in many cases. Therefore, using the PLOG copy itself is often not a viable approach to capturing changes in near-real-time.

Another option might be to devise a way to capture and transmit PLOG data to **tRelational/DPS** using ADABAS user-exits. This would require a significant programming effort, compounded with a potential maintenance issue as new versions of ADABAS are implemented. Furthermore, a robust and fault-tolerant surrounding architecture is required to accommodate log capture overflow and temporary capture failure. Most importantly, the direct impact on ADABAS must be absolutely minimized or the

system performance will be compromised.

The challenge does not end at capturing log data. The data must be decompressed, interpreted, filtered, evaluated, and transformed according to the mapping relationships between the ADABAS sources and the RDBMS targets. This requires an efficient and scalable extraction and transformation engine that allows rules to be defined, and standard and user-defined transformations to be implemented. Furthermore, a means is required to do the initial synchronization between ADABAS and the RDBMS, as well as periodic refreshes when data structures change.

To facilitate the analysis and design of ADABAS-to-RDBMS data mapping, there is a need for an analysis and modeling toolset that provides insight into ADABAS structures and data usage; supports flexible mapping of ADABAS-oriented concepts and features (PEs/MUs, record types, etc.) to native RDBMS concepts and features (RDBMS-specific datatypes, primary keys, and referential integrity constraints); and delivers productivity and flexibility through automatic schema generation and a metadata repository.

Add these up, and for most sites, the required development (and ongoing support) is an extremely complex technical task, or the cost to implement exceeds any reasonable budget, or the time to implement is unacceptably long--or, most likely, all three.



#### **DPSync Meets the Challenge**

**DPSync** addresses the above technical challenges with vendor-supported software that is mature and best-of-breed.

The critical issue of PLOG data capture and transmission is handled by technology designed expressly for that purpose. With minimal impact on the ADABAS database, PLOG data blocks are captured as they are written, and passed downstream to other **DPSync** components. This fault-tolerant architecture has minimal impact on the ADABAS database, and includes automatic and robust

handling of data gaps (e.g., when the capture is interrupted temporarily) and data spills (e.g., when high-volume ADABAS activity temporarily exceeds the capacity of the receiving process). In addition, **DPSync** contains flexible implementation configuration options to meet the needs of multiple-database, multiple-application, and multiple-CPU scenarios. For PLOG data capture, **DPSync** uses technology from **E-Net Corporation** called **Remote Recovery Data Facility (RRDF)**. Analysis, modeling, and mapping of ADABAS source data to RDBMS tables is accomplished in **DPSync** through the **tRelational** 

(continued on page 5)



#### Near-Real-Time ADABAS Data Propagation with DPSync (continued from page 4)

component. This mainframe-based NATURAL application includes:

- A metadata repository implemented in ADABAS
- A high-productivity modeling environment
- The ability to report and reconcile ADABAS structure differences between PREDICT and Field Definition Table (FDT)
- Analysis of usage of repeating structures (MUs/PEs)
- Analysis of usage of alphanumeric fields
- Analysis of usage of descriptors and superdescriptors
- Automatic generation of target schema and mappings for Oracle, DB2, SQL Server, Sybase, Teradata, and other targets
- Modeling of any combination of normalization and denormalization of repeating structures

Modeling and mapping activities may also be undertaken using a native

Windows application that provides a GUI, drag-and-drop environment and the ability to import existing RDBMS schemata.

High-performance, flexible, and scalable processing of PLOG data to create native-syntax RDBMS SQL is delivered through **Data Propagation System (DPS)**. **DPS** selects, filters, and transforms ADABAS sources, and includes:

- A built-in library of standard transformations
- Completely flexible mapping options
- Extensible architecture for user-defined transformations
- Operator console interface
- Statistical reporting
- Automatic detection and response to database management events (e.g., file restores and refreshes, field definition changes)
- Initial-load/periodic-refresh (ETL or materialization) capability using static input (ADASAV backup), imposing no workload on ADABAS
- Native formatting of materialization output for RDBMS load utilities, including utility control/format files
- Ability to produce delimited or fixed-length flat-file outputs, including optional "action code"
- Optional-use ADABAS PLOG Consolidation (APC) feature reduces output SQL volume



These core components comprise a complete architecture for near-real-time ADABAS-to-RDBMS data propagation. The **DPSync** toolset also includes optional components that can facilitate process automation and robustness.

**George Szakach**, President of Treehouse Software answers some questions about near-real-time ADABAS data propagation with **DPSync**.

#### George, how fast does DPS process PLOG data?

Both **DPS** and **DPSync** process PLOG data quickly and are robust in capability. **DPS** is more than adequate for a number of sites that require ADABAS data propagation nightly or every hour or more. **DPS** processes the PLOG data in non-real-time (i.e., after the dual PLOG switch is made and the filled log is copied by an ADABAS utility to another dataset).

#### What if the data is needed at the RDBMS site quicker?

Unfortunately, the required PLOG switches and copies necessary to be made in order to run **DPS** more often than hourly can be operationally intolerable at some sites. This is why **DPSync** was created. **DPSync** can be viewed as "running **DPS** with mini-PLOGs with switches invisibly happening practically instantly, hundreds of times a day". These are not actual PLOG switches but are PLOG data captures, courtesy of the **RRDF** component of **DPSync**. Along with this PLOG data capture, normal PLOG activity will occur, and switches and copies are done for normal ADABAS operations. But the ADABAS data changes that

(continued on page 6)



-5-

#### Near-Real-Time ADABAS Data Propagation with DPSync (continued from page 5)

are necessary to send to the RDBMS get captured rapidly by the RRDF component and are immediately sent to **DPS** for processing.

### Some sites tout writing data extract programs in NATURAL. Is this feasible?

In a few ADABAS site cases, there may be minimal (one or two) simple ADABAS files that are necessary to be converted one time into relational format. For small, simple files to be processed one time (no propagation of changes needed), it may be reasonable to write NATURAL or COBOL programs to do extractions to flat files. This process effectively freezes the ADABAS data, unless the process is to be done often. But this process directly accesses ADABAS, which affects production, and therefore cannot or should not be done often.

Once a site has determined that it is necessary for multiple ADABAS files (and/or complex ones) to be converted, even if only one time, this may possibly result in hundreds or thousands of relational tables to be created and populated. It becomes a costly venture to write, and a logistical problem to maintain, these conversion programs. Simpleappearing changes in ADABAS data format and related RDBMS schemata can radically affect the programs. CPU consumption will obviously become a concern.

#### How about those who want to do propagation of ADABAS changes (i.e., a one-time conversion will not suffice, and multiple-time conversions are not practical)?

There are several non-glamorous (manual-like) ways to do this. One way would be to change the ADABAS applications to not only update the database but to also write an "update log", which can then be later processed. This would be done for each NATURAL, COBOL, or other application program that updates the concerned ADABAS files. Assuming one has access to these source programs, this would be costly, time-consuming, error-prone, and a maintenance problem.

#### If all the ADABAS updates are on a PLOG, why wouldn't a site just write programs to process PLOGs and propagate data on their own?

First, the average programmer would not be able to write a program to go against the extremely complex, "compressed" data in an ADABAS PLOG. Even experienced NATURAL programmers should not attempt to do the bit twiddling necessary to decompress a PLOG. The process really requires a heavy-duty assembler program. Even if they could write these programs, the result would be that there are many such programs all needing to be run against the same PLOGs. That's why DPS, and now DPSync, were created, so that users all over the world do not have to concern themselves with PLOG data. Meanwhile, yes, there are several "programs" on the market that process PLOGs, including TRIM, AUDITRE, PLEU, and ADASEL "utilities". However, these utilities only go so far as to make selected PLOG data readable or further processable (i.e., they "flatten" the data). Therefore, it is necessary to write and maintain post-processing programs, maybe many of them, to process this flattened data to make it into the INSERTs. DELETEs and UPDATEs that must go against the possibly

many relational tables. And, with any appreciable amount of data, this two-step process would be prohibitively slow.

With **DPSync**, one system processes the PLOG data directly and sends the update data to the RDBMS in nearreal-time. **DPSync** provides a consistent framework for transferring ADABAS data to the relational system. Additional ADABAS files and related relational tables can be added to the system easily. Development time is saved. CPU cycles are saved. The cost savings over any other approach are evident. Having the ADABAS data echoed nearly immediately in the RDBMS can have a radical positive affect on the plan, design, coding, and maintenance of the applications on the RDBMS.

#### Near-Real-Conclusion...

With the **DPSync** automated, near-real-time ADABAS data transfer method, you will benefit from:

- a consistent framework for transferring ADABAS data to an RDBMS
- trust that the data is transferred correctly and completely
- no need to write and maintain programs to transfer data
- fewer operational issues (no messy PLOG switching)
- ability to easily map ADABAS files to resultant relational tables
- ease of incorporating additional ADABAS files and related RDBMS tables into the mix
- minimal impact on the production ADABAS system
- minimized CPU consumption
- overall cost savings
- time-proven professional service from Treehouse Software
- time-proven product reliability from Treehouse Software and E-Net Corporation

#### Special Announcement:

## Basic Web Enablement Open Seminar!

#### Who should take this seminar?

ANYONE who is currently in IT and needs to understand and be able to participate in either web enabling an existing legacy system or in creating a new web enabled, cross platform application.

At the end of this seminar, the participant will be able to understand and put into practice the necessary skills to web enable an application, such as:

Create and maintain web pages.

- Re-engineer legacy systems for a web user interface.
- Understand the various forms of web pages and their uses.
- Understand what XML is and why it's used.
- Understand CGIs and the use of middle ware.
- Use Natural to write an application using the web as the user interface - from the mainframe!

See the web site at www.accesstrhodes.com/OpenSeminars.html for more information on this and other seminars, or contact Tanya Rhodes @ (800) 659-2951.



custome

Treehouse

for

discount

%01

#### Instant XML-Based Access to ADABAS

As the XML revolution continues, organizations are recognizing a need to extract data and metadata from legacy systems and transform it into appropriate XML formats for purposes of XML-based EDI processing, Web Services, web-enablement of applications, and porting of data from/to native XML databases.

Unfortunately, to date, legacy users have been left to code their own solutions.

We at Treehouse Software feel that this is simply *not good enough*, and we are delivering **DPS X-Link**, an extension of **Treehouse's Data Propagation System (DPS)**, the industry leading ADABAS-to-RDBMS data extraction and transformation tool.



**DPS X-Link** provides instant XML-based access to mainframe ADABAS data and metadata, and completely automates the processes of legacy-to-XML data and metadata extraction and transformation. For more information on **DPS X-Link**, please contact Treehouse Software.

#### Choosing the Right Approach to Data Integration (continued from page 3)

#### Example:

Consider the case of retail franchise stores. The stores operate somewhat independent of the headquarters, and each has their own POS, inventory and other systems. In order to protect the franchiser's reputation for quality, several functions are centralized, such as purchasing and finance. Additionally, to provide better customer service, the franchise locations have the ability to locate inventory at other local stores.

Without automated data integration, the interaction between the stores and headquarters would be a time-consuming process, requiring manual communications and constant



attention. With automated data integration, the inventory and POS systems could be integrated, keeping the tracking of inventory levels current. And inventory levels at other store locations would be readily accessible. Pricing information would always be current. Payment history and order status could be obtained without calling support personnel. Franchises often operate on small margins that can be significantly improved by streamlining overhead processes like the ones just described.

Let's explore another example -- that of adding a new product for sale. In this case, our business process requires a number of steps as follows:

- The product must be added to the master catalog.
- A new UPC code may need to be added.
- A new product category may need to be added if this is a new business area.
- Pricing information will need to be added.
- Starting inventory levels will need to be established.

Each of these pieces of information might be managed by a separate application and included in a separate database table or file. In fact, a number of steps must happen, but all steps are related as part of a overall business requirement. **iBahn** allows you to define this set of updates as part of a cohesive business transaction (e.g., AddNewProduct) rather than a collection of autonomous reads and writes.

With **iBahn**, the integration steps can be defined according to the way a business is run. For example, the integration steps could be organized by store or functional area. Messages on **iBahn**'s Operator Monitor could also be grouped accordingly. This allows support personnel to verify successful operations and to quickly identify and resolve issues if they arise. Due to a business unit view, the progress of integration steps can visually

be monitored and checked. **iBahn** is designed to be an integral part of your project, not just an implementation technology. Business needs are tied to technical solutions. The product carries that business orientation into your production environment, providing multiple views of the solution (Business Workflow, Application View, Network View, Integration View), as well as a real-time monitor to view and manage processes when required. **iBahn** is built to match your business process.

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



-7-

# Tree • times

#### Treehouse Software products include:

#### Data Integration:

**iBahn** - Integration suite that connects data within the enterprise or between business partners ADABAS-to-RDBMS Data Transfer:

**DPS** - ADABAS-to-RDBMS data materialization (ETL), replication, and propagation (CDC) software DPS X-LINK - ADABAS-to-XML extension of DPS 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. 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: **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 N20 - NATURAL application change management system N20/3GL - 3GL support within N20 for PANVALET, LIBRARIAN, ENDEVOR, and PDSs **PEEK** - ADABAS file browsing utility **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