Converting 4GL Databases and Apps to a Relational Database

Joe Elliott
Nationwide Insurance

Thursday - August 11
Session Number (9636)
Reasons to Convert 4GL Apps

- Proprietary Data Formats
  - Segmented
  - Unusual Relational Formats
  - Flat Files
- Expensive Vendor Licensing
- Limited Internal Technology Expertise
- Nonstandard Software Solution
Justifying the Cost of Replacement

• Demonstrate the Ease of the Solution
• Break Effort Into Bite Size Pieces
• Use Support Budget as Funding
• Cost of Supporting Nonstandard Solutions
Develop a Plan for Data Extraction

• Revisit Business Requirements
• Update Nonfunctional Requirements
• Survey the User Community
• Plan for Export and Backup of Existing Data
Exporting Data from Proprietary Source

- Determine Common Indexes
- Run Reports to Count Related Data Segments
- Design New Relational Tables and Relationships
- Run Exports
Data Cleanup

- Review Exported Data for Format Errors
- Cleanup Missing Data and Invalid Columns
- Watch for Duplicate Rows in the Export Process
- Multiple Approaches to Correct Data Errors
Relational Database Design

• Leverage Existing Resources if Possible
• Using Index Keys from 4GL Source Files
• Include Lookup Tables
• Consider All Interfaces in old 4GL Application
Rebuilding the User Interface from 4GL

- Must Update Business Requirements
- Consider Multiple Approaches (BI tools, dev tools)
- Solution Fit into Technology Standards
- More Complex than Building Matching Entry Forms
Testing the New Solution

• Start with Simple Reporting Against New Database
• Involve the End User Early in the Process
• Show Benefits of Change Up Front in Testing
• User Acceptance Testing with Management Signoff
Implement

• Develop a Strong Backout Plan
• Involve the End User Early in the Process
• Show Benefits of Change Up Front in Testing
• User Acceptance Testing with Management Signoff
Demonstration

- Export of IBI WebFOCUS Database
- Loading of Relational Database
- Replacing Existing User Interface
Summary

Questions?