



Analysis of Existing ABL

Dr. Thomas Mercer-Hursh

VP Technology
Computing Integrity, Inc.

Let me begin by introducing myself. I began working with Progress in 1984 and I have been a Progress Application Partner since 1986. For many years I was the architect and chief developer for our ERP application. In recent years, I have refocused on the problems of transforming and modernizing legacy ABL applications. To transform an application, one must first understand the application. That, and a long history of modifying existing systems is the background for today's workshop.

We are going to cover a lot of ground today. This means that I am going to point to a lot of possibilities and try to give you an idea of how it might be useful to you, but I'm not going to go into detail on any one tool. Hopefully, I can make you aware of what is possible and you can apply this to your own work.



Agenda

- Introduction: What Is the Problem?
- Program Structure Tools for One Compile Unit
- Program Structure Tools for Multiple Compile Units
- Search and Compare Tools
- Documentation Tools
- Parser Tools
- UML Tools
- Summary

So, here's our agenda for today. First we are going to talk a bit about the problem and why it is important and then we are going to go through some, certainly not all, of the tools in each of a number of different categories.



Agenda

- Introduction: What Is the Problem?
- Program Structure Tools for One Compile Unit
- Program Structure Tools for Multiple Compile Units
- Search and Compare Tools
- Documentation Tools
- Parser Tools
- UML Tools
- Summary

First, let's talk a little about why analysis is important.



Introduction

Knowing the desired behavior for a program modification is only part of the problem. One also needs to know where to change the behavior and the impact of making that change.

In mature systems, it is common for a “simple” change to cause unexpected consequences which take more work to fix than the original change ... not to mention other consequences.

Knowing the desired behavior for a program modification is only part of the problem. One also needs to know where to change the behavior and the impact of making that change.

In mature systems, it is common for a “simple” change to cause unexpected consequences which take more work to fix than the original change ... not to mention other consequences.



Introduction

Good analysis is good risk management.

Good analysis is like a surgeon knowing where to cut and knowing where not to cut.

Good analysis is understanding before doing.

Not doing thorough analysis means **unexpected results**, inevitably causing longer cycles to make changes.

Good analysis is good risk management.

Good analysis is like a surgeon knowing where to cut and knowing where not to cut.

Good analysis is understanding before doing.

Not doing thorough analysis can mean unexpected results and much longer cycles to make changes.

Q Introduction

Tell me your story!

6 Analysis of Existing ABL © 2011 Computing Integrity

At this point in the workshop, participants were asked:

- Version
- OEA
- Existing tools
- One paragraph about why you are here



Agenda

- Introduction: What Is the Problem?
- Program Structure Tools for One Compile Unit
- Program Structure Tools for Multiple Compile Units
- Search and Compare Tools
- Documentation Tools
- Parser Tools
- UML Tools
- Summary



COMPILE LIST

- Illustrates what's in the program with includes in-line.
- Resolves pre-processor code.
- Shows scope of transactions and buffers.
- Built-in to ABL compiler.

COMPILE LIST

Illustrates what's in the program with includes in-line

Resolves pre-processor code (i.e. code which is evaluated before compile, e.g. operating system dependent code)

Shows scope of transactions and buffers

Built-in to ABL compiler

The attendees looked at (in the Cloud):

Project IS in PCA2011

Compilelist

Run

Look at output in OpenEdge/WRK

.lst

Expanded listing

Transaction scope at end

Buffer scope at end.



COMPILE XREF

- Shows index usage of each line which references an index.
- Shows table and field access by line.
- Shows string references and other less frequently useful information.
- Built-in to ABL compiler.

COMPILE XREF

Shows index usage of each line which references an index.

Shows table and field access by line.

Shows string references and other less frequently useful information.

Built-in to ABL compiler.

Cloud exercise included:

.xrf

Access, index



Program Structure for One Compile Unit

XML XREF

- Shows index usage, table and field access by line (same as COMPILE XREF) but in XML form:
 - Easier to parse into database.
 - Harder for human to read.
- Built-in to OpenEdge Architect.

XML XREF in OpenEdge Architect

Shows index usage, table and field access by line (same as COMPILE XREF) but in XML form:

Easier to parse into database

Harder for human to read

Cloud:

Xml xref

Same info in more easily processed form for databasing



Agenda

- Introduction: What Is the Problem?
- Program Structure Tools for One Compile Unit
- Program Structure Tools for Multiple Compile Units
- Search and Compare Tools
- Documentation Tools
- Parser Tools
- UML Tools
- Summary



Database of XREF Data

- XREF data of each compile unit is loaded into a database which allows querying, e.g. of all places where a table or index is used.
- Many people have built XREF databases.
- XREF databases exist in several frameworks.
- No standard implementation.

Database of XREF Data

- XREF data of each compile unit is loaded into a database which allows querying, e.g. of all places where a table or index is used.
- Many people have built XREF databases.
- XREF databases exist in several frameworks.
- No standard implementation.



Super XREF

- XREF database based on Proparse.
- Much richer set of information could be included.
- Proposed open source tool at <http://www.oehive.org/node/1112>
- Possible development this summer.

Super XREF

XREF database based on Proparse

Much richer set of information could be included.

Proposed open source tool at

<http://www.oehive.org/node/1112>

Possible development this summer.



Agenda

- Introduction: What Is the Problem?
- Program Structure Tools for One Compile Unit
- Program Structure Tools for Multiple Compile Units
- Search and Compare Tools
- Documentation Tools
- Parser Tools
- UML Tools
- Summary



Search and Compare Tools

OpenEdge Architect

- Search and replace can be performed in current file, workspace or selected resources (directory, files, CTRL + click, etc.).
- Has problem shared by non-ABL tools where search is performed on string patterns, not ABL syntax (e.g. search for string will find in comment or variable name).

OpenEdge Architect

Search and replace can be performed in current file, workspace or selected resources (directory, files, CTRL + click, etc.).

Has problem shared by non-ABL tools where search is performed on string patterns, not ABL syntax (e.g. search for string will find in comment or variable name).

Ctrl-F and Ctrl-H in OEA



Non-ABL-specific Search Tools

- Many tools are available.
- Share problem of being unaware of ABL syntax.
- No need to be in OpenEdge Architect (i.e. good for Version 9 or before).
- Able to search a body of code without creating a project in OpenEdge Architect (e.g. search entire codebase).

Non-ABL-specific Search Tools

- Many tools are available.
- Share problem of being unaware of ABL syntax.
- No need to be in OpenEdge Architect (i.e. good for anyone Version 9).
- Able to search a body of code without creating a project in OpenEdge Architect (e.g. search entire codebase).

Other tools nice interface, but still not ABL specific



Search and Compare Tools

substitute

- Script uses Unix “sed” to do mass changes on selected files.
- Can be run in preview mode to insure that changes will be as expected.
- Output illustrates before/after.
- Uses unix scripts (need unix extension to run on Windows).
- Command and scripts are available to anyone who wants them, just ask.

substitute

- Command uses unix “sed” to do mass changes on selected files.
- Can be run in preview mode to insure that changes will be as expected.
- Output illustrates before/after.
- Uses unix scripts (need unix extension to run on Windows).
- Command and scripts are available to anyone who wants them, just ask.

Substitute – look at example

Scripts available



Search and Compare Tools

File and Directory Comparison Tools

- Many tools available which compare 2 different versions of programs.
- Preferred are tools which can compare 3 items (programs or directories). Handy when base program has been modified by 2 programmers and need to figure out how to bring them together again.
- Araxis Merge <http://www.araxis.com/merge/>
- Possibly: KDiff3 <http://kdiff3.sourceforge.net/>

File and Directory Comparison Tools

- Many tools available which compare 2 different versions of programs.
- Preferred are tools which can compare 3 items (programs or directories). Handy when base program has been modified by 2 programmers and need to figure out how to bring them together again.
- Araxis Merge <http://www.araxis.com/merge/>
- Possibly KDiff3 <http://kdiff3.sourceforge.net/>
(no experience with this one yet)

Demo was performed from own laptop compare IS/rc/rcal with /tmp and show program



Agenda

- Introduction: What Is the Problem?
- Program Structure Tools for One Compile Unit
- Program Structure Tools for Multiple Compile Units
- Search and Compare Tools
- Documentation Tools
- Parser Tools
- UML Tools
- Summary



AutoDox2

- Commercial product available from Joanju:
<http://joanju.com/autodox2/index.php>
- Parses code and annotations

This:

```
/** Submit a PrintDocument to this printer.  
 * @param printDocument The document to print.  
 * @return The print job ID.  
 */  
METHOD PUBLIC INT64 addToQueue(printDocument AS CLASS PrintDocument):
```

becomes:

```
PUBLIC INT64 addToQueue(PrintDocument printDocument)
```

Submit a PrintDocument to this printer.

Parameters:

printDocument - The document to print.

Returns:

The print job ID.

Commercial product available from Joanju:
<http://joanju.com/autodox2/index.php>

Parses code and annotations

Cloud

Click link

explore



AutoDox2

- Creates html pages which can be browsed, reducing searches through code.

The screenshot shows a web browser window with a navigation menu on the left and a main content area. The navigation menu includes links for 'All Classes', 'Packages', 'Print', 'Print.HP', and 'Sales'. The main content area has tabs for 'Overview', 'Package', and 'Unit', with 'Overview' selected. Below the tabs, the text 'Tiny example code for AutoDox2' is displayed. A table titled 'Packages' lists three items: 'Print', 'Print.HP', and 'Sales', each with a brief description.

Packages	
Print	Provides services for printing, print previews, and managing print jobs.
Print.HP	Provides HP printers.
Sales	Master records and documents for sales and quotations.

Creates html pages which can be browsed, reducing searches through code.

Cloud

Click link

explore



AutoDox2

[Overview](#) [Package](#) [Unit](#)

Package Print

Provides services for printing, print previews, and managing print jobs.

All Compile Units

PrintDocument.cls	<i>INTERFACE</i> Declare print services for a Document which may be printed.
Printer.cls	Represents a printer on the network.
printerManager.p	Manage the printer master table.

Package Description

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Package relationships and dependencies

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita



AutoDox2

Overview	Package	Unit
--------------------------	-------------------------	----------------------

Print
CLASS Printer

CLASS Print.Printer

Represents a printer on the network.

Author:
John Green August 2009

Method Summary

INT64	addToQueue (PrintDocument printDocument)
	Submit a PrintDocument to this printer.
VOID	removeFromQueue (INT64 jobId)
	Remove a job from this printer's queue.

Method Detail

addToQueue

```
PUBLIC INT64 addToQueue(PrintDocument printDocument)
```

Submit a PrintDocument to this printer.

Parameters:
printDocument - The document to print.

Returns:
The print job ID.



Agenda

- Introduction: What Is the Problem?
- Program Structure Tools for One Compile Unit
- Program Structure Tools for Multiple Compile Units
- Search and Compare Tools
- Documentation Tools
- Parser Tools
- UML Tools
- Summary



Parser Tools

Parsers analyze code in same way compiler does, recognizing and resolving:

- tokens
- keywords
- table names
- field names
- etc.

Parser Tools

Parsers analyze code in same way compiler does, recognizing and resolving:

- tokens
- keywords
- table names
- field names
- etc.



Proparse

- Parses code and creates abstract symbol tree in memory.
- Open source product created and updated by John Green (Joanju).
- Available at <http://www.joanju.com/proparse/index.php>

Proparse

- Parses code and creates abstract symbol tree in memory.
- Open source product created and updated by John Green (Joanju).
- Available at <http://www.joanju.com/proparse/index.php>



ProLint

- Uses Proparse to apply wide variety of code quality tests which help:
 - Avoid errors
 - Impose shop standards
- Open source product created by Jurjen Dijkstra with many contributors.
- Available at <http://www.oehive.org/prolint/download>

ProLint

- Uses Proparse to apply wide variety of code quality tests which help:
 - Avoid errors
 - Impose shop standards
- Open source product created by Jurjen Dijkstra with many contributors.
- Available at <http://www.oehive.org/prolint/download>

Demo ProLint on laptop



Proparse Scripting

- ABL code can be written which uses Proparse to perform special tasks outside of Proparse's purview, e.g. find all input and output statements in a set of code (finds all places where code interacts with external files).
- Searches are ABL syntax aware.
- Write it yourself!
- Sample code expected to be published soon on OpenEdge Hive.

Proparse Scripting

- ABL code can be written which uses Proparse to perform special tasks outside of Proparse's purview, e.g. find all input and output statements in a set of code (finds all places where code interacts with external files).
- Searches are ABL syntax aware.
- Write it yourself!

Demo scripting locally



Joanju Analyst

- Uses Proparse, reads all code, builds database.
- Commercial tool available from Joanju at:
<http://joanju.com/analyst/index.php>
- Data base includes:
 - Connections between run statements, methods, procedures.
 - Connections to database tables (to field level).
 - Dynamic call resolution (uses combination of automatic analysis and hints when unable to resolve calls).

Joanju Analyst

- Uses Proparse, reads all code, builds database.
- Commercial tool available from Joanju at:
<http://joanju.com/analyst/index.php>
- Data base includes:
 - Connections between run statements, methods, procedures.
 - Connections to database tables (to field level).
 - Dynamic call resolution (uses combination of automatic analysis and hints when unable to resolve calls).

Watch videos at:

<http://joanju.com/analyst/index.php>



Joanju Analyst (con't)

- Html browser allows one to follow links:
 - From the run statement to the code that is run.
 - From an internal procedure or method to all the places which call it (where used)..
- Very flexible, ABL syntax aware search tool.
- Produces Bill of Materials output (XML) which drives ABL2UML.
- Used for productivity, impact and flow analysis, debugging and re-engineering.

Joanju Analyst (con't)

•Html browser allows one to follow links:

- From the run statement to the code that is run.
- From an internal procedure or method to all the places which call it (where used).

•Very flexible, ABL syntax aware search tool.

•Produces Bill of Materials output (XML) which drives ABL2UML.

•Used for productivity, impact and flow analysis, debugging and re-engineering.

Talk and watch videos



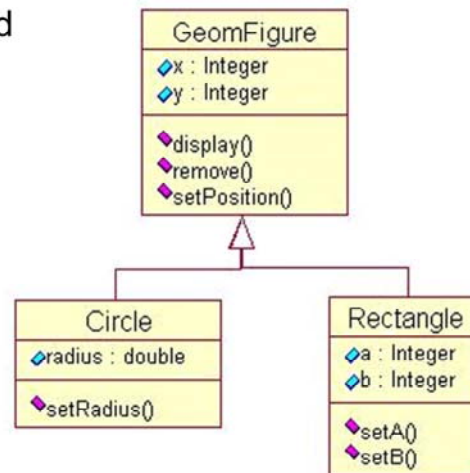
Agenda

- Introduction: What Is the Problem?
- Program Structure Tools for One Compile Unit
- Program Structure Tools for Multiple Compile Units
- Search and Compare Tools
- Documentation Tools
- Parser Tools
- UML Tools
- Summary



UML (Unified Modeling Language)

- Predominant method used in OO programming.
- Used differently: as a sketch, a detailed analysis, or to generate code.
- Created in mid-1990s to unify a diverse set of modeling systems.



Among those emphasizing analysis and design, the strongly predominant way of expressing that design is UML (Unified Modeling Language).

UML was created in the mid-1990s to unify a diverse set of modeling languages which had grown up, primarily for OO development. A standards body, the OMG or Object Management Group was created to oversee this and other standards and UML has undergone considerable expansion and development since the original version.

Different people use UML in different ways. Some use it simply as a sketching tool, something to put on a white board or in a document to facilitate discussion. Some will use it more completely to do a detailed analysis of a system and then write code from that design. They may or may not keep the design in sync with the code as the system evolves, although it is usually regrettable if they don't. And, there are those ... which is what interests us today ... who actually generate the working code directly from the model.



Enterprise Architect

- Commercial tool <http://www.sparxsystems.com/>
- Most favored in ABL community because of vendor support for using OpenEdge database as repository and multiple supporting tools.
- OE datatypes available:
 - Dr. Thomas Mercer-Hursh
<http://www.oehive.org/node/1073>
 - Phil Magnay
<http://communities.progress.com/pcom/docs/DOC-6208>

Enterprise Architect

Commercial tool <http://www.sparxsystems.com/>

Most favored in ABL community because support for using OpenEdge database as repository and supporting tools.

OE datatypes available:

Dr. Thomas Mercer-Hursh <http://www.oehive.org/node/1073>

Phil Magnay <http://communities.progress.com/pcom/docs/DOC-6208>



ABL2UML

- Open source tool created in 2007 by Dr. Thomas Mercer-Hursh <http://www.oehive.org/ABL2UML>
- Takes schema from database and Bill of Materials from Analyst and builds UML Component diagram with code units, links and connections, database tables and fields.
- Contains diagram builder to automatically and flexibly build UML diagrams at any level of detail starting with any compile unit, table, field.

ABL2UML

- Open source tool created in 2007 by Dr. Thomas Mercer-Hursh
- Takes schema from database and Bill of Materials from Analyst and builds UML component diagram which has
 - all code units down to internal procedures, function, method level (detail level)
 - all links to detail level
 - summary of compile unit connections
 - database tables and fields
 - connections between code units (including tables and fields)
 - how/when table/field is read, modified or written
 - all where clauses which connect code
- Contains diagram builder to automatically and flexibly build UML diagrams at any level of detail starting with any compile unit, table, field.



ABL2UML Status

- Revision of tool currently under discussion by Mike Fechner, David Abdala and Dr. Thomas Mercer-Hursh.
- Goals:
 - Move to OO
 - Incremental builds
 - Use pieces separately, e.g., schema only
 - Support non-OpenEdge databases
 - Support .df alternative to direct schema

ABL2UML Status

Revision of tool currently under discussion by Mike Fechner, David Abdala and Dr. Thomas Mercer-Hursh.

Goals:

Move to OO

Incremental builds

Use pieces separately, e.g., schema only

Support non-OpenEdge databases

Support .df alternative to direct schema

And more!



Agenda

- Introduction: What Is the Problem?
- Program Structure Tools for One Compile Unit
- Program Structure Tools for Multiple Compile Units
- Search and Compare Tools
- Documentation Tools
- Parser Tools
- UML Tools
- Summary



Summary

Let's talk more about what brings you here.

Go back over issues and review how it applies and what else might be done.
Get ideas from audience.



Summary

There are lots of tools available. Some built-in, some free open source, some commercial – most are inexpensive. Not knowing is far more expensive.

A small investment in tool building, purchase, and/or learning will pay dividends year after year.

Would you perform surgery with a blindfold?

AutoDox2 starts at \$490/seat up to \$3900 for 10+ site license

Analyst starts at \$1050/seat up to \$8500 for 10+ site license



For More Information, go to...

Program Structure Tools for one Compile Unit

- **COMPILE LIST** – Built-in to ABL compiler
- **COMPILE XREF** – Built-in to ABL compiler
- **XML XREF in OpenEdge Architect** – Built-in to OpenEdge Architect

Program Structure Tools for Multiple Compile Units

- **Database of XREF data** – Site/Framework Specific; no standard implementation
- **“SuperXREF”** – Proposed Open Source tool at <http://www.oehive.org/node/1112>

Here are some links for more information. Generally, look at OpenEdge Hive, Joanju.com, and Cintegrty.com, my own website.



For More Information, go to...

Search and Compare Tools

- **OpenEdge Architect Search** – Built-in to OpenEdge Architect
- **Non-ABL Specific Search and Index Tools** – many available
- **substitute** – scripts available on request
- **Compare tools** – many available, but note Araxis Merge <http://www.araxis.com/merge/>

Documentation Tools

- **AutoDox2** – Commercial available at <http://joanju.com/autodox2/index.php>



For More Information, go to...

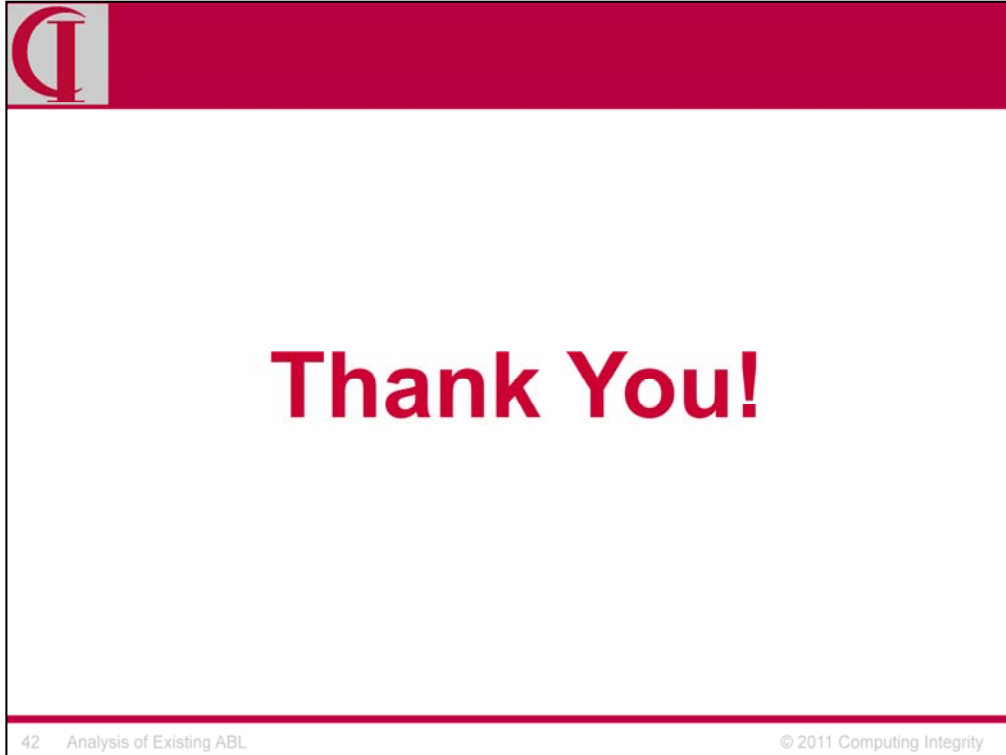
ABL Parser Tools


- **Proparse** – Open source at <http://joanju.com/proparse/index.php>
- **ProLint** – Open source at <http://www.oehive.org/prolint/download>
- **Proparse scripting** – Write it yourself
- **Analyst** – Commercial available at <http://joanju.com/analyst/index.php>

UML Tools

- **Enterprise Architect** – Commercial available at <http://www.sparxsystems.com/>
- **ABL2UML** – Open source available at <http://www.oehive.org/ABL2UML>

Here are some links for more information.





Thank You!

42 Analysis of Existing ABL

© 2011 Computing Integrity

Thank you.



Questions ?

For more information:

<http://www.cintegrity.com>

thomas@cintegrity.com

510-233-5400

And now for questions.