



Analysis Problems in ABL and How To Solve Them

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 presentation.

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?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- 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 broad categories of analysis problems and what tools there are to solve them.



Agenda

- Introduction: What Is the Problem?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- 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 and bugs 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 understanding before doing.

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

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



Agenda

- Introduction: What Is the Problem?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- Summary

The simplest analysis problem in ABL is finding out basic information about a specific compile unit which is the current focus of attention.



What's Going On in This Compile Unit?

COMPILE LIST

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

File Name	Line	Blk. Type	Tran	Blk. Label
...amples\src\test.p	0	Procedure	No	
...amples\src\test.p	20	Do	No	OUTSIDE-BLOCK
...amples\src\test.p	22	For	Yes	
Buffers: Sports.bCustomer				

COMPILE LIST

Built-in to ABL compiler

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



What's Going On in This Compile Unit?

COMPILE XREF

- Built-in to ABL compiler.
- 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.

```
...src\test.p ...src\test.p 21 REFERENCE Sports.Customer  
...src\test.p ...src\test.p 22 ACCESS Sports.Customer Name  
...src\test.p ...src\test.p 22 SEARCH Sports.Customer Name WHOLE-INDEX  
...src\test.p ...src\test.p 23 ACCESS Sports.Customer Name  
...src\test.p ...src\test.p 23 ACCESS Sports.Customer Name  
...src\test.p ...src\test.p 23 UPDATE Sports.Customer Name
```

COMPILE XREF

Built-in to ABL compiler.

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.



What's Going On in This Compile Unit?

```
<Reference Object-identifier="Sports.Customer" Reference-type="SEARCH">
  <Source-guid>1F8qct1Ezo3gEf/88GLimg</Source-guid>
  <File-num>1</File-num>
  <Ref-seq>8</Ref-seq>
  <Line-num>22</Line-num>
  <Object-context>Name</Object-context>
  <Access-mode/>
  <Data-member-ref/>
  <Temp-ref/>
  <Detail>WHOLE-INDEX</Detail>
  <Is-static>>false</Is-static>
  <Is-abstract>>false</Is-abstract>
</Reference>
```

XML XREF

- Built-in to 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.

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



Agenda

- Introduction: What Is the Problem?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- Summary

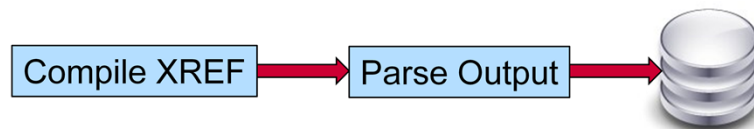
In addition to wanting to know about a single compile unit, one often wants to know about bigger issues like where a particular table or routine is accessed throughout the code base, e.g., when one needs to make a change and wants to see the impact.



What's Going On in This Body of Code?

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.
- I am currently working on an open source offering in this area.



What's Going On in This Body of Code?

Roundtable TSMS

In association with Configuration Management, databases XREF data to allow inquiry on:

- Object relationships.
- Object references and type of reference.
- The context of the reference.
- In-depth searching (with plug-in).
- Properties of referenced objects.

This is for both schema and code objects. See <http://tugboatsoftware.com/products/roundtable/>

Exchange Session: Roundtable Enterprise: It's Not Just for OpenEdge by Jeff Ledbetter

The closest thing to a “standard” implementation which includes an XREF database of this type is the Roundtable TSMS Software Configuration Management system. It is a commercial product and does a lot more than just provide the XREF data. They also store searchable information like annotations, properties, methods, internal procedures, shared variables, etc.



Agenda

- Introduction: What Is the Problem?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- Summary

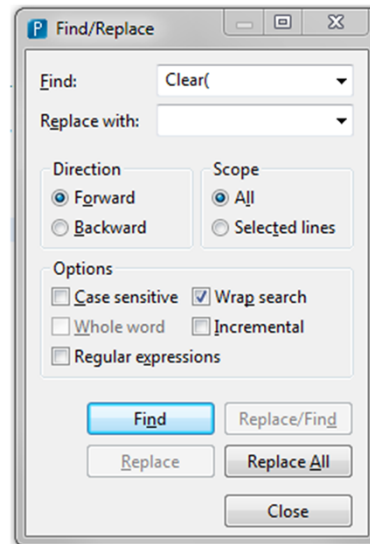
Another common analysis task is to find references to a table, procedure, variable, etc. in the code. A related task is to compare two or more sets of code to determine what the differences are between them.



Where Is That? or What's Different?

OpenEdge Architect

- Ctrl-F: Search and replace in current file.



OpenEdge Architect

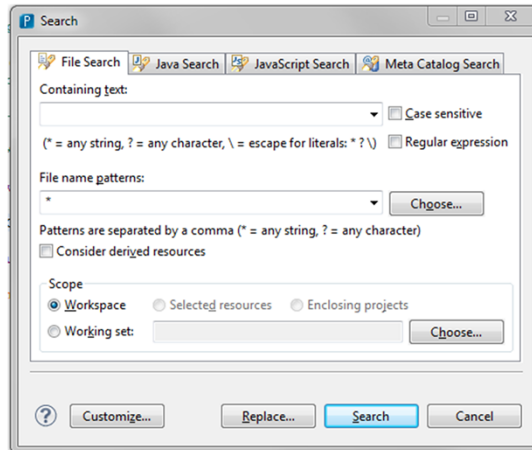
Search and replace can be performed in current file, workspace or selected resources (directory, files, CTRL + click, etc.).

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



Where Is That? or What's Different?



OpenEdge Architect

- Ctrl-H: Search and replace on workspace or selected resources (directory, files, CTRL + click, etc.).

OpenEdge Architect

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



Where Is That? or What's Different?

OpenEdge Architect

- 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).

E.g., consider a search and replace for Customer with bfCustomer:

```
/* Total all orders for the Customer */
```

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



Where Is That? or What's Different?

Non-ABL-specific Search Tools

- Many tools are available.
- Share problem of being unaware of ABL syntax.
- Can be used with Version 9 and earlier code (i.e., no need for OpenEdge Architect).
- 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.
- Can be used with Version 9 and earlier code (i.e., no need for OpenEdge Architect).
- 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



Where Is That? or What's Different?

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)



Where Is That? or What's Different?

Demo of Araxis Merge

See slide at end. Very quick illustration of finding differences in two trees.

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



Agenda

- Introduction: What Is the Problem?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- Summary

Sometimes, the best way to do analysis is to record design decisions when the code is created in the first place because then you can just read how something works without having to figure it out from scratch. While comments provide some assistance here, one can do much better.



How Does That Work Anyway?

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 “formerly” available from Joanju:
<http://joanju.com/autodox2/index.php>

Parses code and annotations

Handles classes and procedures

Does not handle anything beyond 10.2B



How Does That Work Anyway?

SmartDox

- Product to be available from Consultingwerk and Riverside Software.
- Based on two new tasks in PCT.
- Parses code and annotations.
- Produces HTML documentation.
- Applicable only to OO code.

Exchange Session: Automated Class Reference Generation by Mike Fechner

- Product to be available from Consultingwerk and Riverside Software.
- Based on two new tasks in PCT.
- Parses code and annotations.
- Produces HTML documentation.
- Applicable only to OO code. Gilles might extend PCT tasks to .ps later.

Exchange Session: Automated Class Reference Generation by Mike Fechner

Q How Does That Work Anyway?

SmartDox Demo

23 Analysis Problems in ABL and How to Solve Them © 2013 Computing Integrity

Need some samples and more information from Mike.

Demo results from
http://help.consultingwerkcloud.com/smartcomponent_library/trunk/index.htm
|



Agenda

- Introduction: What Is the Problem?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- Summary

So far, we have been looking only at static analysis, i.e., the code sitting in a file on a disk. Some problems, though, are only easily identified when the code is running. Other than putting in message or log statements, there are several tools which can help us understand running code.



When It Runs, What Happens?

Debugger

Allows controlled execution and monitoring of a running program instance:

- Breakpoints – fixed and computed.
- Stepped execution.
- Variable values.
- Object status.

Standard part of both Windows and Unix distributions.



When It Runs, What Happens?

Debugger

This breakpoint:

```
chLogName = "CollectionPerformance_Log.txt".          /* Start New Log */  
output to value(chLogName).  
output close.
```

Allows seeing current variable values:

Name	Value
chLogName	"CollectionPerformance_Log.txt"
obExceptionStack	?
lgGotError	no
ABase	?
MyTestAETFSet	?
MyTestAOSetTT	?
MyTestAOSetSA	?
MyIterator	0
deTime	0



When It Runs, What Happens?

Profiler

Collects execution times on blocks of code so that one can evaluate performance. Returns:

- Line numbers.
- Number of repetitions.
- Average time.
- Cumulative time.

Use `COMPILE DEBUG-LIST` for correct line numbers.

The profiler is built into standard ABL, but is not documented or officially supported. There is a document at the URL shown which briefly documents a GUI tool for using the profiler. See also `$DLC/src/samples/profiler` depending on the installation choices you have made. There is a Knowledgebase entry documenting the options for the profiler object at http://knowledgebase.progress.com/articles/Article/19495?q=profiler+handle&l=en_US&fs=Search&pn=1



When It Runs, What Happens?

Profiler (cont.)

To use:

- -profile run time parameter. Profiles whole session, but does not require modifying code.
- A GUI tool which can be found for running selected procedures (<http://communities.progress.com/pcom/docs/DOC-2808>)
- The built-in profiler object which allows selective execution on selected blocks of code, but does require modifying the code.
(http://knowledgebase.progress.com/articles/Article/19495?q=profiler+handle&l=en_US&fs=Search&pn=1)

To use:

- -profile run time parameter. Profiles whole session, but does not require modifying code.
- A GUI tool which can be found at <http://communities.progress.com/pcom/docs/DOC-2808> or in `$DLC/src/samples/profiler` depending on the installation options chosen.
- The built-in profiler object which allows selective execution on selected blocks of code, but does require modifying the code. This is documented in a Knowledgebase entry at http://knowledgebase.progress.com/articles/Article/19495?q=profiler+handle&l=en_US&fs=Search&pn=1



When It Runs, What Happens?

Test Program

```
1 define variable i as integer no-undo.  
2  
3 assign  
4   profiler:enabled = yes  
5   profiler:profiling = yes  
6  
7  
8 do i = 1 to 1000000:  
9 end.  
10  
11 i = 0.  
12 do while i < 1000000:  
13   i = i + 1.  
14 end.  
15  
16 i = 0. do while i < 1000000: i = i + 1. end.  
17  
18 assign  
19   profiler:enabled = no  
20   profiler:profiling = no  
21  
22  
23 profiler:write-data().  
24  
25 return.
```

Profiler

Sample Output

Line	Reps	Avg. Time	Cumm. Time
0	1	0	7.177015
9	1000000	0.050981	0.050981
8	1000001	0.117987	0.117987
11	1	0.000004	0.000004
13	1000000	1.443773	1.443773
12	1000001	1.158929	1.158929
14	1000000	0.051977	0.051977
16	1	0.000001	0.000001

Here is a very simple example using the built-in Profiler object.



When It Runs, What Happens?

LOG-MANAGER System Handle

Provides a reliable method to collect data during execution without interrupting the execution process with **MESSAGE** statements and with modes for systematic collection of many events and transitions without coding.

To help you understand the output, see

- <http://www.oehive.org/project/Profiler>
- <http://www.oehive.org/project/log-manager-reporter>



When It Runs, What Happens?

ProTop

Provides real time monitoring of many aspects of database performance, thus creating insight into possible coding problems that may not be apparent from examining the code itself.

Obtain from <http://dbappraise.com/protop.html>



Agenda

- Introduction: What Is the Problem?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- Summary

Improving code quality isn't analysis, per se, but can make it less likely to encounter problems in production and make it easier to understand code during maintenance. Moreover, some of the tools used for code quality improvement can also be used for certain analytical issues. There are many behavioral approaches to code quality, but today I am focused specifically on tools so I am going to look at tools which can read and understand the syntactic structure of the code.



What Is Parsing?

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

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

What Is Parsing?

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

- tokens
- keywords
- table names
- field names



How Do I Get Better Code Quality?

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>
- Is no longer maintained and has no support for any syntax past 10.2B.

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>



How Do I Get Better Code Quality?

ProLint

- Uses Proparse to apply wide variety of code quality tests which help avoid errors and impose shop standards.
- Open source product created by Jurjen Dijkstra with multiple contributors.
- Rules are written in ABL, so easily modified.
- Available at <http://www.oehive.org/prolint/download> .
- Has the same version limitation as Proparse.

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>



How Do I Get Better Code Quality?

Proparse Scripting

- Uses Proparse.
- 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 your own scripts to suit your individual need.
- Has the same version limitation as Proparse.

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!



How Do I Get Better Code Quality?

Joanju Analyst

- Based on Proparse, reads all code, builds database.
- Commercial tool available from Joanju at: <http://joanju.com/analyst/index.php> .
- Same version issues.

Joanju Analyst

- Uses Proparse, reads all code, builds database.
- Commercial tool available from Joanju at:
<http://joanju.com/analyst/index.php>



Joanju Analyst (cont.)

- 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).

- 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).



How Do I Get Better Code Quality?

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.

Analyst is no longer marketed, but can be available by special arrangement. See me if interested.

Watch videos?



Agenda

- Introduction: What Is the Problem?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- Summary

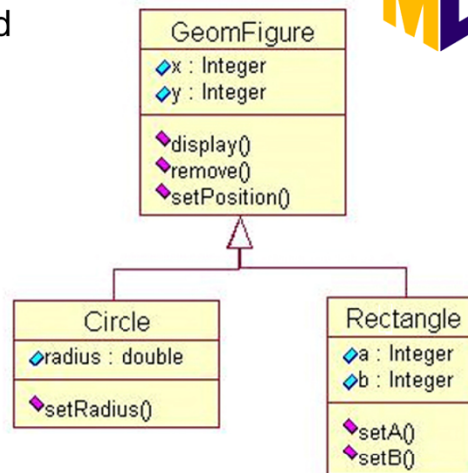
Tools discussed so far provides a utilities to help explore the code, allow tracing some feature of the code, or examines how the code behaves. But, what if one is looking for the big picture, how it all fits together, the structure of the interaction of the code. For this, we want *pictures!*



UML (Unified Modeling Language)



- Predominant method used in OO programming.
- Used in multiple ways: 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.



UML Tools

ABL2UML

Demo

44 Analysis Problems in ABL and How to Solve Them © 2013 Computing Integrity

ABL2UML

- Open source tool created in 2007 by Dr. Thomas Mercer-Hursh
- Takes schema from database and Bill of Materials from Analyst and builds UMLcomponent 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 way.
- 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?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- Summary



PCT and Jenkins

At Exchange: Beginners Guide to Continuous
Integration by Gilles Querret



Sonic Source

At Exchange: Buttonhole Gilles Querret at previously mentioned talk.



Agenda

- Introduction: What Is the Problem?
- What's Going On in This Compile Unit?
- What's Going On in This Body of Code?
- Where Is That? or What's Different?
- How Does That Work Anyway?
- When It Runs, What Happens?
- How Do I Get Better Code Quality?
- Understanding The Big Picture
- Looking Forward
- Summary




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?



Thank You!

51 Analysis Problems in ABL and How to Solve Them © 2013 Computing Integrity

Thank you.



Questions ?

For more information:

<http://www.cintegrity.com>

thomas@cintegrity.com

510-233-5400

And now for questions.



Slide Equivalents for Demos

Use of slides vs demos to be determined by timing later



Where Is That? or What's Different?

File and Directory Comparison Tools

File Name	Count
AOSetSA.cls	2
AOSetTT.cls	2
iSet.cls	1

Slide equivalent of demo.



How Does That Work Anyway?

SmartDox Demo

Slide equivalents to be built.

Need some samples and more information from Mike.

Demo results from

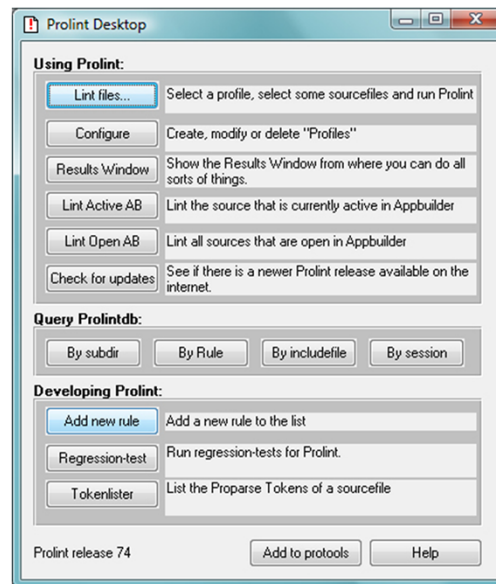
http://help.consultingwerkcloud.com/smartcomponent_library/trunk/index.htm

|



How Do I Get Better Code Quality?

ProLint Desktop



Demo ProLint on laptop



How Do I Get Better Code Quality?

ProLint Results

autoedge/server/autoedge/bedealer.p	112	Possibly group ASSIGN with line 110	groupassign	5
autoedge/server/autoedge/bedealer.p	78	#202: Node RETURN has less indent than IF on line 76. Expected to be 4, is 2.	ifindent2	2
autoedge/server/autoedge/bedealer.p	26	Compile will fail on Unix, use only lower-case includefiles: support/beEntity.i	lowercase	9
autoedge/server/autoedge/beEntity.i	46	Compile will fail on Unix, use only lower-case includefiles: support/extendProcedure.i	lowercase	9
autoedge/server/autoedge/bedealer.p	27	Compile will fail on Unix, use only lower-case includefiles: support/proExceptionStart.i	lowercase	9
autoedge/server/autoedge/bedealer.p	67	OUTPUT PARAMETER eError CHARACTER scope LOCAL getDealerId should start with "o"	nameconv	5
autoedge/server/autoedge/beEntity.i	38	CLOSE in CLOSE is a Progress keyword	nameconv	5
autoedge/server/autoedge/beEntity.i	38	ON CLOSE scope LOCAL CLOSE should start with "T"	nameconv	5
autoedge/server/autoedge/bedealer.p	131	INPUT PARAMETER dsDealer scope LOCAL eDealerModifyPreTrans should start with "T"	nameconv	5
autoedge/server/autoedge/bedealer.p	144	INPUT PARAMETER dsDealer scope LOCAL eDealerCreatePreTrans should start with "T"	nameconv	5
autoedge/server/autoedge/bedealer.p	156	INPUT PARAMETER dsDealer scope LOCAL eDealerValidation should start with "T"	nameconv	5
autoedge/server/autoedge/dsdealer.i	26	dsDealer scope MAIN should start with "g"	nameconv	5
autoedge/server/autoedge/etdealer.i	22	TEMPTABLE eDealer scope MAIN should start with "g"	nameconv	5
autoedge/server/autoedge/bedealer.p	137	PROCEDURE eDealerCreatePreTrans scope LOCAL eDealerCreatePreTrans should start with "T"	nameconv	5
autoedge/server/autoedge/bedealer.p	124	PROCEDURE eDealerModifyPreTrans scope LOCAL eDealerModifyPreTrans should start with "T"	nameconv	5
autoedge/server/autoedge/bedealer.p	150	PROCEDURE eDealerValidation scope LOCAL eDealerValidation should start with "T"	nameconv	5
autoedge/server/autoedge/bedealer.p	29	PROCEDURE fetchWhere scope LOCAL fetchWhere should start with "T"	nameconv	5
autoedge/server/autoedge/beEntity.i	62	FUNCTION getAccessProc scope LOCAL getAccessProc should start with "T"	nameconv	5
autoedge/server/autoedge/bedealer.p	57	PROCEDURE getDealerId scope LOCAL getDealerId should start with "T"	nameconv	5
autoedge/server/autoedge/bedealer.p	81	PROCEDURE getDealerSiteID scope LOCAL getDealerSiteID should start with "T"	nameconv	5
autoedge/server/autoedge/beEntity.i	81	PROCEDURE getValidation scope LOCAL getValidation should start with "T"	nameconv	5
autoedge/server/autoedge/bedealer.p	107	VARIABLE F hAccessProc HANDI F scope LOCAL saveChannels should start with "T"	nameconv	5



Proparse Scripting

Sample calling program is simply:

```
using proparse-scripts.*.
```

```
def var finder as class FindIO no-undo.  
run proparse/setup/setup.p.  
finder = new FindIO("autoedge").  
finder:ParseAllFiles().
```

FindIO is a simple 60 line ABL which checks for INPUT and OUTPUT verbs and creates report.

Demo scripting locally



How Do I Get Better Code Quality?

Proparse Scripting

Searches 209 programs under “autoedge” in about a minute and produces output like:

```
C:\Work\tmp\PPScript\autoedge\web\escript\src\web\objects\psagent.p
Line: 120
OUTPUT STREAM WebStream TO "WEB":U.
```

```
C:\Work\tmp\PPScript\autoedge\web\escript\src\web\objects\psagent.p
Line: 193
OUTPUT STREAM WebStream TO "WEB":U CONVERT TARGET WEB-CONTEXT:HTML-
CHARSET.
```

```
C:\Work\tmp\PPScript\autoedge\web\escript\src\web\objects\spooler.p
Line: 111
/* Open the file for reading */
input from value (file-info:full-pathname) binary no-map no-convert.
```

Demo scripting locally



How Do I Get Better Code Quality?

Joanju Analyst (con't)

Powerful query capability

The screenshot displays the Joanju Analyst web interface. It features a search bar with a 'Submit Query' button and a 'Search Help' link. Below the search bar is a section titled 'Browse a compile unit' with a text input field containing 'src/web/objects/psagent.p' and a 'Submit Query' button. A mouse cursor is positioned over the input field. To the right of the input field is the text 'Enter the name of a .p, .w, or .cls file on the proppath.' At the bottom of the interface, there are three links: 'Build Manager', 'Administrative Page', and 'Log file'.

Browse a specific compile unit ...



How Do I Get Better Code Quality?

Joanju Analyst (con't)

Source is fully pre-processed and “pretty printed” with collapsible includes, links to original source, and many other features. One can click on a run

```
do:
  output stream WebStream to "WEB":U.
  run init-cgi in web-utilities-hdl.
  run init-request in web-utilities-hdl.
  init-request in stateaware.p
  init-request in devpath.p
  init-request in runlog.p
  init-request in web-util.p
  init-request in session.p
  ping :U then "webutil/ping.p":U else ( if AppProgram eq
    AppProgram ) )
  "debug":U then "webutil/debug.p":U else ( if AppProgram eq
    AppProgram ) )
  am, output cFileExtn ) no-error.
  tn ) then
  undo.
  close
```

and see all possible destinations based on the value of web-utilities-hdl and can go to that source.



How Do I Get Better Code Quality?

Joanju Analyst (con't)

One can also click on the definition of an internal procedure or function and see a list of all places where that code is referenced and link to that source..

```
function devCheck returns logical ( :  
input "srvrAppMode":U ) begins "Dev".  
devpath.p  
psagent.p
```



How Do I Get Better Code Quality?

Joanju Analyst (con't)

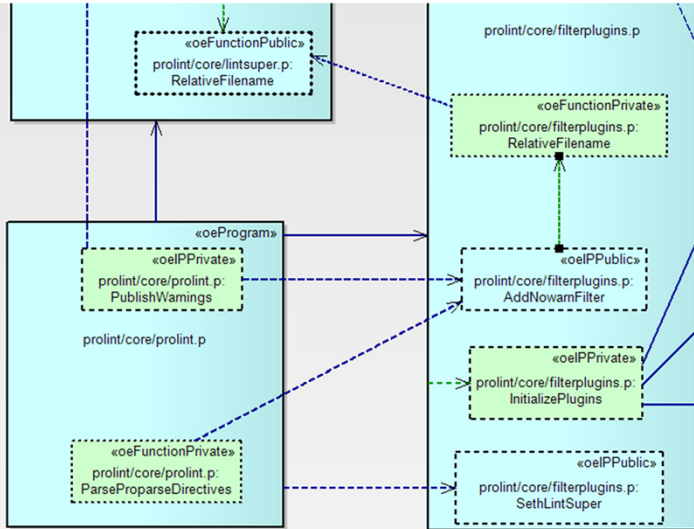
One can also query on includes, table and field references, calls, and ABL syntax.

The screenshot displays the Joanju Analyst web interface. It features three main sections:

- Joanju.Analyst**: The main title of the tool.
- Search**: A section with a search input field containing the query `tables:customer -(fields:"customer.customerID")`. To the right of the input is a [Search Help](#) link. Below the input is a **Submit Query** button.
- Browse a compile unit**: A section with an input field containing the path `src/web/objects/psagent.p`. To the right of the input is the instruction "Enter the name of a .p, .w, or .cls file on the propath." Below the input is a **Submit Query** button.
- Server**: A section with three links: [Build Manager](#), [Administrative Page](#), and [Log file](#).

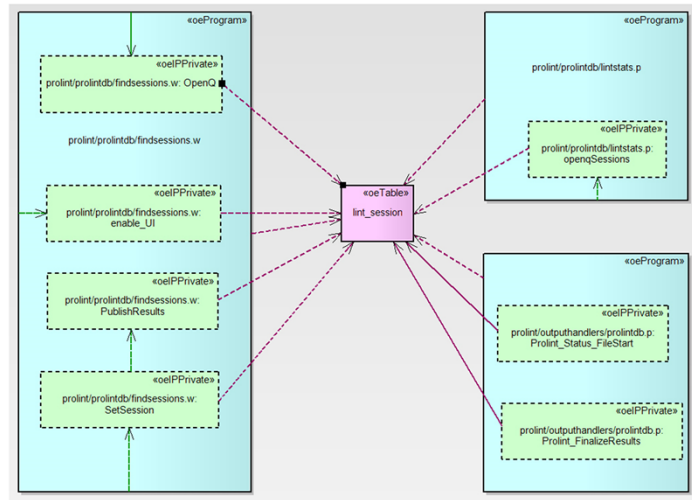


ABL2UML
Sample showing
links between
IPs, functions,
and compile
units.





ABL2UML
Sample showing
links between
IPs, functions,
and compile
units to a table.





Consider whether to update the following slides and add after Summary or just to go with the per slide links.



For More Information, go to...

What's Going On in This 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

What's Going On in This Body of Code?

- **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 my own website, Cintegrity.com.



For More Information, go to...

Where Is That? or What's Different?

- **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/>

How Does That Work Anyway?

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



For More Information, go to...

ABL How Do I Get Better Code Quality?

- **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.