Taking your COBOL apps to .NET and the Java Virtual Machine

Scot Nielsen
Product Manager
Moving to managed code

• What is managed code?
• Different perspectives on why deploy to .NET or JVM?
• Some lessons learned
Terminology

- Managed code = .NET or JVM
- Native code or unmanaged = everything else
What is Managed code

• Virtual execution environment
  – Portable Intermediate Language
  – JIT

• Garbage collection
• Object-oriented
• Extensive frameworks/Class Libraries
• COBOL code generated directly to byte code
# Why managed code

**An industry perspective**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>Java</td>
</tr>
<tr>
<td>3</td>
<td>C++</td>
</tr>
<tr>
<td>4</td>
<td><strong>Objective-C</strong></td>
</tr>
<tr>
<td>5</td>
<td>C#</td>
</tr>
<tr>
<td>6</td>
<td>Visual Basic</td>
</tr>
<tr>
<td>7</td>
<td>PHP</td>
</tr>
<tr>
<td>8</td>
<td>Python</td>
</tr>
<tr>
<td>9</td>
<td>Perl</td>
</tr>
<tr>
<td>10</td>
<td>Ruby</td>
</tr>
<tr>
<td>11</td>
<td>JavaScript</td>
</tr>
<tr>
<td>12</td>
<td>Visual Basic .NET</td>
</tr>
<tr>
<td>13</td>
<td>Delphi/Object Pascal</td>
</tr>
<tr>
<td>14</td>
<td>Lisp</td>
</tr>
<tr>
<td>15</td>
<td>Logo</td>
</tr>
<tr>
<td>16</td>
<td>Pascal</td>
</tr>
<tr>
<td>17</td>
<td>Transact-SQL</td>
</tr>
<tr>
<td>18</td>
<td>Ada</td>
</tr>
<tr>
<td>19</td>
<td>PL/SQL</td>
</tr>
<tr>
<td>20</td>
<td>Lua</td>
</tr>
</tbody>
</table>

How popular is managed code?

www.tiobe.com
Why managed code
reuse what already works
Why managed code
21st century user interfaces
Why managed code
Application Integrity

Micro Focus COBOL runtime error:
114 Attempt to access item beyond bounds of memory (Signal 11)
Application Integrity

Quick Demo

Try/Catch Exception handling in COBOL

(if you would like a demo please feel free to contact us using the following e-mail address: Harris.hartsiotis@microfocus.com)
Why managed code
End-to-end debugging

- Debug C#/Java and COBOL together
- End-to-end testing
- Single IDE for building and testing
- Single debug process
- Same developer
Why managed code
A developer’s perspective

Reusable Frameworks

Quick demo:
Date/time arithmetic

(if you would like a demo please feel free to contact us using the following e-mail address:
Harris.hartsiotis@microfocus.com)
Object Oriented Programming

- .NET/JVM are fundamentally built on OO principles
- Procedural semantics still apply
- Visual COBOL extensions
- Samples browser
Performance
### Suite Name/Chart

**General_ussteel_il_32**

<table>
<thead>
<tr>
<th>Time</th>
<th>Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>50,000</td>
<td>nwb-cben-wv21</td>
</tr>
<tr>
<td>40,000</td>
<td>nwb-cben-wv22</td>
</tr>
</tbody>
</table>
| 30,000 | \n| 20,000 | \n| 10,000 | \n| 0 | \n
**Benchmark Data**

<table>
<thead>
<tr>
<th>ussteel</th>
<th>time</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>running time</td>
<td>0000005301 0000003016</td>
<td>38.468</td>
</tr>
</tbody>
</table>

More detail?

### Suite Name/Chart

**General_ussteel_il_64**

<table>
<thead>
<tr>
<th>Time</th>
<th>Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>50,000</td>
<td>nwb-cben-wv21</td>
</tr>
<tr>
<td>40,000</td>
<td>nwb-cben-wv22</td>
</tr>
</tbody>
</table>
| 30,000 | \n| 20,000 | \n| 10,000 | \n| 0 | \n
**Benchmark Data**

<table>
<thead>
<tr>
<th>ussteel</th>
<th>time</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>running time</td>
<td>0000005506 00000034287</td>
<td>37.687</td>
</tr>
</tbody>
</table>

More detail?

### Suite Name/Chart

**General_ussteelJvm_32**

<table>
<thead>
<tr>
<th>Time</th>
<th>Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>100,000</td>
<td>nwb-cben-wv21</td>
</tr>
<tr>
<td>75,000</td>
<td>nwb-cben-wv22</td>
</tr>
</tbody>
</table>
| 50,000 | \n| 25,000 | \n| 0 | \n
**Benchmark Data**

<table>
<thead>
<tr>
<th>ussteel</th>
<th>time</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>running time</td>
<td>0000100369 0000046313</td>
<td>53.857</td>
</tr>
</tbody>
</table>

More detail?
Data
File Handling

- Good to go in .NET and JVM
- Be mindful of performance for IO intensive batch operations
- Optimizations in this area planned delivered in VC 2.2
procedure division.

exec sql
    connect to Northwind
end-exec

if sqlcode not = 0
    display "Error during CONNECT : " SQLCODE
    display mfsqlmessagetext
goback
end-if
Database access under Managed Code

- Oracle does not support managed code using Pro*COBOL
- Use OpenESQL for JDBC and ADO.NET
- Use Pro*COBOL directives `MODE=ANSI` and `FIPS` to determine non-standard ANSI syntax
- Some Oracle extensions supported by OpenESQL other extensions will need to be reworked
- No PL/SQL support
Java App Servers

• To do anything “Thin” will involve an app server
• All major Java App Servers supported in Visual COBOL 2.2
  – Tomcat
  – JBOSS
  – Webshere
  – WebLogic
• Web Services and JSPs
• Java coding required

• See community.microfocus.com for walkthrus
COBOL JVM Roadmap

From desktop to J2EE EJB deployment...

- **R3/4**
  - JVM COBOL Preview
- **JVM COBOL Launch**
- **2.0**
  - JDBC support
  - Pure JVM
- **2.1**
  - Web Container support under Tomcat
  - Web services, JSP and Servlet deployment
- **2.2**
  - Web container support
  - WebLogic
  - Websphere
  - JBOSS
- **2.3**
  - EJB container deployment

...re-use core COBOL applications directly within the Java Virtual Machine
Multi-process applications
App Servers

App Server Process

Thread

Data

REENTRANT(2) directive won’t help
Run-units as containers

- Run units associate COBOL programs, data and working-storage with a given user
## Modernizing your architecture

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Presentation</strong></td>
<td>• COBOL, Java or C#, VB…</td>
</tr>
<tr>
<td><strong>Business Access</strong></td>
<td>• OO COBOL Wrapper</td>
</tr>
<tr>
<td></td>
<td>• Run unit</td>
</tr>
<tr>
<td><strong>Business logic</strong></td>
<td>• Existing Procedural COBOL</td>
</tr>
<tr>
<td><strong>Data Access</strong></td>
<td>• Isolated data IO modules</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>• RDBMS</td>
</tr>
<tr>
<td></td>
<td>• Data files</td>
</tr>
</tbody>
</table>
Other considerations

- Dialog System
- 3rd Party support
- Win32APIs
- CBL_ routines
- Pointers
- Native code
Case studies

- Midland National Life
- Nationwide
- Zucchetti
Re-hosting to the Azure Cloud
Summary

• Managed code is a fantastic way to modernize your application
  – Get more done using Java and .NET frameworks
  – Create new user interfaces and services based deployments

• Be prepared
  – New skills – OO, Java, .NET, frameworks
  – Check for unsupported options such as APIs and database pre-compilers

• Speak to Micro Focus before you start or at least, let us know once you’ve deployed!
Questions