The tool developers Tool-Stack

Mike Fechner, Consultingwerk Ltd.
mike.fechner@consultingwerk.de
Consultingwerk Ltd.

- Independent IT consulting organization
- Focusing on **OpenEdge** and related technology
- Located in Cologne, Germany, subsidiary in UK
- Customers in Europe, North America, Australia and South Africa
- Vendor of tools and consulting services
- 27 years of Progress experience (V5 … OE11)
- Specialized in GUI for .NET, Angular, OO, Software Architecture, Application Integration

http://www.consultingwerk.de/
Introduction

- Developer of **SmartComponent Library** Framework for OpenEdge Entwickler
- Source code shipped to clients, 99% ABL code
- Used by over 20 customer
- Up to weekly releases (customer usually during development on maximum 3 month old release)
- Fully automated update of the framework DB at client
- Almost no regression bugs within last 8 years
Introduction

- Supported from OpenEdge 10.2B, 11.3.2 … 11.7
- Very high quality standards
- Framework code should act as reference code
- Framework provides a large number of APIs
  - Documentation of huge importance; developers won’t use API’s which are unknown
Agenda

- **Progress Developer Studio**
  - OEDT
  - Perforce SCM
  - ANT & PCT
  - Jenkins
  - ABLUnit and SmartUnit
  - Proparse
  - SonarSource and SonarLint
  - JIRA and Mylyn
Progress Developer Studio

- Eclipse-based, extensible with plug-ins
- Available since OpenEdge 10.1A (permanently improved – although some releases were step backwards)
- Available for Windows OS only
- Alternative to AppBuilder / OpenEdge Studio
- Separate license; included in OEDK) parallel installation with OE Studio is possible
- AppBuilder, ADM1/2 and Dynamics integrated
- OpenEdge 11.7 is a very good release!
  - Pretty recent Eclipse release
PDSOE at Consultingwerk

- Central integration hub for a number of tools
  - SCM
  - Issue-Tracking
  - Unit-Testing, Code-Quality
  - Automated Builds
  - …

- Productivity while development
  - OEDT, Intelligent Macros, Templates
  - Integration with SmartComponent Library
Demo

- Editor-Features
  - Navigation your source code
  - Active templates, Macros
  - Block selection
Agenda

- Progress Developer Studio
- OEDT
- Perforce SCM
- ANT & PCT
- Jenkins
- ABLUnit and SmartUnit
- Proparse
- SonarSource and SonarLint
- JIRA and Mylyn
OEDT

- Extension for Progress Developer Studio for OpenEdge (PDSOE is required)
- Commercial Eclipse plug-in developed by Progress AP H&H Berlin, Germany
- [https://www.hh-berlin.de/oedt/](https://www.hh-berlin.de/oedt/)
- Currently in public beta (expected FCS in July 2017)
OEDT Features

- Optimized non-blocking Build-process
  - Compilation based on dependencies (.i, .cls)
  - Compilation of previously failed source
  - Preprocess/Debug listing view
- Editor with *true* code-completion
  - very fast
  - Method and procedure arguments
  - Include file names
  - CamelCaseClassNames
  - Doc Comments
Demo

- OEDT web site
- Editor Features
- Doc comments
Agenda

- Progress Developer Studio
- OEDT
- Perforce SCM
- ANT & PCT
- Jenkins
- ABLUnit and SmartUnit
- Proparse
- SonarSource and SonarLint
- JIRA and Mylyn
Requirements for the SCM Tool

- Efficient management of source & related assets
- Central hub for *every* task related to the source (I get very upset when developers mail source)
- Many integrations (Build tools, task management, code review tools, IDE’s)
- Scriptable: Batch files and ANT
- Reliable and maintenance free (just like OE)
- Performant (just like OE)
- Allowing remote access to the source
- SCM means more than version control
Perforce SCM

- SCM based on a central VCS
- DVCS features (e.g. local developer branches)
- Perforce Server, Perforce Proxy, Perforce Client(s), Integrations
- Very powerful branching
  - between code-lines
  - Between individual files (Interfile-Branching)
Perforce SCM

- Powerful integration of code lines (Merge); honoring previous decisions made during merge
- Fully automatic merging
- Optional locking of sources, rarely used
  - Developers don’t work on same files that often in reality
  - Merge features of Perforce take fear away from parallel changes by different developers
DVCS Features (D as in Distributed)
Agenda

- Progress Developer Studio
- OEDT
- Perforce SCM
- ANT & PCT
  - Jenkins
  - ABLUnit and SmartUnit
  - Proparse
  - SonarSource and SonarLint
  - JIRA and Mylyn
ANT

- Apache Build Scripting Language
- XML based batch file, OS-independent
- ANT-File may contain multiple targets (sub routines)
- Sub routines may have dependencies to each other
- Macros
- Error-Handling & Conditional execution
- Properties/Variables/Parameters
ANT

- Originally a Java-Build System
- Compiles Java-Code, executes JUnit Tests, etc.
- Other built in features (among many others):
  - File manipulations, copy, delete, …
  - ZIP, UNZIP
  - SCM Interaction
- Extensible via plug-ins (offering further ANT Tasks)
### File Tasks

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attr</td>
<td>Changes the permissions and/or attributes of a file or all files inside the specified directories. Currently, it has effect only under Windows.</td>
</tr>
<tr>
<td>Checksum</td>
<td>Generates a checksum for a file or set of files. This task can also be used to perform checksum verifications.</td>
</tr>
<tr>
<td>Chgrp</td>
<td>Changes the group ownership of a file or all files inside the specified directories. Currently, it has effect only under Unix.</td>
</tr>
<tr>
<td>Chmod</td>
<td>Changes the permissions of a file or all files inside the specified directories. Currently, it has effect only under Unix. The permissions are also UNIX style, like the arguments for the <code>chmod</code> command.</td>
</tr>
<tr>
<td>Chown</td>
<td>Changes the owner of a file or all files inside the specified directories. Currently, it has effect only under Unix.</td>
</tr>
<tr>
<td>Concat</td>
<td>Concatenates multiple files into a single one or to Ant's logging system.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies a file or Fileset to a new file or directory.</td>
</tr>
<tr>
<td>Copydir</td>
<td>Deprecated. Use the Copy task instead.</td>
</tr>
<tr>
<td>Copyfile</td>
<td>Deprecated. Use the Copy task instead.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes either a single file, all files and sub-directories in a specified directory, or a set of files specified by one or more FileSets.</td>
</tr>
<tr>
<td>Deltree</td>
<td>Deprecated. Use the Delete task instead.</td>
</tr>
<tr>
<td>Filter</td>
<td>Sets a token filter for this project, or reads multiple token filters from a specified file and sets these as filters. Token filters are used by all tasks that perform file-copying operations.</td>
</tr>
<tr>
<td>FixCRLF</td>
<td>Modifies a file to add or remove tabs, carriage returns, linefeeds, and EOF characters.</td>
</tr>
<tr>
<td>Get</td>
<td>Gets a file from a URL.</td>
</tr>
<tr>
<td>Mkdir</td>
<td>Creates a directory. Non-existent parent directories are created, when necessary.</td>
</tr>
<tr>
<td>Move</td>
<td>Moves a file to a new file or directory, or a set(s) of file(s) to a new directory.</td>
</tr>
<tr>
<td>Patch</td>
<td>Applies a &quot;diff&quot; file to originals.</td>
</tr>
<tr>
<td>Rename</td>
<td>Deprecated. Use the Move task instead.</td>
</tr>
<tr>
<td>RenameExtensions</td>
<td>Deprecated. Use the Move task with a glob mapper instead.</td>
</tr>
<tr>
<td>Replace</td>
<td>Replace is a directory-based task for replacing the occurrence of a given string with another string in selected file.</td>
</tr>
<tr>
<td>ReplaceRegExp</td>
<td>Directory-based task for replacing the occurrence of a given regular expression with a substitution pattern in a file or set of files.</td>
</tr>
<tr>
<td>SetPermissions</td>
<td>Changes the permissions of a collection of resources.</td>
</tr>
<tr>
<td>Sync</td>
<td>Synchronize two directory trees.</td>
</tr>
<tr>
<td>Tempfile</td>
<td>Generates a name for a new temporary file and sets the specified property to that name.</td>
</tr>
<tr>
<td>Touch</td>
<td>Changes the modification time of a file and possibly creates it at the same time.</td>
</tr>
</tbody>
</table>

### Java2 Extensions Tasks

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jarch-available</td>
<td>Check whether an extension is present in a FileSet or an ExtensionSet. If the extension is present, the specified property is set.</td>
</tr>
</tbody>
</table>
Demo

- Simple ANT script
## Command-line Options Summary

```
ant [options] [target [target2 [target3] ...]]
```

**Options:**

- `-help, -h` print this message and exit
- `-projecthelp, -p` print project help information and exit
- `-version` print the version information and exit
- `-diagnostics` print information that might be helpful to diagnose or report problems and exit
- `-quiet, -q` be extra quiet
- `-silent, -S` print nothing but task outputs and build failures
- `-verbose, -v` be extra verbose
- `-debug, -d` print debugging information
- `-emacs, -e` produce logging information without adornments
- `-lib <path>` specifies a path to search for jars and classes
- `-logfile <file>` use given file for log
- `-l <file>` `..`
- `-logger <classname>` the class which is to perform logging
- `-listener <classname>` add an instance of class as a project listener
- `-noinput` do not allow interactive input

- `-buildfile <file>` use given buildfile
- `-file <file>` `..`
- `-f <file>` `..`
- `-D<property>=<value>` use value for given property
- `-keep-going, -k` execute all targets that do not depend on failed target(s)
- `-propertyfile <name>` load all properties from file with `-D` properties taking precedence

- `-inputhandler <class>` the class which will handle input requests
- `-find <file>` (s)earch for buildfile towards the root of
- `-s <file>` the filesystem and use it
- `-nice number` A niceness value for the main thread:
  1 (lowest) to 10 (highest); 5 is the default
- `-nouserlib` Run ant without using the jar files from `{$user.home}/.ant/lib`
- `-noclasspath` Run ant without using CLASSPATH
- `-autoproxy` Java 1.5+ : use the OS proxies
- `-main <class>` override Ant's normal entry point
<!-- Create a delta.df file -->
<echo />
<echo message="Creating delta.df to prepare schema update of your SmartDB ..." />
<echo />

<PCTDumpIncremental destFile="temp/delta.df" dlcHome="${dlc}"
dbname="temp/ref-SmartDB" singleUser="true" />
<SourceDB name="${smartdb}" hostName="${smartdbhost}" dbPort="${smartdbport}" />
</PCTDumpIncremental>

<!-- Cleanup -->
delete file="incrdump.e" />

<!-- Verify if delta.df file was created and load Schema, if there are schema changes -->
<available file="temp/delta.df" property="delta.df.present"/>
<antcall target="RemoveDropTable" />
<antcall target="LoadSchema" />

<target name="LoadSchema" if="${delta.df.present}">

<!-- Prompt user before continue -->
<echo />
<echo message="We are now going to load the new schema into the SmartDB. " />
<echo message="Please back it up first!" />
<echo message="The pending schema changes can be reviewed in temp/delta.df." />

<input
  message="Continue (y/n)?"
  validargs="y,n"
  addproperty="do.continue"/>

<condition property="do.abort">
  <equals arg1="n" arg2="${do.continue}"/>
</condition>

<fail if="do.abort">Upgrade aborted by user.</fail>
Filtering of drop table

- SmartDB may be integrated into application database – so during schema update, we need to skip DROP TABLE, DROP SEQUENCE, ...

```xml
<target name="RemoveDropTable" if="${delta.df.present}">

    <copy file="temp/delta.df" tofile="temp/delta2.df">
        <filterchain>
            <linecontainsregexp negate="true" caseSensitive="false">
                <regexp pattern="drop \(table|sequence\) *" />
            </linecontainsregexp>
        </filterchain>
    </copy>

    <copy file="temp/delta2.df" tofile="temp/delta.df"/>
    <delete file="temp/delta2.df"/>

</target>
```
PCT

- https://github.com/Riverside-Software/pct
- ANT tasks for OpenEdge
- Progress Compiler Tools
- OpenSource
- „Support“ via Github Issue-Tracking

Tasks
- PCT
- Dlchome
- PCTRun
- PCTCompile
- PCTWSComp
- PCTCreateBase
- Sports2000
- PCTDumpSchema
- PCTDumpSequences
- PCTLoadSchema
- PCTDumpIncremental
- PCTBinaryDump
- PCTBinaryLoad
- PCTDumpData
- PCTLoadData
- PCTSchemaDoc
- PCTLibrary
- PCTProxygen
- PCTXCode
- ProgressVersion
- PCTVersion
- ClassDocumentation
- HtmlDocumentation
- XmlDocumentation
- OEUnit
- ABLUnit
- RestGen
<target name="SmartComponentLibrary" description="" depends="create-icfdb, create-SmartDB, create-SmartComponentLibrary" />

<PCTCompile destDir="." graphicalMode="true" dlcHome="$\{progress.DLC\}"
    md5="$true" minSize="false" cpinternal="iso8859-1" cpstream="iso8859-1"
    compileUnderscore="true" inputchars="16000" assemblies="Assemblies"
    iniFile="ini/progress.ini" token="4000" progPerc="10">
    <fileset dir="." casesensitive="false">
        <patternset refid="smartcomponents.sources.consultingwerk"/>
    </fileset>
    <fileset dir="." casesensitive="false">
        <patternset refid="smartcomponents.sources.openedge"/>
    </fileset>
    <fileset dir="." casesensitive="false">
        <patternset refid="smartcomponents.sources.allprocedural"/>
    </fileset>
    <propath refid="smartcomponents.propath.demo"/>
    <DBConnectionSet refid="smartcomponents.db.demo"/>
</PCTCompile>

<deleteEmptyFolders dir="."/>
<!-- Targets for creating databases -->
<target name="create-emptydb" description="Builds an empty DB">
  <PCTCreateBase dbName="emptydb" destDir="..\DB"
    blockSize="8" dlcHome="${progress.DLC}" />
</target>

<target name="create-icfdb" description="Builds an empty ICFDB">
  <PCTCreateBase dbName="icfdb" destDir="..\DB"
    schemaFile="..\DB\build\icfdb\icfdb.df" structFile="..\DB\build\icfdb\icfdb.st"
    blockSize="8" dlcHome="${progress.DLC}" />
</target>

<target name="create-SmartDB" description="Builds an empty SmartDB">
  <PCTCreateBase dbName="SmartDB" destDir="..\DB"
    schemaFile="..\DB\build\SmartDB\SmartDB.df" structFile="..\DB\build\SmartDB\SmartDB.st"
    blockSize="8" dlcHome="${progress.DLC}" />
</target>

<target name="create-Sports2000" description="Builds an empty Sports2000 DB">
  <PCTCreateBase dbName="sports2000" destDir="..\DB"
    schemaFile="..\DB\build\Sports2000\Sports2000.df" structFile="..\DB\build\Sports2000\Sports2000.st"
    blockSize="8" dlcHome="${progress.DLC}" />
</target>
PCT

- Sample SmartComponent Library Database Upgrade Utility
ANT in PDSOE

- ANT editor with outline view and syntax validation
- Execution and configuration of ANT jobs from within the IDE
- ANT Builder
HTML class documentation

- **PCTCompile** with Debug-Listing Output
- **PCT ClassDocumentation** task extracts information about classes and methods into XML
- **HtmlDocumentation** Task generates documentation website
- **Jenkins Post-Build activity** performs SFTP upload to hosting webserver

- Alternative to **ABLDoc** included in OpenEdge
Agenda

- Progress Developer Studio
- OEDT
- Perforce SCM
- ANT & PCT
- **Jenkins**
  - ABLUnit and SmartUnit
  - Proparse
  - SonarSource and SonarLint
  - JIRA and Mylyn
Jenkins CI Server

- Continuous Integration – permanent merging of various changes
- Forked from Hudson CI
- Standard tool for centralized execution of build and test jobs
- Controlled environment for the execution of (Build or Test) „Jobs“
- Visualization of success or failure of jobs, visualization of Unit Test results
- Emails on fail or other events
<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>Name</th>
<th>Letzter Erfolg</th>
<th>Letzter Fehler</th>
<th>Letzte Dauer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>SmartComponent Library 117-64</td>
<td>23 Stunden - #252</td>
<td>5 Tage 5 Stunden - #279</td>
<td>10 Minuten</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>SmartComponent Library 117-64 - Class Documentation Test</td>
<td>1 Monat 10 Tage - #1</td>
<td>Unbekannt</td>
<td>38 Minuten</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>SmartComponent Library 117-64 - Checkin Classdoc</td>
<td>1 Tag 14 Stunden - #24</td>
<td>10 Tage - #19</td>
<td>33 Minuten</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>SmartComponent Library 117-64 - Class Documentation</td>
<td>5 Stunden 33 Minuten - #49</td>
<td>5 Tage 6 Stunden - #42</td>
<td>56 Minuten</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Unit-Test SCL 117-64 - colon U Test</td>
<td>17 Stunden - #186</td>
<td>Unbekannt</td>
<td>7 Minuten 25 Sekunden</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Unit-Test SCL 117-64 - Localizable Test</td>
<td>16 Stunden - #186</td>
<td>Unbekannt</td>
<td>2 Minuten 32 Sekunden</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Unit-Test SCL 117-64 - ParameterComment Validation</td>
<td>17 Stunden - #186</td>
<td>4 Tage 23 Stunden - #172</td>
<td>10 Minuten</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Unit-Test SCL 117-64 - Common</td>
<td>8 Stunden 48 Minuten - #300</td>
<td>4 Tage 21 Stunden - #274</td>
<td>1 Stunde 39 Minuten</td>
</tr>
</tbody>
</table>
Job execution triggered by

- Manually via web site
- Manually via command line (Jenkins CLI)
- (Manually) via web service
- Timer/schedule based
- After (successful) execution of other jobs
- After every SCM-Commit, Pull Request, Shelve, … depending on the SCM Tool
Build-Auslöser

- Builds von außerhalb starten (z.B. skriptgesteuert)
- Build when another project is promoted
- Builds zeitgesteuert starten
  - Zeitplan: 0 6 ***

- GitHub hook trigger for GITScm polling
- Perforce triggered build.
- Source Code Management System abfragen
  - Zeitplan: ****

- Meinen sie mit "***" wirklich "Jede Minute"? Vielleicht meinen Sie eher "H ****"
  - Letzter Lauf am Freitag, 5. Mai 2017 05:34 Uhr MESZ; Nachster Lauf am Freitag, 5. Mai 2017 05:34 Uhr MESZ.

- Ignore post-commit hooks

- Starte Build, nachdem andere Projekte gebaut wurden

Speichern  Apply
Master/Slave configuration

- Supports central management of jobs on multiple hosts
- Integration of different OS’s, like Linux on a VM
- Permanent verification that the backend runs on slave: Compile & Unit Tests
- Jenkins OpenEdge plug-in simplifies management of OE releases on slaves

https://github.com/riverside-software/openedge-jenkins-plugin
### Agent Linux (Linux VM auf Buildserver)

#### Labels
- linux

#### Projekte, die an Linux gebunden sind

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>Name</th>
<th>Letzter Erfolg</th>
<th>Letzter Fehlschlag</th>
<th>Letzte Dauer</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌞</td>
<td>🌞</td>
<td>11.6 » SmartComponent Library 116_64 - Linux</td>
<td>23 Stunden - #667</td>
<td>16 Tage - #625</td>
<td>6 Minuten 56 Sekunden</td>
</tr>
<tr>
<td>🌞</td>
<td>🌞</td>
<td>11.6 » Unit-Test SCL 116_64 - Common - Linux</td>
<td>9 Stunden 27 Minuten - #1217</td>
<td>16 Tage - #1103</td>
<td>11 Minuten</td>
</tr>
<tr>
<td>☁️</td>
<td>🌞</td>
<td>P4Review - Linux</td>
<td>5 Monate 8 Tage - #689</td>
<td>5 Monate 8 Tage - #932</td>
<td>2,2 Sekunden</td>
</tr>
<tr>
<td>🌞</td>
<td>🌞</td>
<td>SCL-Trunk » SmartComponent Library Trunk - Linux</td>
<td>23 Stunden - #291</td>
<td>3 Monate 2 Tage - #131</td>
<td>8 Minuten 59 Sekunden</td>
</tr>
<tr>
<td>🌞</td>
<td>🌞</td>
<td>SCL-Trunk » Unit-Test SCL Trunk - Common - Linux</td>
<td>9 Stunden 25 Minuten - #577</td>
<td>9 Tage 8 Stunden - #541</td>
<td>8 Minuten 22 Sekunden</td>
</tr>
</tbody>
</table>

**Symbol:**
- S: Small
- M: Medium
- L: Large

**Legende:**
- 🌞: Erfolgreich
- ☁️: Fehlschlag

Erstelldatum dieser Seite: 05.05.2017 05:39:03 MEZ

REST API: Jenkins ver. 2.40.2

Consultingwerk Toolstack
Performance considerations

- Server or Workstation
- Large number of CPU’s
- Lots of RAM
- SSD
- VM-Ware for alternative OS
  - Currently Linux
  - Maybe soon we’ll include different Window releases (more on next slides)
- The only physical server remaining at Consultingwerk (due to costs)
Agenda

- Progress Developer Studio
- OEDT
- Perforce SCM
- ANT & PCT
- Jenkins
- **ABLUnit and SmartUnit**
  - Proparse
  - SonarSource and SonarLint
  - JIRA and Mylyn
A very recent example …

- Windows 10 Creators Upgrade (April 2017) breaks INPUT THROUGH statements from Progress 8.3 - OpenEdge 11.7
- Used in a method to verify email addresses (MX record lookup), manual test not likely
- Jenkins Job alerted us around noon after the Windows update was applied to the build server
- Only two days later, discussions around the issue on Stackoverflow, Progress Communities and later in PANS – Unit Tests saved the day!
`//depot/SmartComponents\NET/117_64/Auto/Consultingwerk\UI\Network\NetworkHelper.cs#3

464 * param pcDomain Domain string to lookup as an MX
465 * return Returns TRUE if the MX lookup was successful
466 */
467 METHOD PROTECTED STATIC LOGICAL VerifyMXRecord (pcDomain AS CHARACTER);
468 DEFINE VARIABLE cOutput AS CHARACTER NO-UNDO FORMAT "x(70)";U.
469 DEFINE VARIABLE cFilename AS CHARACTER NO-UNDO.
470 DEFINE VARIABLE cErrorMessage AS CHARACTER NO-UNDO.
471 DEFINE VARIABLE iReturnedValue AS LOGICAL NO-UNDO.
472 DEFINE VARIABLE iValue AS INTEGER NO-UNDO.
473 DEFINE VARIABLE cValue AS CHARACTER NO-UNDO.
474
475 IF OPSYS BEGINS "WIN":"U THEN DO:
476 cFilename = SUBSTITUTE ("%1\nslookup.txt";U, SESSION\TEMP\DIRECTORY).
478 OUTPUT TO VALUE (cFilename).
479 PUT UNFORMATTED "set q=mx";U SKIP.
480 PUT UNFORMATTED pcDomain SKIP.
481 OUTPUT CLOSE.
482
483 INPUT THROUGH VALUE (SUBSTITUTE ("type ii | nslookup";U, QUOTER (cFilename)))
484 importLoop:
485 REPEAT ON ERROR UNDO, ITHROW:
486 IMPORT UNFORMATTED cOutput.
488 IF INDEX (cOutput, "****";U) > 0 THEN DO:
489 IF NUM-ENTRIES (cOutput, ";";U) >= 2 THEN DO:
490 cErrorMessage = TRIM (ENTRY (2, cOutput, ";";U)) + " (\1)";U.
491 LEAVE importLoop.
492 ELSE
cErrorMessage = "Unknown Error occurred for Domain: ";U.
493 END.
494 END.
495 END.
496
497 IF INDEX (cOutput, "****";U) > 0 THEN DO:
498 IF NUM-ENTRIES (cOutput, ";";U) >= 2 THEN DO:
499 cErrorMessage = TRIM (ENTRY (2, cOutput, ";";U)) + " (\1)";U.
500 LEAVE importLoop.
501 ELSE
cErrorMessage = "Unknown Error occurred for Domain: ";U.
502 END.
503 END.
504 IF INDEX (cOutput, "****";U) > 0 THEN DO:
505 IF NUM-ENTRIES (cOutput, ";";U) >= 2 THEN DO:
506 cErrorMessage = TRIM (ENTRY (2, cOutput, ";";U)) + " (\1)";U.
507 LEAVE importLoop.
508 END.
509 END.
510 END.
Build #182 (18.04.2017 13:11:22)

Build-Artefakte
profiler.zip 784,39 KB

Summary Of Changes - View Detail
by Mike Fechner@Consultingwerk
on 17.04.2017 21:44:05
SCL-1740: Implement Web Handlers that allow interacting with the session context

Gestartet durch vorgelagertes Projekt "SmartComponent Library 117-64", Build 229 ursprünglich ausgelöst durch:

- Build wurde durch eine SCM Änderung ausgelöst

Testergebnis (2 fehlgeschlagene Tests / +2)
Consultingwerk.UtilTest.NetworkHelper.EmailValidationTest.TestAddressNonValidMX
Consultingwerk.CoreTests.ValidateTests.ValidateTests.TestValidateEmailInvalidMX

Vor 15 Tage gestartet
Dauer: 48 Minuten auf Master
ABLUnit

- ABLUnit Unit Test Werkzeug available since OpenEdge 11.4
- Integrated into Progress Developer Studio
- Starting OpenEdge 11.7 Unit tests can be debugged in PDOSE
- Procedural und class based tests und testsuites
- @Test Annotation, @before, @after, ...
- @Test (expected=“ErrorClassName“)
- JUnit compatible output
ABLUnit Demo

- Demo ABL Unit in Progress Developer Studio
ABLUnit
Gilles QUERRET edited this page on 29 Jul 2016 · 5 revisions

Description
Run an ABLUnit tests sequence. For further information, refer to the progress documentation.

XML namespace

<ablunit />

Parameters

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>destDir</td>
<td>Directory where to put result file. Don't use destDir under Linux, as a bug prevents results.xml from being generated</td>
<td>Base directory</td>
</tr>
<tr>
<td>writeLog</td>
<td>Creates ablunit.log in temp directory in case of error</td>
<td>False</td>
</tr>
<tr>
<td>haltOnFailure</td>
<td>Stop the build process if a test fails (errors are considered failures)</td>
<td>False</td>
</tr>
</tbody>
</table>
destDir | Directory where to put result file. Don’t use destDir under Linux, as a bug prevents results.xml from being generated | Base directory
writeLog | Creates ablunit.log in temp directory in case of error | False
haltOnFailure | Stop the build process if a test fails (errors are considered failures as well) | False

† Only one of those attributes is mandatory † Mandatory attribute

ABLUnit inherits attributes from PCT and PCTRun.

Parameters as nested elements

Fileset

Support the standard Ant Fileset to get a set of test files.

Examples 1

```xml
<ABLUnit>
  <fileset dir="src" includes="**/*.p"/>
  <fileset dir="src2" includes="**/*.cls"/>
  <propath>
    <pathelement path="src"/>
    <pathelement path="src2"/>
    <!-- When running on OE up to 11.3, don’t forget to include ablunit.pl and OpenEdge.Core.
    <pathelement location="/path/to/ablunit.pl"/>
    <pathelement location="/path/to/OpenEdge.Core.pl"/>
    -->
  </propath>
</ABLUnit>
```
SmartUnit

- Unit-Test tool by Consultingwerk
- Only class based tests supported (tests can however execute procedures)
- Developed before other OpenEdge Unit-Test tools supported OO
- OpenEdge 10.2B – OpenEdge 11.7
- compatible output
- Alternative Test-Runner, e.g. based on scenario definitions or code quality checks like validation of comments in source
SmartUnit

- @Test Annotation für Klassen
- @Test Annotation für Methoden
- @before, @after, …
- @Test (expected="ErrorClassName")
- @Test (stop-after=5)
- Relies on PCT
- Tests compatible between ABLUnit and SmartUnit
- We are executing test within PDSOE using ABLUnit
Agenda

- Progress Developer Studio
- OEDT
- Perforce SCM
- ANT & PCT
- Jenkins
- ABLUnit und SmartUnit
- **Proparse**
  - SonarSource und SonarLint
  - JIRA und Mylyn
Proparse

- ABL syntax parser, abstract view on ABL source code, based on ANTLR
- Foundation for various migration routines
- Open source
  - github.com/oehide/proparse
  - github.com/consultingwerk/proparse
  - github.com/riverside-software/proparse
- Actively maintained in various forks
- Support for 11.7 ABL syntax
Proparse

- Details skipped here – shown in my next presentation here
Agenda

- Progress Developer Studio
- OEDT
- Perforce SCM
- ANT & PCT
- Jenkins
- ABLUnit and SmartUnit
- Proparse

- **SonarSource and SonarLint**
- JIRA and Mylyn
Code Quality measuring

Code Quality Measurement: WTFs/Minute

Good Code

Bad Code

http://commadot.com
SonarQube by SonarSource

- Frequently used Lint tool
- Support for various programming languages via plug-ins
  - Java, JavaScript, C#, HTML, XML, CSS, ...
- OpenEdge Plugin developed by Riverside Software (Gilles Querret)
  - engine open source
  - rules commercial
- Available since 2016, permanently new features added
SonarQube by SonarSource

- Locates problems or potential bugs
- Violation of coding-standards
- Code duplication
- Unit-Test coverage

- Web-Dashboard
- CLI Utility (HTML or XML Reports)
- Eclipse Integration
Consultingwerk Toolstack

Quality Gate: **Failed**
- Coverage on New Code: 26.4%
- Coverage should be at least 80.0%

**Bugs & Vulnerabilities**
- 45 Bugs (C)
- 0 Vulnerabilities (A)

**Code Smells**
- 56T Debt (A)
- 792 Code Smells

**Abdeckung**
- 8.7% Coverage

**Duplications**
- 2.6% Duplications
- 222 Duplicated Blocks

**Activity**
- 4. Juni 2017
- Project Analyzed

**Leak Period** since 41771
- 0 New Bugs (A)
- 0 New Vulnerabilities (A)
- 0 New Debt
- 0 New Code Smells

**Quality Profiles**
- SonarQube way
- SonarQube way (Standard)
- SonarQube way (OpenEdge)
- SonarQube way (OpenEdgeDB)
- SonarQube way (Web)
- SonarQube way (XML)
- SonarQube way

**Schlüssel**
- Consultingwerk:SCL

**Consultingwerk Toolstack**
<target name="sonarsource-standard">

   <p4jcounter name="change" property="p4.change" />
   <p4jcounter name="sonar_current_release" property="p4.sonar_current_release" />
   <!-- keep p4.change for release name -->
   <property name="sonar.projectVersion" value="${p4.sonar_current_release}" />

   <loadproperties srcFile="standard.properties"/>

   <echoproperties>
      <propertyset>
         <propertyref prefix="sonar."/>
      </propertyset>
   </echoproperties>

   <taskdef uri="antlib:org.sonar.ant" resource="org/sonar/ant/antlib.xml"/>

   <sonar:sonar/>

</target>
sonar.projectKey=Consultingwerk:SCL
sonar.projectName=SmartComponent Library
# Or use your Continuous Integration build number
sonar.projectVersion=1
sonar.projectDescription=...
# Comma-separated List of directories - Dump contains .df files, and src/openedge contains source code
sonar.sources=dump,src/openedge
# Comma-separated list of patterns to be excluded
sonar.exclusions=src/procedures/sample/excl/**
sonar.sourceEncoding=iso8859-1
# Directory where r-code can be found
sonar.oe.binaries=build
# Comma-separated list of PL or directories
sonar.oe.propath=src/openedge,src/openedge/src,src/openedge/Consultingwerk/Studio/ProductDumpHack/src,src/openedge/OERA
# DLC installation path (see next property)
sonar.oe.dlc=/Progress/OpenEdge117_64
# Append $DLC/gui, $DLC/tty and $DLC/src to the propath
sonar.oe.propath.dlc=true
# Comma-separated list of full DF files
sonar.oe.databases=dump/smartdb.df,dump/icfbd.df,dump/sports2000.df
# Semi colon separated list of entries, each entry is a comma-separated list of aliases (first entry is the db name)
sonar.oe.aliases=
# No Copy-Paste Detection on DF files
sonar.cpd.exclusions=dump/**
# Set to true to generate token listing in .tokens subdirectory
sonar.oe.cpd.debug=false
# Set to true to generate AST for each file in .proparse subdirectory
sonar.oe.proparse.debug=true

# extracted files from profiler.zip file from the unit tests jobs (artefact)
sonar.oe.coverage.profiler.dirs=profiler
SonarSource Demo

- Review the SonarSource web-console
SonarLint for Eclipse Demo

- Integration into Progress Developer Studio

```plaintext
/*
 Purpose:
 Notes:

@VisualDesigner.
METHOD PRIVATE VOID button1_Click( INPUT sender AS System.Object, INPUT e AS System.EventArgs )

    DEFINE VARIABLE cTest AS CHARACTER NO-UNDO INIT "abc" .

    cTest = "def" .

    REPEAT:

    END |

END METHOD.

METHOD PRIVATE VOID InitializeComponent() :
```
Development of custom rules

- Java Plugin in SonarQube, SDK
- Based on proparse or XREF, ...
- Rule requests can be issued at Github https://github.com/Riverside-Software/sonar-openedge/issues
- e.g. SUBSTITUTE ('&1 &2', lcLongchar1, cCharacter1)
- Community needs to provide knowledge of critical code!!!
Agenda

- Progress Developer Studio
- OEDT
- Perforce SCM
- ANT & PCT
- Jenkins
- ABLUnit and SmartUnit
- Proparse
- SonarSource and SonarLint

- JIRA and Mylyn
JIRA and Mylyn

- JIRA used as issue tracking
  - Bugs, new features, improvements, Tasks
  - Workflows
  - Agile planning
- JIRA integrated into PDSOE via Mylyn plug-in
- JIRA synchronises with Perforce (Changes linked to tickets)
- JIRA integrates with Jenkins (Build-Problems may create JIRA tickets)
Mylyn

- Eclipse Foundation project
- Eclipse plugin with highest number of downloads
- Integrates within Eclipse and with each other
  - Task Repositories (issue tracking systems, e.g. JIRA, Bugzilla)
  - SCM Tools (Perforce, Git, SVN)
  - Build Monitoring (Jenkins)
  - Time Tracking
- Installation in OpenEdge 11.7 easy!
Task Focussed UI

- When activating a task, Mylyn automatically maintains a task context by monitoring your interaction.
- The Task-Focused UI uses the task context to reduce information overload and to automate the management of editors, views, change sets and other UI elements.
- This increases productivity while working on the task and also makes it much easier to multitask because task contexts can be switched with a single click.
Task Focussed UI

- Filtering uninteresting elements from views (e.g. the Package Explorer) and decorating the most interesting elements
- Automatic expansion management in views
- Automatic code folding in editors
- Reordering of content assist proposals
- **Automatic management of open editors**
- **Automatic change set management**
- **Commit message auto-population**
Mylyn Demo

- Create JIRA Ticket
- Start Working in JIRA Ticket
- Activating and Deactivating Task Context
Mylyn Downloads

- http://www.eclipse.org/mylyn/
- May need to install archived version that works with your PDSOE version on 11.6
- OpenEdge 11.7 works with most recent version

- https://www.perforce.com/product/components/eclipse-plugin
Questions

http://www.consultingwerk.de/