

Is it Internationalized? A New Way to Keep Score

Contact sales@lingoport.com for a detailed discussion of your i18n needs

Try Globalyzer - register for a Globalyzer account at <http://globalyzer.com>

Thursday, June 16th, 2011

Today's Presenters



Adam Asnes
President & CEO
adam@lingport.com



Chris Raulf
Marketing Director
chris@lingoport.com



Olivier Libouban
Globalization Lead
olivier@lingoport.com



Mike McKenna
Zynga
Sr. Manager, International
Engineering



Leandro Reis
Adobe
Sr. Globalization Program
Manager

Poll #1 – Do you have Development Experience?

- Yes!
- Yes, but it's been a while
- No

Scorecard Initiative Background

- Requests from Lingoport customers
- Aha! Moment at Worldware presentation
 - Yahoo, Adobe, Autodesk

Lingoport

- Founded 2001
- Internationalization Services
 - Assessment
 - Project planning
 - I18n development
 - I18n testing
 - Localization integration
- Globalyzer
 - Internationalization software
 - Find and fix i18n issues in code

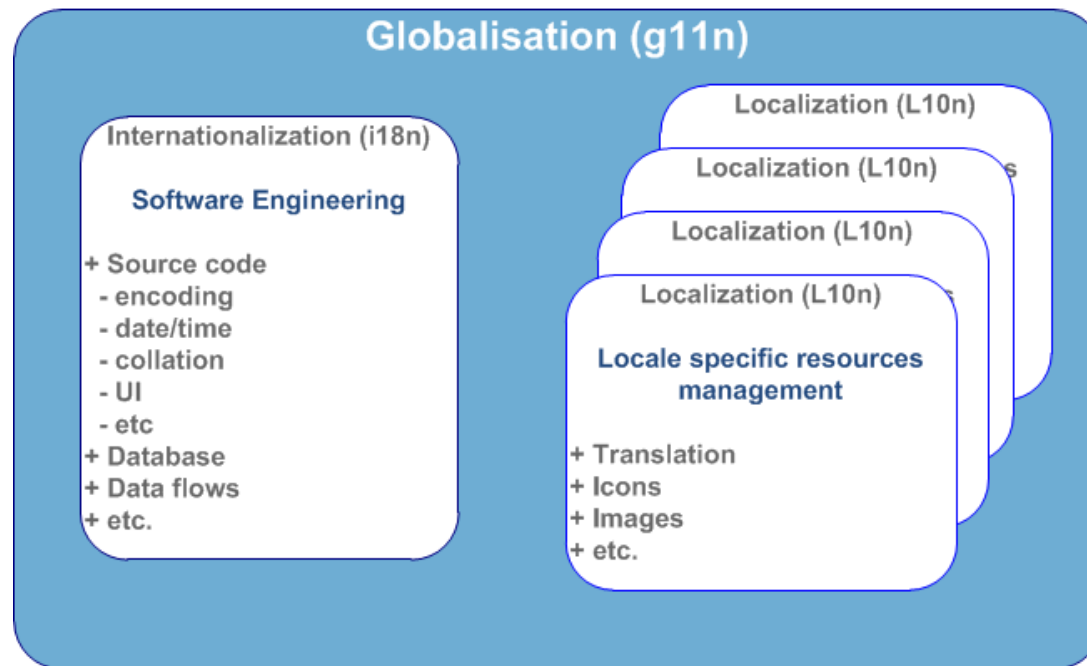
Terminology and basics

Introduction to business issues and Globalyzer

I18N ISSUES

Globalyzer deals with i18n

- Globalization (g11n) has two components :
 - Internationalization (i18n) : software engineering to enable localization
 - Localization (L10n): culture specific resources (translation, etc.)



Business Requirements

- Big revenue growth opportunity
- Little systematic tools and support for global-ready software development
- Development moves fast, teams can be diverse
- Lots to lose in time, revenue, costs and product acceptance when code isn't well internationalized

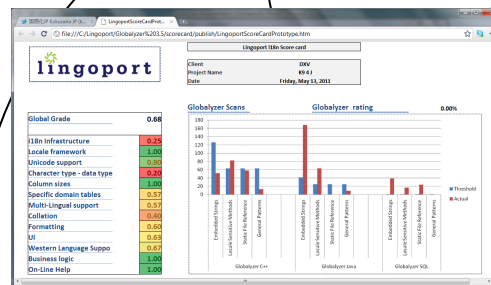
Organizational Perceptions of i18n

Developers:

- Straightforward, simple, handled
- Tier 3 bugs, at best
- Features come first
- Not enough time

L10n Managers:

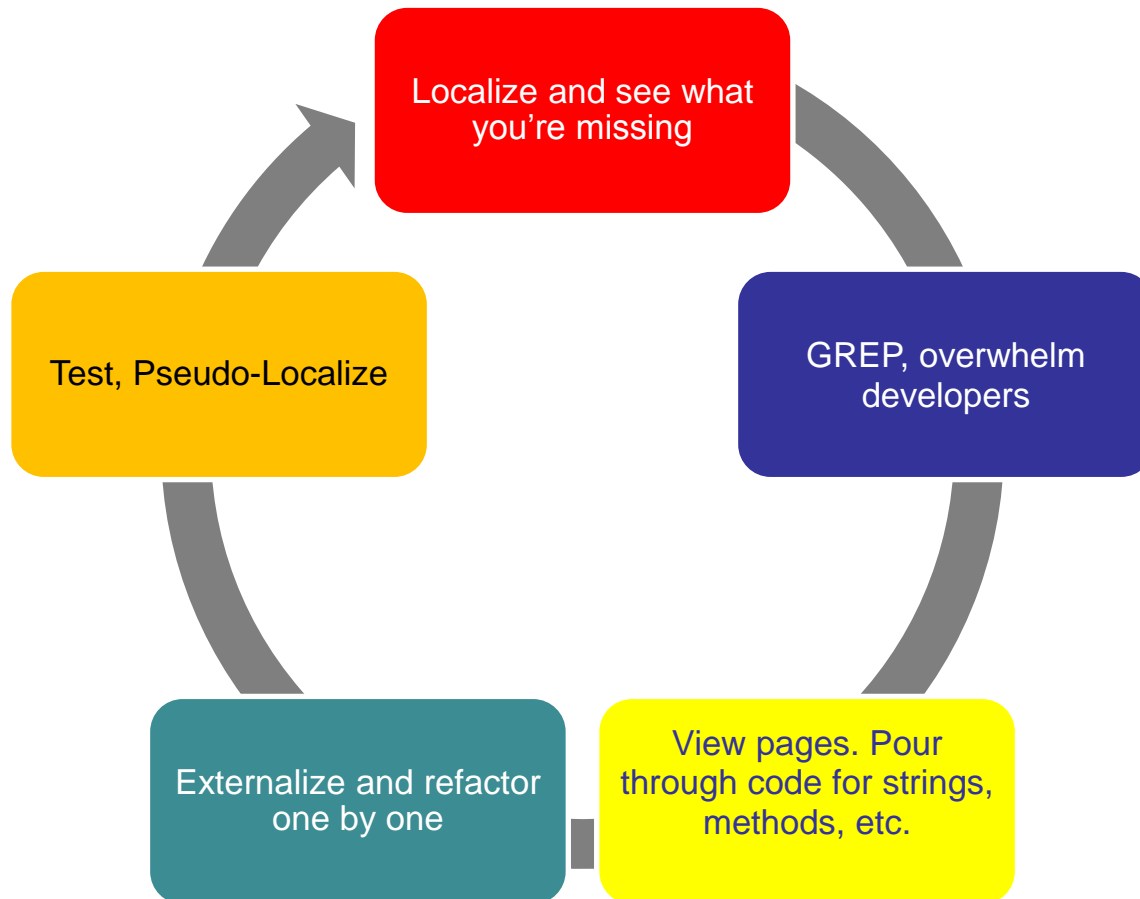
- Issues come up with every release
- Tier 1,2 &3 bugs
- Not enough support from dev.
- No way to verify until localization
- Lack deep knowledge of code



Business Managers:

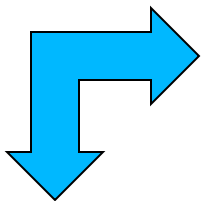
- Sales/biz expectations
- Time to market
- Over budget and late

Traditional Approach - repeat, and repeat, and repeat, and repeat

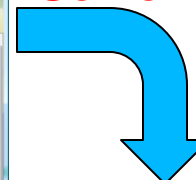


Globalyzer Server and Clients – Static Analysis

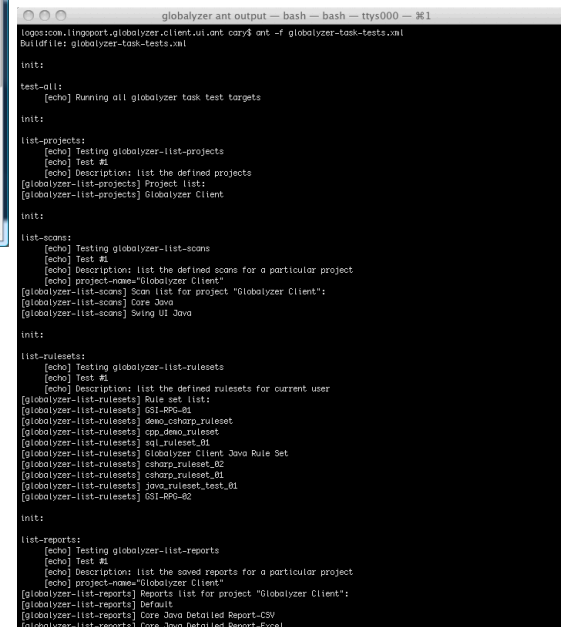
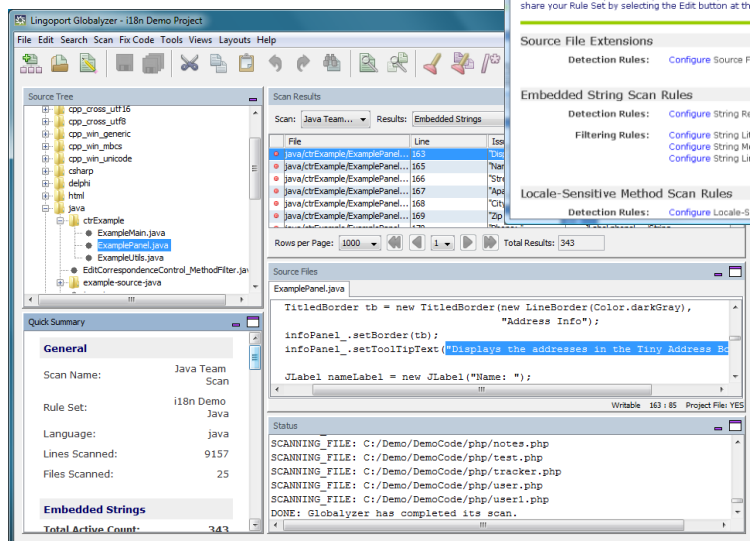
Client



Server

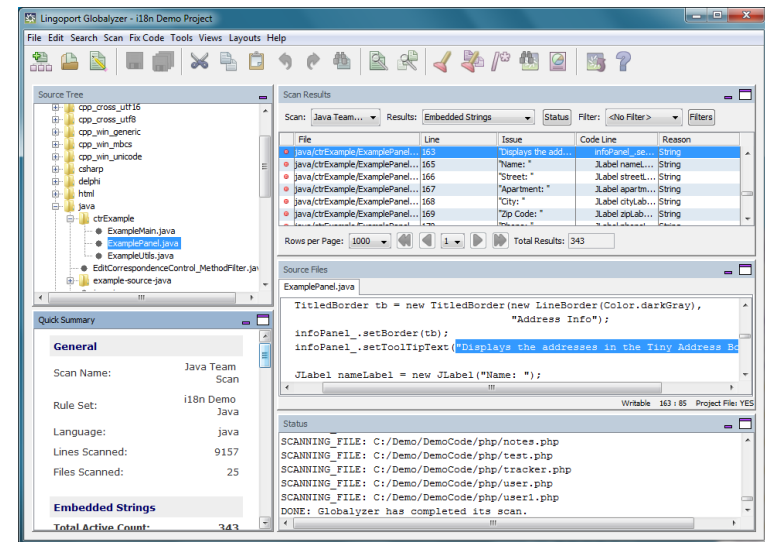


Command Line



Static Analysis for i18n

- Advantages
 - Analyze source, instead of test & repeat
 - Quantifies issues, location and fixes quickly
- Disadvantage
 - Takes time to configure
 - Too many “false” positives can discourage developers



Tuning and Filtering Process

Initial raw code report



Issue handling



Filtering

Tagging

Database status

Customized report



- XSL reporting

- Score card

Globalyzer: Usage model

There is value in knowing the global readiness of your code

- i18n assessment of a code base
 - Lots of code, need to find and fix complex i18n issues
 - Static analysis rather than limited iterative testing
 - Spread expertise across development teams
- Ongoing i18n detection and resolution
 - detects issues
 - proposes solutions and fixes
- i18n validation
 - to be integrated in a build or a QA process for instance
 - burns into the software i18n best practices
 - Integrates with Agile environments

Let's work with some code!

BRIEF GLOBALYZER DEMO

Poll #2 – i18n Process

- Do you have a development process for ensuring i18n requirements?
 - Yes
 - No

Continuous Integration

Monitoring i18n status for ongoing software development

GLOBALYZER SCORECARD

Continuous Integration & Analysis

- Use Globalyzer Command Line
- Track i18n issues over time
- Scorecard data
 - i18n requirements analysis
 - i18n hard data from Globalyzer

```

globalyzer ant output — bash — bash — ttys000 — #1
logos@com.lingoport:~/globalyzer_client/ui/ant$ ant -f globalyzer-task-tests.xml
Buildfile: globalyzer-task-tests.xml

init:

test-all:
[echo] Running all globalyzer task test targets

init:

list-projects:
[echo] Testing globalyzer-list-projects
[echo] Test #1
[echo] Description: list the defined projects
[globalyzer-list-projects] Project list:
[globalyzer-list-projects] Globalyzer Client

init:

list-scans:
[echo] Testing globalyzer-list-scans
[echo] Test #1
[echo] Description: list the defined scans for a particular project
[echo] project-name="Globalyzer Client"
[globalyzer-list-scans] Scan list for project "Globalyzer Client":
[globalyzer-list-scans] Core Java
[globalyzer-list-scans] Swing UI Java

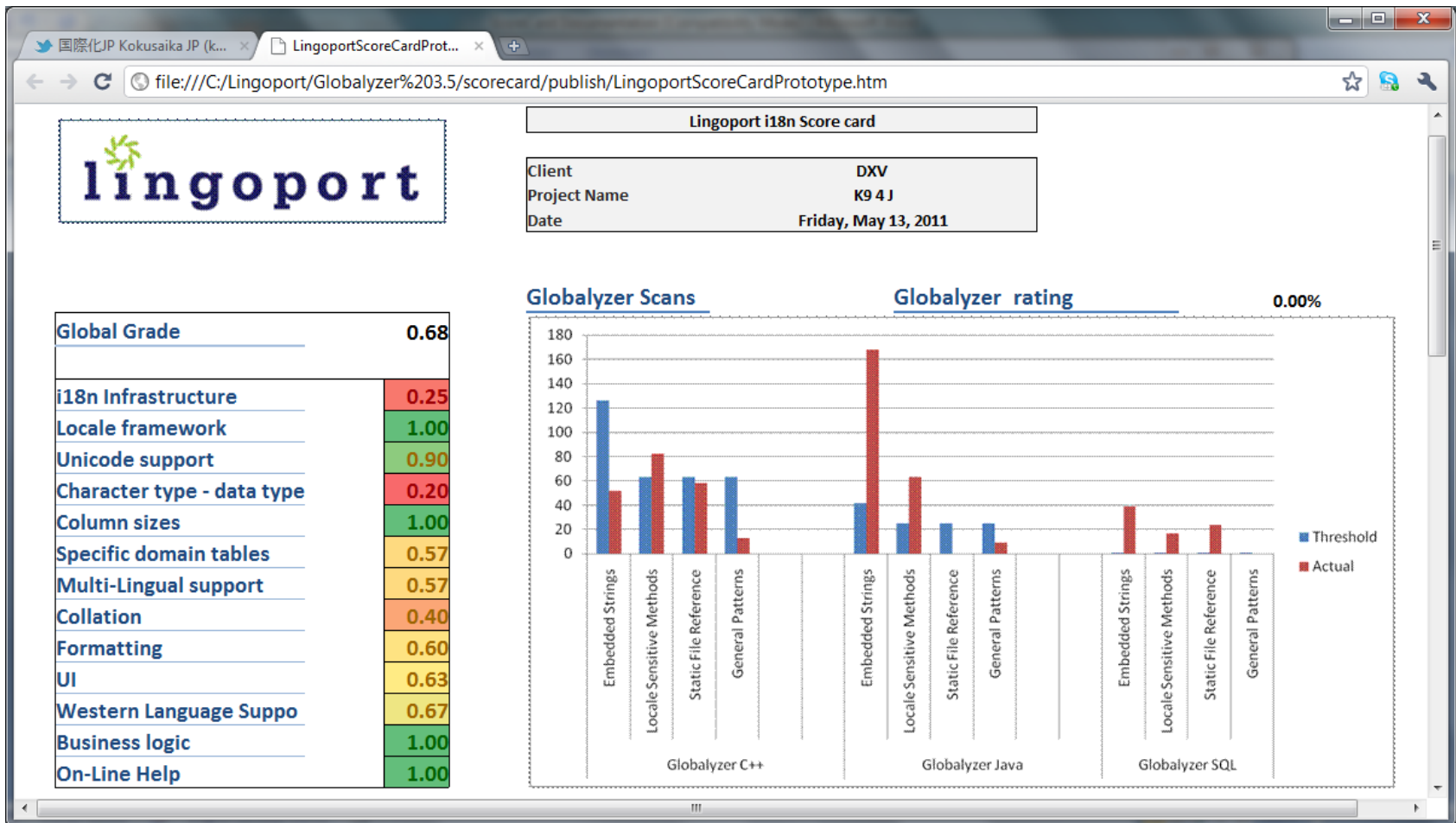
init:

list-rulesets:
[echo] Testing globalyzer-list-rulesets
[echo] Test #1
[echo] Description: list the defined rulesets for current user
[globalyzer-list-rulesets] Rule set list:
[globalyzer-list-rulesets] GSI-RPG-01
[globalyzer-list-rulesets] demo_csharp_ruleset
[globalyzer-list-rulesets] cpp_demo_ruleset
[globalyzer-list-rulesets] sql_ruleset_01
[globalyzer-list-rulesets] Globalyzer Client Java Rule Set
[globalyzer-list-rulesets] csharp_ruleset_02
[globalyzer-list-rulesets] csharp_ruleset_01
[globalyzer-list-rulesets] java_ruleset_test_01
[globalyzer-list-rulesets] GSI-RPG-02

init:

list-reports:
[echo] Testing globalyzer-list-reports
[echo] Test #1
[echo] Description: list the saved reports for a particular project
[echo] project-name="Globalyzer Client"
[globalyzer-list-reports] Reports list for project "Globalyzer Client":
[globalyzer-list-reports] Default
[globalyzer-list-reports] Core Java Detailed Report-CSV
[globalyzer-list-reports] Core Java Detailed Report-Excel
  
```

An HTML i18n score card



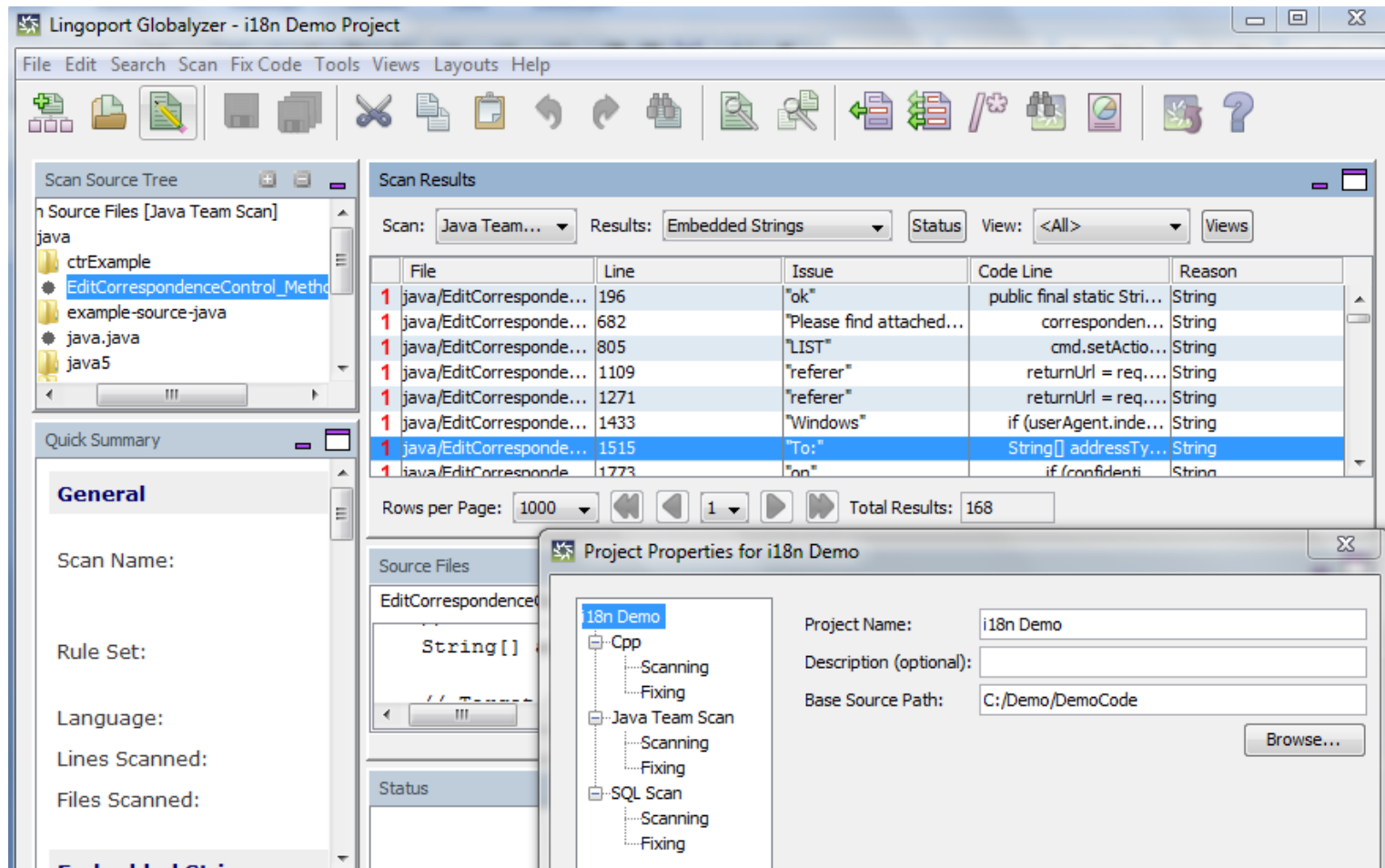
Overview

Lingoport Score Card is helps organizations put together reporting material for their internationalization projects.

It includes the following utilities:

- **Lingoport XL Template Score Card:** presents an overall view of the metrics, based on supporting tabs. Score cards for specific projects use this template as a guide.
 - Some data is filled in by an **i18n analyst**
 - Some data is provided by **Globalyzer**.
 - A **macro** refreshes the Globalyzer data and creates the **HTML score card output**
- **Template Executable utilities:** these utilities are configured to run
 - **Globalyzer scanning and reporting**, based on Rule Sets and Globalyzer projects
 - **Data handling** for input into Lingoport XL Score Card
 - **Running the XL macro** to refresh the data and produce the html result

The Globalyzer Project

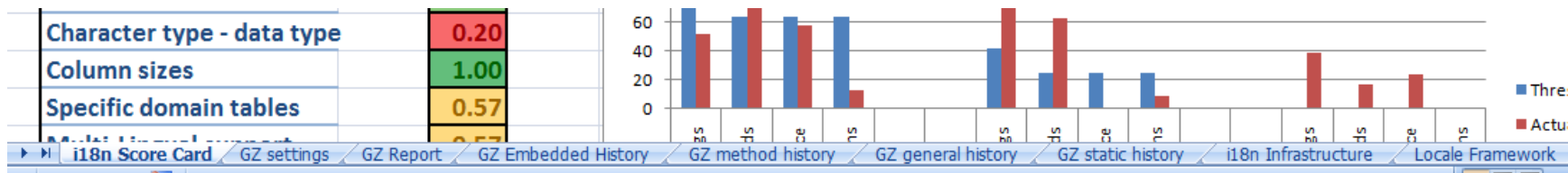


The screenshot displays the Lingoport Globalyzer interface for a project named "i18n Demo Project". The main window features a menu bar (File, Edit, Search, Scan, Fix Code, Tools, Views, Layouts, Help) and a toolbar with icons for file operations and scanning. On the left, a "Scan Source Tree" shows a directory structure including "java", "ctrExample", "EditCorrespondenceControl_Method", "example-source-java", "java.java", and "java5". Below this is a "Quick Summary" panel with fields for "General", "Scan Name:", "Rule Set:", "Language:", "Lines Scanned:", and "Files Scanned:". The central "Scan Results" panel shows a table of findings:

File	Line	Issue	Code Line	Reason
java/EditCorresponde...	196	"ok"	public final static Stri...	String
java/EditCorresponde...	682	"Please find attached...	corresponde...	String
java/EditCorresponde...	805	"LIST"	cmd.setActio...	String
java/EditCorresponde...	1109	"referer"	returnUrl = req....	String
java/EditCorresponde...	1271	"referer"	returnUrl = req....	String
java/EditCorresponde...	1433	"Windows"	if (userAgent.inde...	String
java/EditCorresponde...	1515	"To:"	String[] addressTy...	String
java/EditCorresponde...	1773	"on"	if (confidenti	String

Below the table, it indicates "Rows per Page: 1000" and "Total Results: 168". A "Project Properties for i18n Demo" dialog is open, showing a tree view of scan types (18n Demo, Cpp, Java Team Scan, SQL Scan) and fields for "Project Name: i18n Demo", "Description (optional):", and "Base Source Path: C:/Demo/DemoCode".

XL Score Card Template



The XL Template Score Card has many tabs:

- i18n analyst **requirements**
- Setting is the tab which gives **basic thresholds** for i18n issues
- Globalyzer Report, a tab linking to a Globalyzer **XL generated report**
- **XML generated reports** from Globalyzer
- **i18n Score Card** is the tab which aggregates and presents the results

I18n analyst tabs

lingoport		i18n score card details					
Criteria	Sub-criterion	Project setting	Project setting	Computed	To Be Filled	To Be Filled	Computed
		Required	Weight	Max Score	Implemented	Tested	Grade
			1	0.5			0.9
persistence		1	0.1	0.1	1	1	0.1
code		1	0	0.1	1	0	0.05
External components exchanges		1	0.1	0.1	1	1	0.1
3rd party tool data exchange		0	0.1	0	0	0	0
logging		0	0.1	0	0	0	0

- 1: The criteria.** To add a criterion, insert between the first and last criteria so the formula take it into account right away.
- 2:** If a criterion is not **required** for a specific project, set it to 0. If it is required, set it to 1, and the criterion becomes part of the grade.
- 3:** the **weight** must add up to 1 in the grey cell above. It helps check that the maximum grade will be 1 for the entire tab.
- 4:** the previous settings are done at the beginning of the project. As each criteria for the tab get **implemented and tested**, set the corresponding cell to 1, it will show as green. If 0, it is considered not implemented or tested and is colored red.
- 5:** The **score** for the criteria, based on the weight, if it is implemented and/or tested. The overall score for the tab show with a color, at the top of the grade column. Here it is **0.9**.

Relation with Score Card

Global Grade	0.73
i18n Infrastructure	1.00
Locale framework	1.00
Unicode support	0.90
Character type - data type	0.20
Column sizes	1.00
Specific domain tables	0.57

- The overall score for that tab then shows on the overall score card tab, “i18n score card”. The **‘Unicode support’** tab overall grade is shown, here is it **0.9**.
- Each tab follows the same pattern. You can add tab or ignore existing tabs, modify the criteria, their weight, based on your specific requirements.

Globalyzer XML reports

- **XML:** for the historical data, XML files are generated (note: those files are then handled to only take the last “n” results, for example the last 30 for a month worth of data), here:
 - Globalyzer_Embedded_History_tmp.xml
 - Globalyzer_Method_History_tmp.xml
 - Globalyzer_General_History_tmp.xml
 - Globalyzer_Static_History_tmp.xml

Example: Globalyzer Embedded History

- The XML data tabs leverage the XL ‘**external data**’ connection, under the Data ribbon, for instance the *GZ Embedded History* tabs connects to *Globalyzer_Embedded_History.xml* file.
- To get data which was newly generated, the **refresh** data must be executed. The “scorecardmacro” does just that.

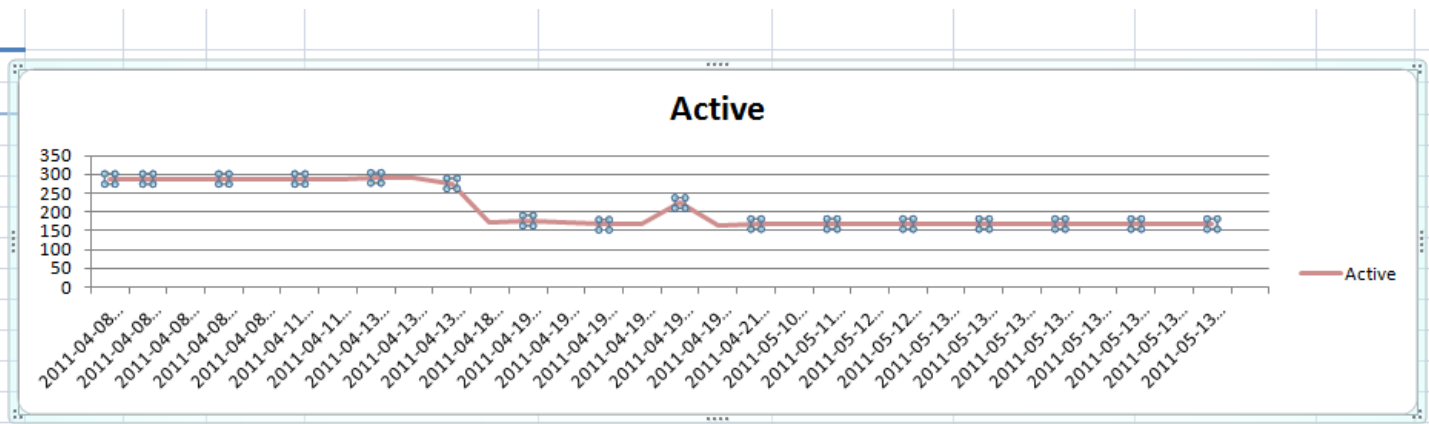
scanName	userName	result-type	history-id	history-date	active	resolved
Java Team Scan	olibouban@lingoport.com	Embedded Strings	297	2011-04-08 13:25 MDT	288	
Java Team Scan	olibouban@lingoport.com	Embedded Strings	301	2011-04-08 13:35 MDT	288	
Java Team Scan	olibouban@lingoport.com	Embedded Strings	305	2011-04-08 13:37 MDT	288	
Java Team Scan	olibouban@lingoport.com	Embedded Strings	309	2011-04-08 13:46 MDT	288	
Java Team Scan	olibouban@lingoport.com	Embedded Strings	313	2011-04-08 14:39 MDT	288	
Java Team Scan	olibouban@lingoport.com	Embedded Strings	317	2011-04-11 11:09 MDT	288	
Java Team Scan	olibouban@lingoport.com	Embedded Strings	321	2011-04-11 11:34 MDT	287	

Navigation: i18n Score Card / GZ settings / GZ Report / **GZ Embedded History**

Using XML data in the score card

Historical Data

Embedded Strings



The XML data is used as data sets for the graph in *the i18n score card* tab.

Creating specific historical graphs

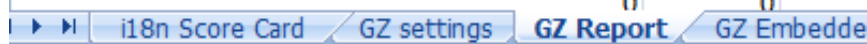
Using the Globalyzer reports (see <http://www.globalyzer.com/gzserver/help/commandline.htm>), one can use the same mechanism to:

- generate other XML files
- massage them for their own purpose
- connect to them from an Excel tab using **'external data'**
- create a graph based on that data

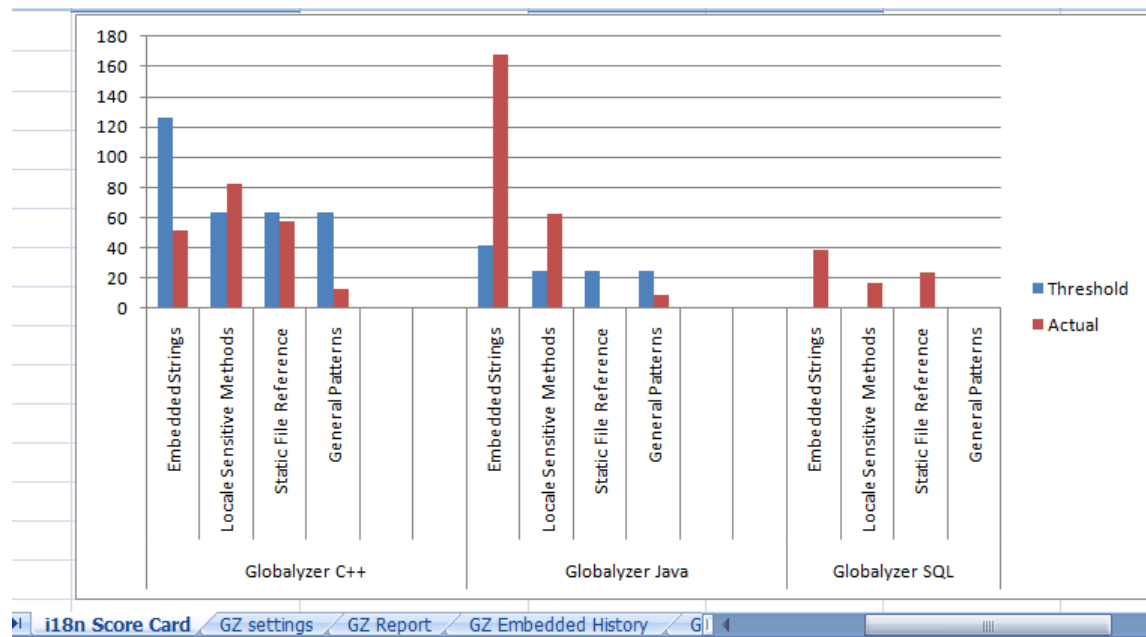
Globalyzer XLS report tab

“Globalyzer Report” tab is **linked** to a Globalyzer generated XLS file. It is refreshed when opening the template scorecard so there is no specific action in the “scorecardmacro” to update the data

<i>Globalyzer Aggregate Report</i>	0
<i>Project: i18n Demo</i>	0
<i>Date: May 13, 2011 4:49:16 PM</i>	0
	0
	0
Result Type	Count
Cpp - Lines Scanned	12622
Cpp - Files Scanned	15
Cpp - Embedded Strings	52
Cpp - Locale-Sensitive Methods	82
Cpp - General Patterns	58
Cpp - Static File References	13
	0
	0



Using the XLS report in the score card



In the score card template, that data is used to create a graph of the current scans as part of the 'i18n score card' tab

Creating specific reports

Using the Globalyzer reports (see <http://www.globalyzer.com/gzserver/help/commandline.htm>), one can use the same mechanism to

- generate other **XLS** files
- massage them for their own purpose
- **link** to them from an XL tab
- create graphs based on that data

Score card utilities

The utilities created for the templates are:

- A bat file to call Globalyzer scanning and reporting from the command line
- An Excel macro to get the latest scorecard, update the data from the reports and publish the HTML score card
- AVB Script to call that macro
- A Java utility to get the 'last n' historical scans
- A bat file to organize the above utilities

Getting Started with Globalyzer Scorecard

- Added in Globalyzer 3.6 – due in July
- Packaged with Globalyzer configuration services
- Discuss at sales@lingoport.com

Conclusion

- i18n is ongoing and supported with Globalyzer
- Scorecard answers: “is it internationalized?”
- Software engineers see and understand the implications of i18n
- QA integrates the reporting capability into the process
- Management and Localization see progress based on reporting
- Feedback, comments, questions ...

Questions & Answers

Adam Asnes

adam@lingoport.com

Olivier Libouban

olivier@lingoport.com

Resources

<http://www.lingoport.com>

Globalyzer

<http://www.globalyzer.com>

Blog

<http://i18nblog.com>

- Talk to us about your i18n needs
- Try Globalyzer at <http://globalyzer.com>