TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics
Since 1973

A Standard Process for Software Code Counting

June 2012

■ Los Angeles ■ Washington, D.C. ■ Boston ■ Chantilly ■ Huntsville ■ Dayton ■ Santa Barbara
■ Albuquerque ■ Colorado Springs ■ Ft. Meade ■ Ft. Monmouth ■ Goddard Space Flight Center ■ Ogden ■ Patuxent River ■ Silver Spring ■ Washington Navy Yard
■ Cleveland ■ Dahlgren ■ Denver ■ Johnson Space Center ■ Montgomery ■ New Orleans ■ Oklahoma City ■ Tampa ■ Tacoma ■ Vandenberg AFB ■ Warner Robins ALC

Outline

- **Background**
  - Software Size
  - SLOC Types
  - Issues
  - Unified Code Counter (UCC) and Differencing Capability
  - UCC Code Counting Examples
- **Graphical User Interface for UCC (GUCC)**
  - Why a GUCC?
  - GUCC
  - GUCC Screens
- **GUCC Demonstration**
- **Summary**

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics
Since 1973

Background

Software Size

- **Software size is a critical cost driver for software development cost estimates**
- **SLOC is…**
  - The most easily obtained size metric
  - The most mature metric
  - Understood by decision makers
  - An appropriate size metric when…
    - System is preceded by analogous programs
    - System requires high levels of complex processing

SLOC Types

Physical SLOC	Logical SLOC

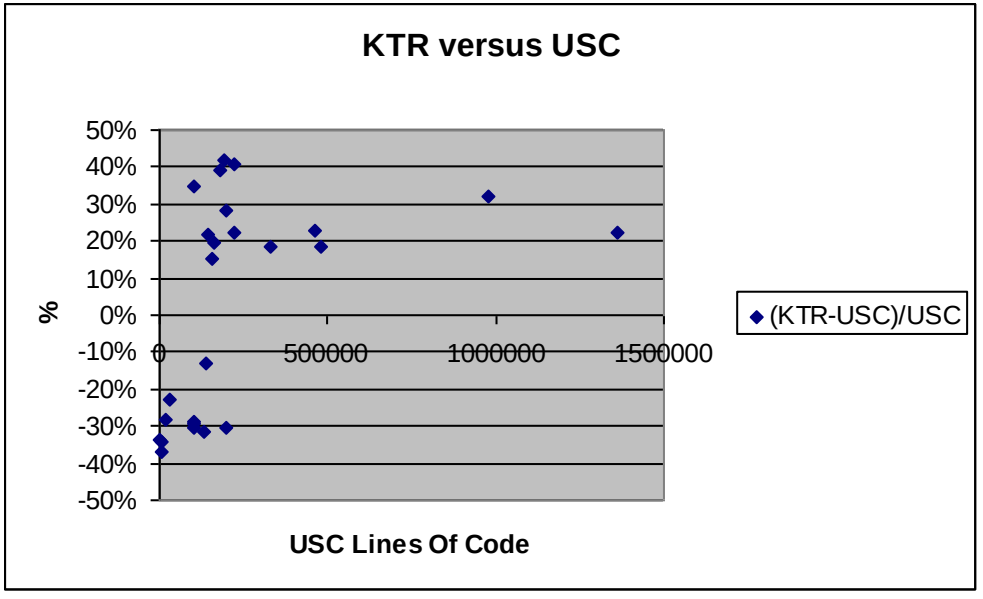
• Count of end-of-line markers
  ➢ Excluding comments and blanks
  ➢ Characterizes software in terms of physical size
• Programming language syntax independent
	• Count of single instructions
	  ➢ Excluding comments and blanks
	  ➢ Characterizes software in terms of instructions
	• Programming language syntax specific

- **Both logical and physical SLOC counts are useful for different parts of a life cycle cost estimate:**
  - Logical SLOC is more closely aligned with code functionality and development effort
  - Physical SLOC is useful for sizing maintenance effort

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics Since 1973

Issues

- **No one standard has been universally accepted for defining SLOC**
  - Every code counter claims to follow the "Software Size Measurement: A Framework For Counting Source Statements" by Robert E. Park
  - The problem is how each code counter interprets the definitions
  - The following chart (Cymerman 2006) shows +/-40% variance in C/C++ language counts vs. UCC (multiple contractors and programs) :



Issues

- **Numerous tools using multiple SLOC definitions are in use today (Jensen 2005):**
  - SEER-SEM
    - Control, mathematical, conditional, I/O, format JCL, and all other executable statements are included
    - Comments and Begin statements are excluded.
  - Price-S
    - Source lines to be developed and/or purchased.
    - Embedded declarations and data statements are included
    - Comments are excluded
  - COCOMO
    - All instructions requiring design, documentation, code and test created by programmers and processed into machine code
    - Generally all delivered logical, format and declaration instructions.
    - Comments and COTS are excluded.
    - Debug and test drivers (unless delivered) are also excluded
- **Code counting needs to be consistent to be used as an accurate basis for cost estimating**

The Unified Code Counter is a solution

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics Since 1973

Unified Code Counter (UCC)

- **Sponsored by USC**
- **Uses logical and physical SLOC definition from CMU/SEI-92-TR-020; Software Size Measurement: A Framework for Counting Source Statements**
- **Supports multiple software programming languages**
  - Tecolote developed several prototype counters for 'new' languages (e.g.: PERL, XML)
  - USC tested the prototype counters and incorporated them into releases of the UCC
  - USC willing to add counters as the need arises
- **Non-proprietary solution and FREE!**

http://sunset.usc.edu/research/CODECOUNT/

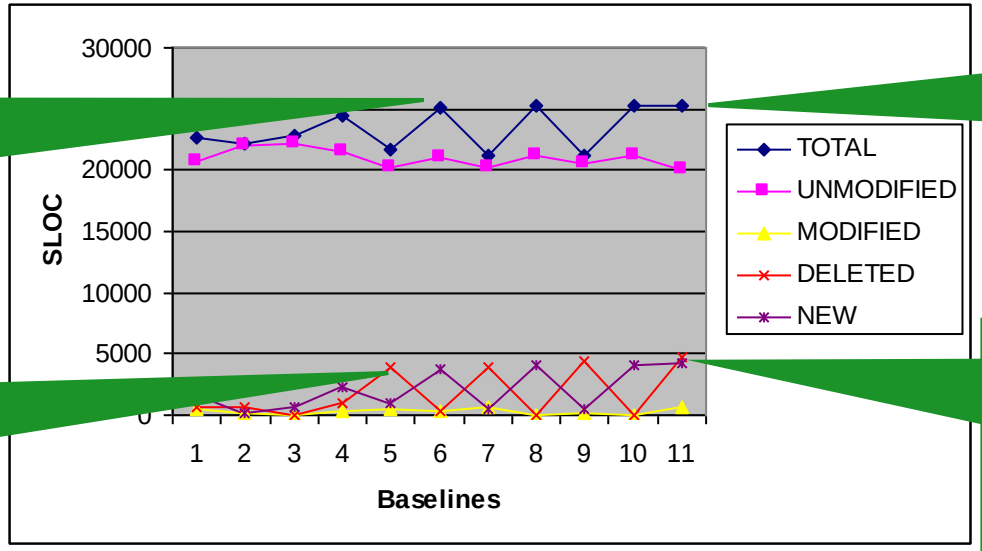
TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics Since 1973

UCC Differencing Capability

- **Prototype differencing capability implemented by Tecolote and incorporated into UCC by USC**
  - It compares source code from two versions of a software development
  - "Diff" tool example (Cymerman 2006) shows the usefulness to breakout the New and Deleted code from the total code counts:

Total counts show change, but without "Diff", you can't see why

Deleted code size is proportional to previous New code size



Code looks stable at this point, but in reality, it was dynamic!!

Notice that there is nearly 5,000 New SLOC and 5,000 Deleted SLOC Doesn't look STABLE

UCC Code Counting Examples

- Mid – Late 1990s: NRO CAIG identified need for consistent code counts
- 2002: Tecolote (Cymerman) identified USC Code Counter as potential solution
- 2003-2005: Tecolote (Cymerman and Legg) developed prototype counters and differencing capability; worked with USC in support of NRO CAIG
- 2008 – present: NRO and NGA form a joint working group using UCC (renamed in 2009)
- 2011: GUCC developed, tested, and briefed to Intelligence Community
- 2012: GUCC in use at NAVAIR; requested by Pt. Magu customer

Presented at the 2012 SCEA/ISPA Joint Annual Conference and Training Workshop - www.iceaaonline.com

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics
Since 1973

Graphical User Interface for the UCC (GUCC)

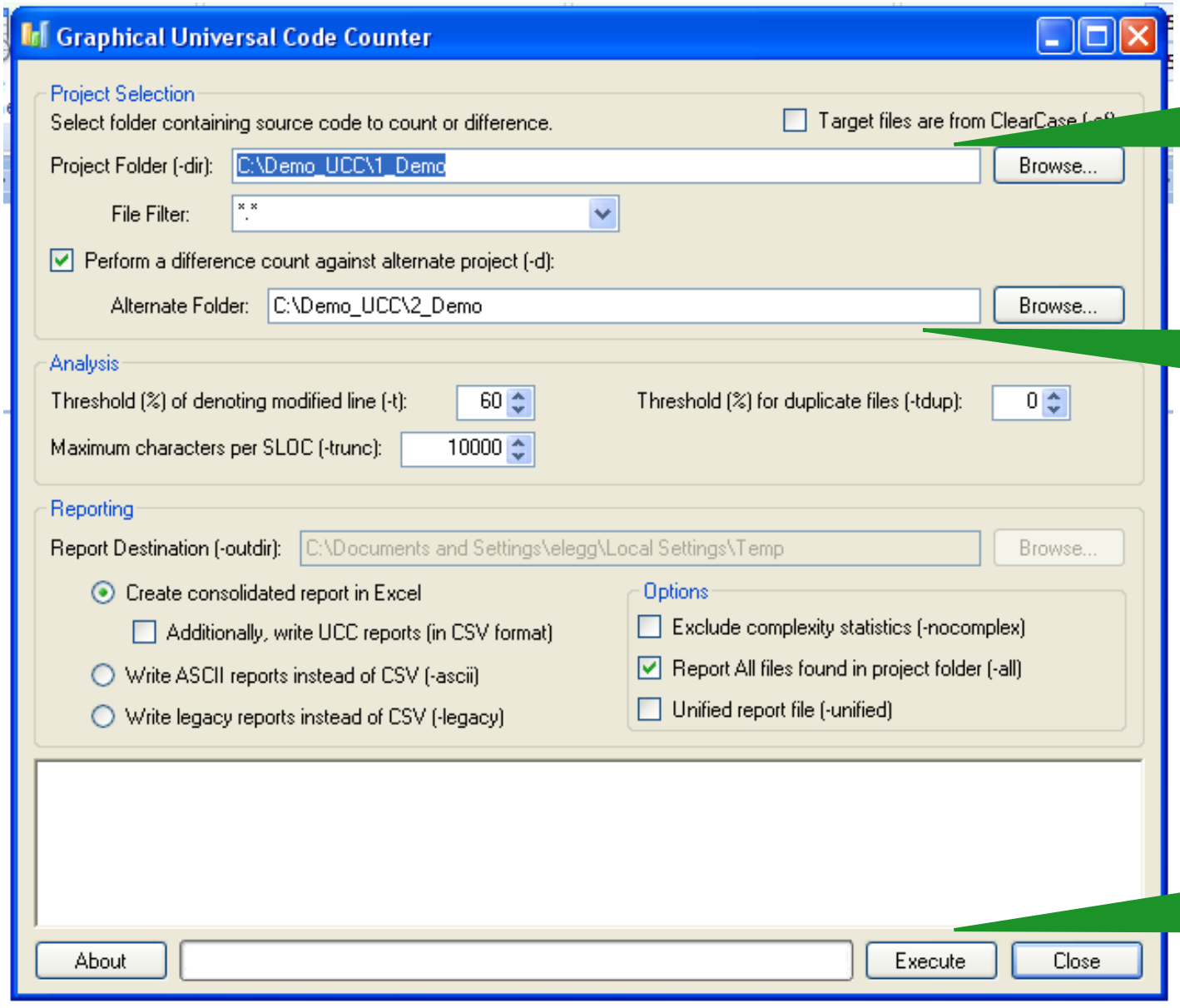
Why a GUCC?

- **The UCC provides credible and consistent counts across programs but can be labor intensive to implement**
  - Requires UCC knowledge of command line options
  - Requires MS DOS or Unix knowledge
  - More suited for developers and not analysts
- **The Graphical User Interface for the UCC (GUCC) was developed by Tecolote on internal research and development funding**
- **It operates in a Windows OS environment and reports the results of the UCC in a single consolidated MS Excel output**

- **User friendly GUCC allows wider use with little to no training**
  - One GUI window
  - Browse mechanism
  - Click options
  - Execute button
- **GUCC shows**
  - Totals on one tab of excel output
  - Files not counted
  - All output from the UCC

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics Since 1973

GUCC Input Screen



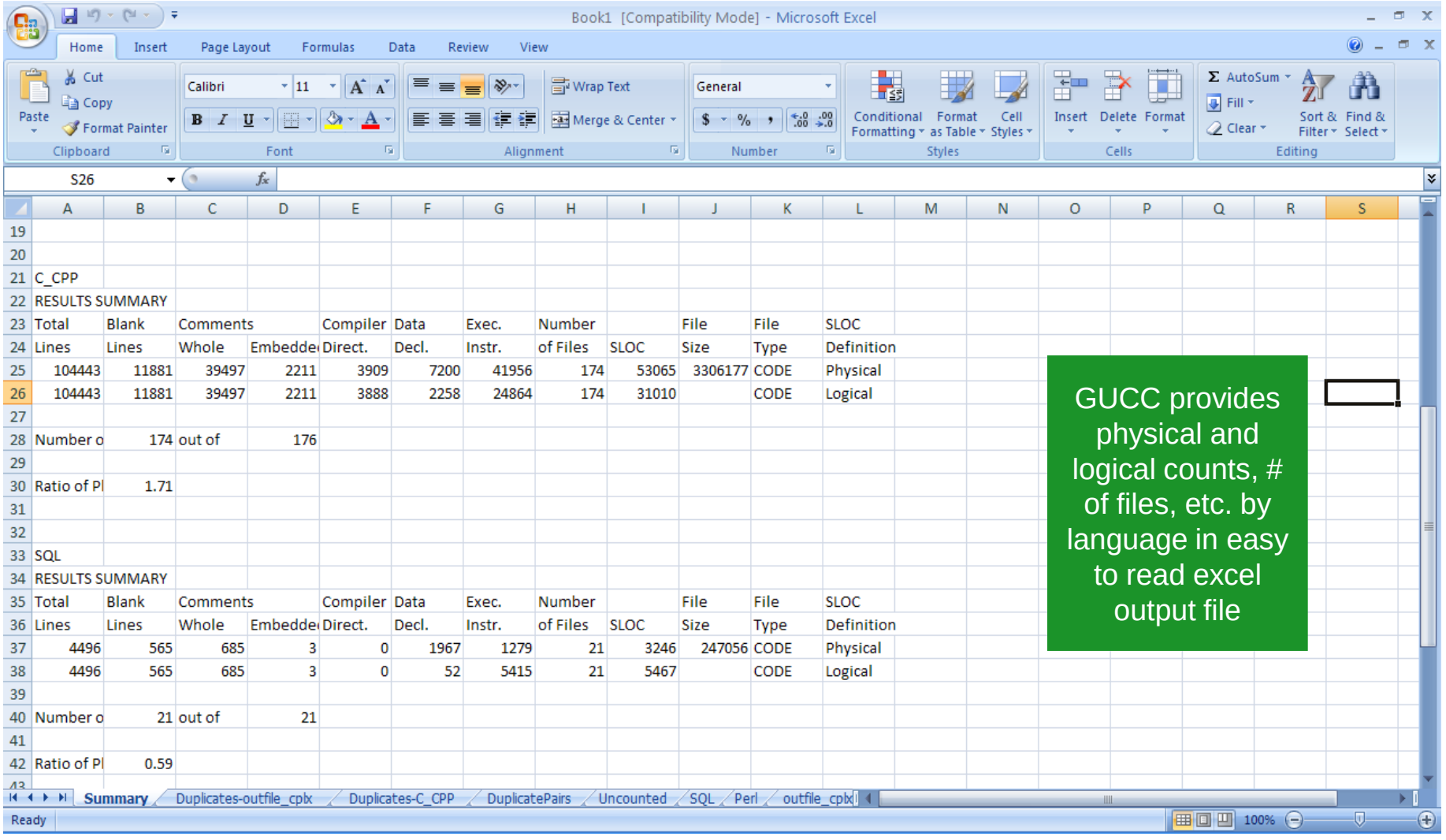
Simply locate the folder where the code is

Add another baseline if using the difference tool

Press 'Execute' to start the GUCC

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics Since 1973

Summary Results Screen



	A	B	C	D	E	F	G	H	I	J	K	L
19												
20												
21	C_CPP											
22	RESULTS SUMMARY											
23	Total	Blank	Comments		Compiler	Data	Exec.	Number		File	File	SLOC
24	Lines	Lines	Whole	Embedde	Direct.	Decl.	Instr.	of Files	SLOC	Size	Type	Definition
25	104443	11881	39497	2211	3909	7200	41956	174	53065	3306177	CODE	Physical
26	104443	11881	39497	2211	3888	2258	24864	174	31010		CODE	Logical
27												
28	Number o	174	out of	176								
29												
30	Ratio of Pl	1.71										
31												
32												
33	SQL											
34	RESULTS SUMMARY											
35	Total	Blank	Comments		Compiler	Data	Exec.	Number		File	File	SLOC
36	Lines	Lines	Whole	Embedde	Direct.	Decl.	Instr.	of Files	SLOC	Size	Type	Definition
37	4496	565	685	3	0	1967	1279	21	3246	247056	CODE	Physical
38	4496	565	685	3	0	52	5415	21	5467		CODE	Logical
39												
40	Number o	21	out of	21								
41												
42	Ratio of Pl	0.59										

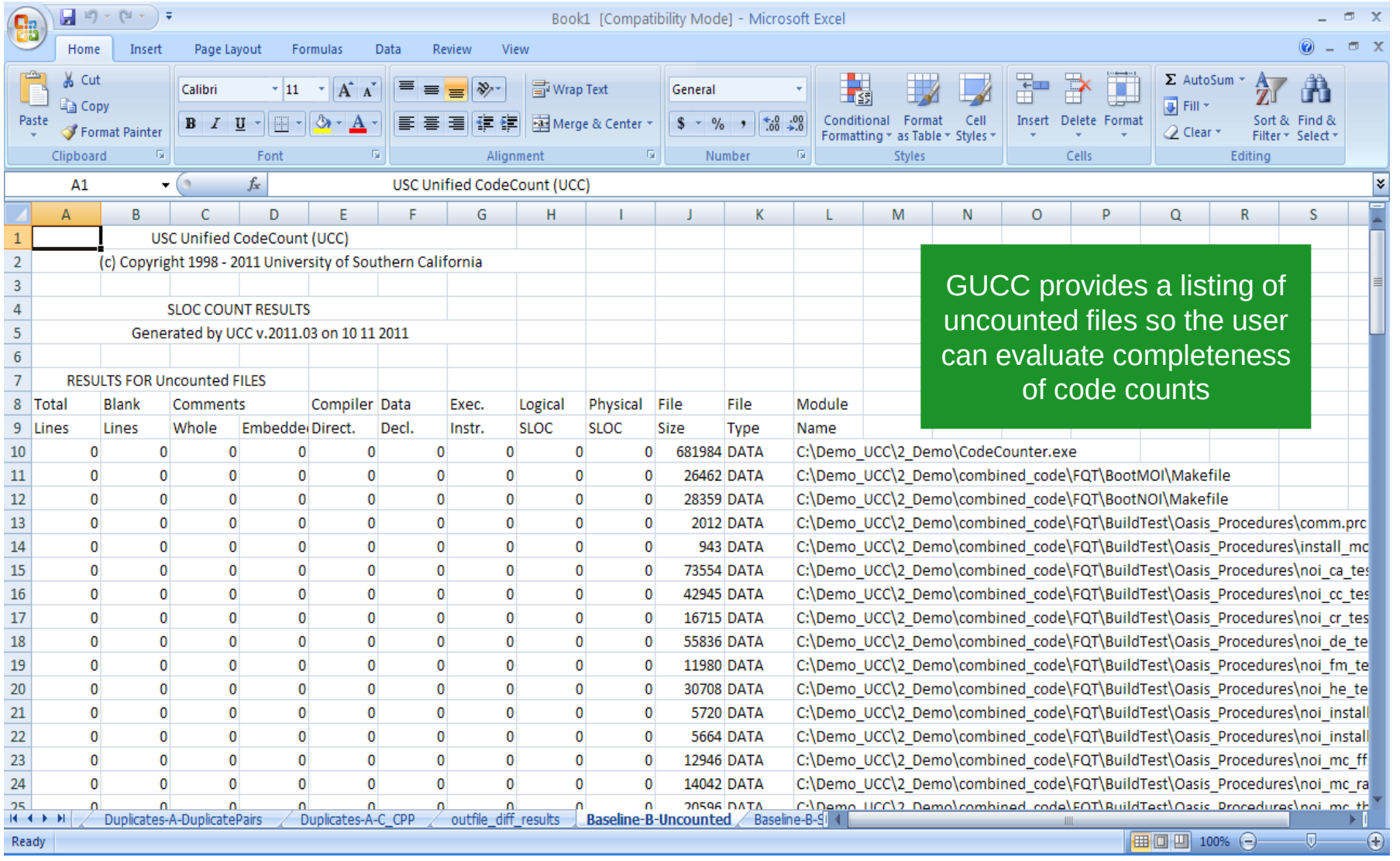
Summary | Duplicates-outfile_cplx | Duplicates-C_CPP | DuplicatePairs | Uncounted | SQL | Perl | outfile_cplx

Ready 100%

GUCC provides physical and logical counts, # of files, etc. by language in easy to read excel output file

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics Since 1973

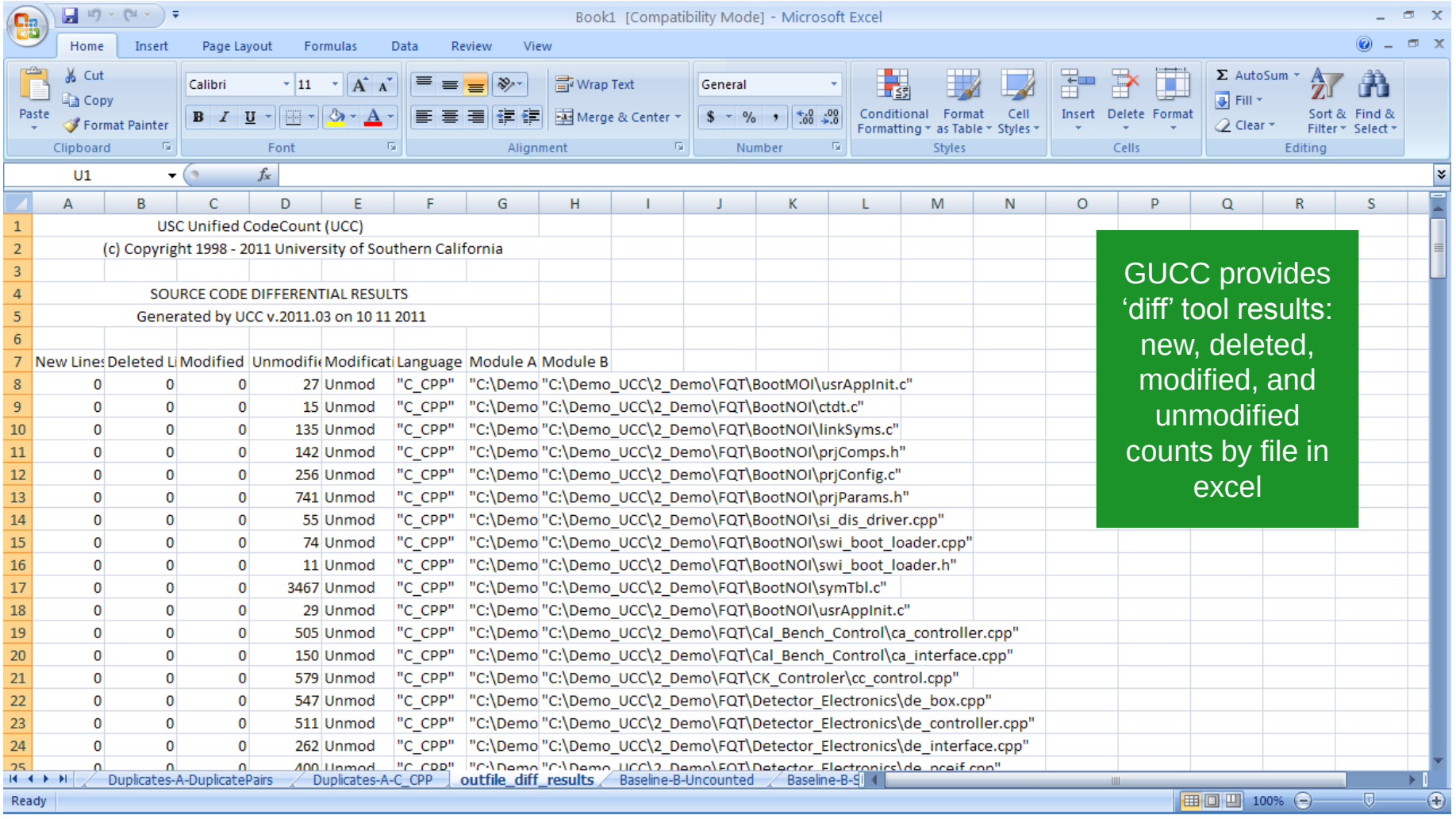
GUCC Detailed Report Screen



GUCC provides a listing of uncounted files so the user can evaluate completeness of code counts

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics Since 1973

GUCC Detailed Report Screen



Book1 [Compatibility Mode] - Microsoft Excel

	A	B	C	D	E	F	G	H
1	USC Unified CodeCount (UCC)							
2	(c) Copyright 1998 - 2011 University of Southern California							
3								
4	SOURCE CODE DIFFERENTIAL RESULTS							
5	Generated by UCC v.2011.03 on 10 11 2011							
6								
7	New Lines	Deleted Li	Modified	Unmodifie	Modificat	Language	Module A	Module B
8	0	0	0	27	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootMOI\usrAppInit.c"
9	0	0	0	15	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\ctdt.c"
10	0	0	0	135	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\linkSyms.c"
11	0	0	0	142	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\prjComps.h"
12	0	0	0	256	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\prjConfig.c"
13	0	0	0	741	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\prjParams.h"
14	0	0	0	55	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\si_dis_driver.cpp"
15	0	0	0	74	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\swi_boot_loader.cpp"
16	0	0	0	11	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\swi_boot_loader.h"
17	0	0	0	3467	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\symTbl.c"
18	0	0	0	29	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\BootNOI\usrAppInit.c"
19	0	0	0	505	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\Cal_Bench_Control\ca_controller.cpp"
20	0	0	0	150	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\Cal_Bench_Control\ca_interface.cpp"
21	0	0	0	579	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\CK_Controler\cc_control.cpp"
22	0	0	0	547	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\Detector_Electronics\de_box.cpp"
23	0	0	0	511	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\Detector_Electronics\de_controller.cpp"
24	0	0	0	262	Unmod	"C_CPP"	"C:\Demo	"C:\Demo_UCC\2_Demo\FQT\Detector_Electronics\de_interface.cpp"

Duplicates-A-DuplicatePairs | Duplicates-A-C_CPP | outfile_diff_results | Baseline-B-Uncounted | Baseline-B-S

Ready 100%

GUCC provides 'diff' tool results: new, deleted, modified, and unmodified counts by file in excel

Presented at the 2012 SCEA/ISPA Joint Annual Conference and Training Workshop - www.iceaaonline.com

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics
Since 1973

GUCC Demo

TECOLOTE RESEARCH, INC.
Bridging Engineering and Economics
Since 1973

Summary

Summary

- **The GUCC operates in a Windows OS environment and reports results in MS Excel**
  - Analyst/tester comment: "Ok, the results are in: GUCC is pretty fantastic. You can count all languages by using the generic file filter of "*.*" so that you don't have to count each language discretely. Plus, it makes you a nice automated summary tab that shows counts per language, while including all of the output files so that you can go back and look at details for each count."
- **Many software development efforts operate in a UNIX environment**
  - UNIX GUI for the UCC reportedly developed by NG
- **Working with Intelligence Community to share information and demo tools**
- **Scheduling a demonstration to USC for possible incorporation in future UCC versions**