Function Fault Injector

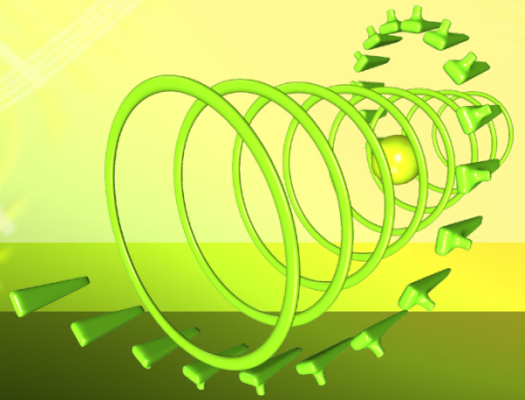
A tool to test your exception handlers

8 Weeks Summer Training
At Microsoft IDC Hyderabad.

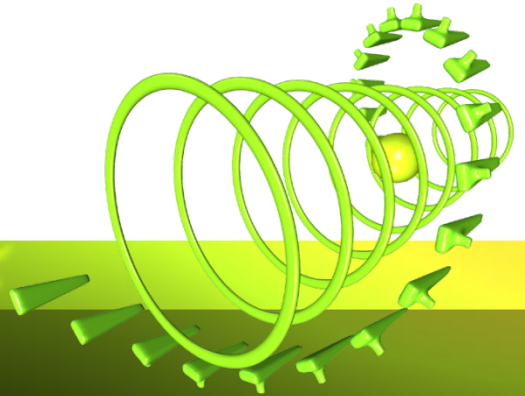
Atishay Jain
10783017
3-T5

Index

- About Fault Injection.
- About C# and Visual Studio IDE as well as WinCE and the Platform Builder.
- Application Verifier and the FFI & their working.
- The FFI features.
- DFD
- Working, Coding & Testing.
- Results & Scope of Improvement.

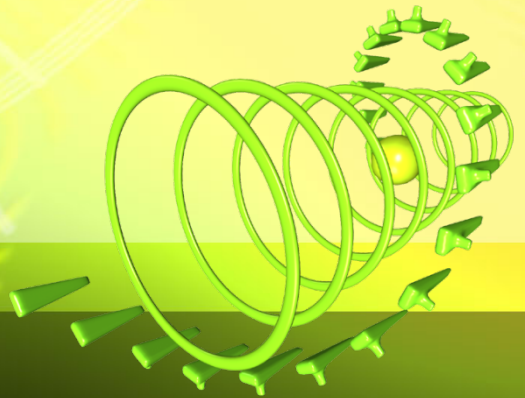


About Fault Injection

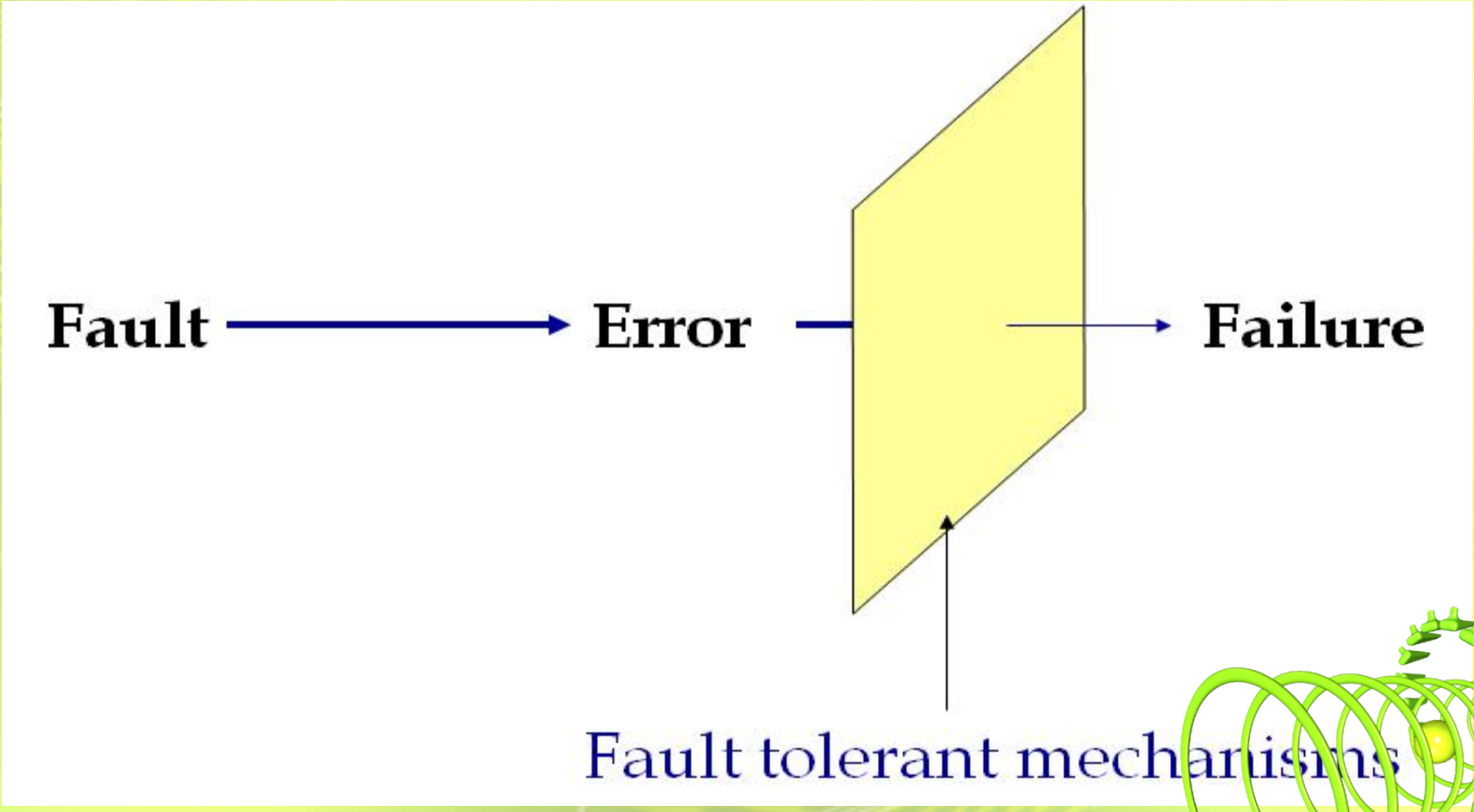


Definition

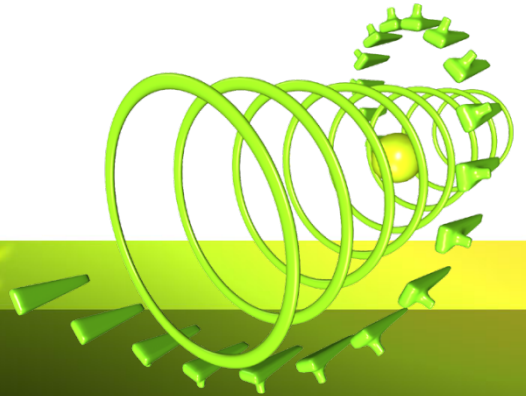
"Deliberate insertion of upsets (faults or errors) in computer systems to evaluate its behavior in the presence of faults or validate specific fault tolerance mechanisms in computers."



What is being tested?

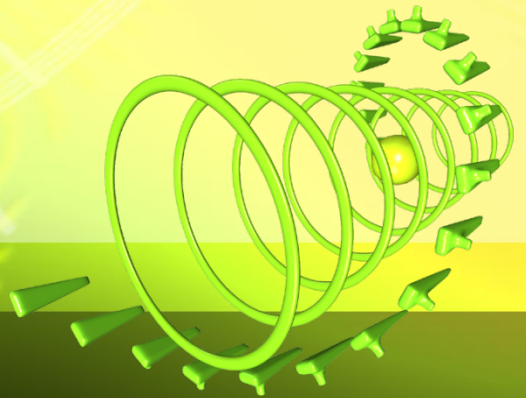


Development Environment



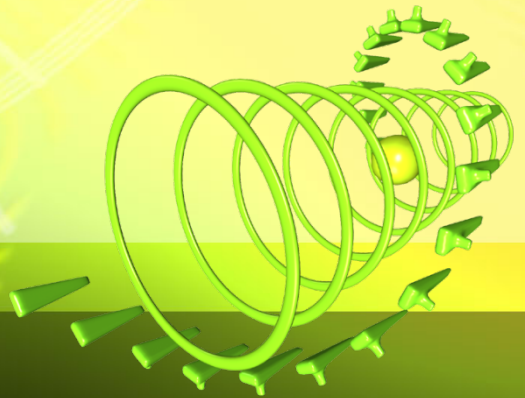
the language...

- Object Oriented.
- Windows Friendly.
- Powerful.
- Integrated into the Visual Studio Suite.
- Popular.
- The language of the reusable components.



Visual Studio

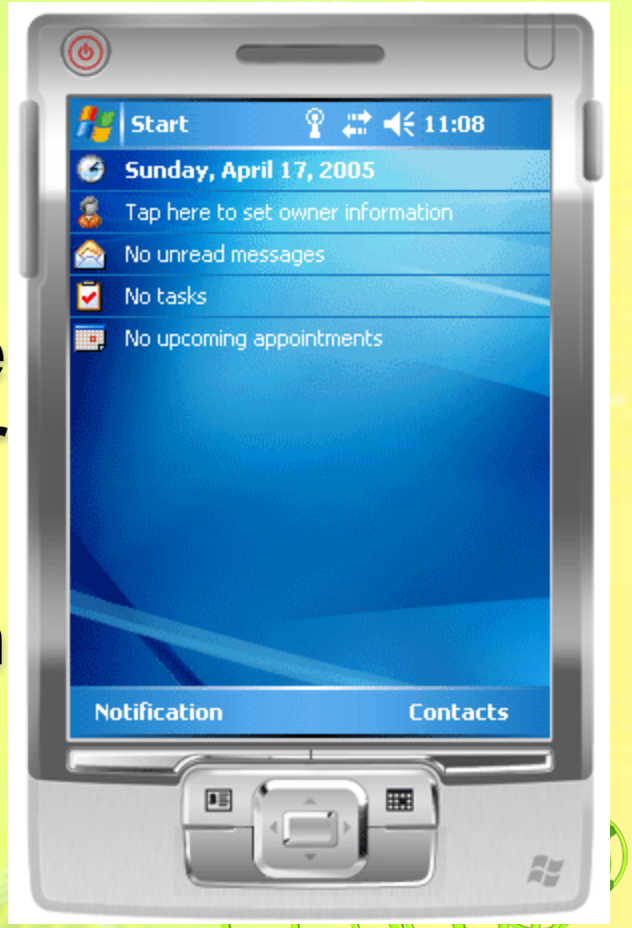
- User Friendly Development IDE.
- Intellisense.
- Debugging features like breakpoint and walk into code.
- Support for direct connection into Windows CE devices.
- Code Generation.



Microsoft Windows CE

FFI

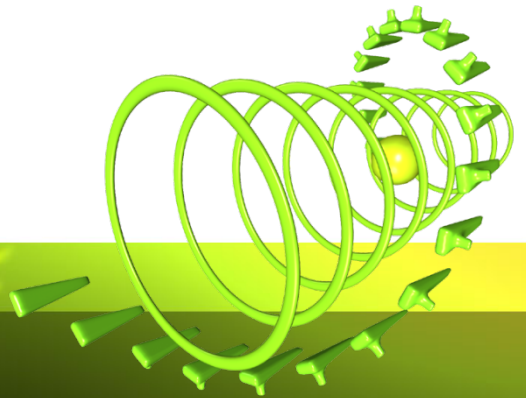
- Target OS.
- Required for Application testing.
- Directly testing of code via the use of the mobile emulator available with the Visual Studio.
- Windows programming on reduced ARMv4 architecture.



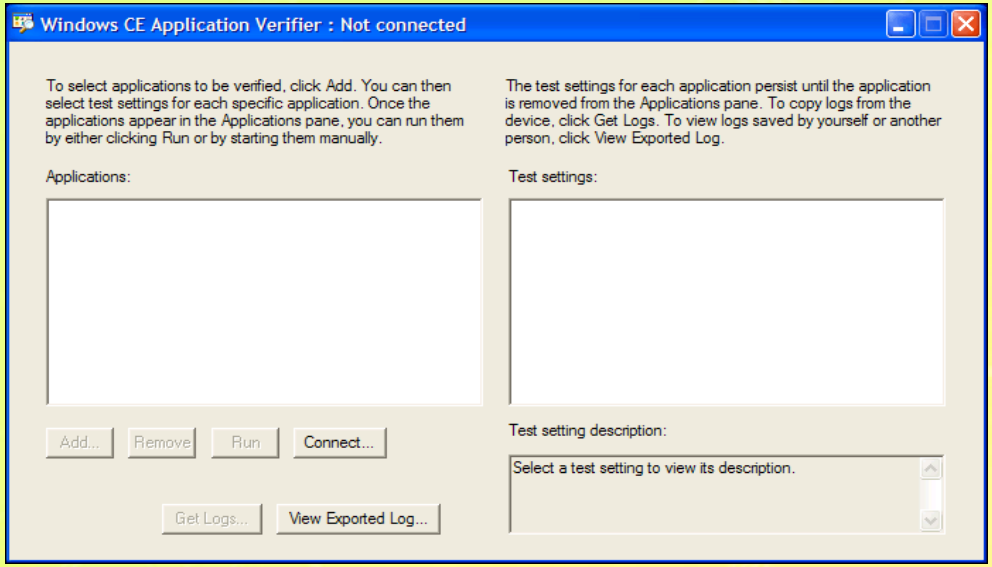
Platform Builder

- Makes OS for windows mobile device with embedded software.
- Contained the code to be tested.
- The target software development environment as well as the environment to connect to the actual device.

Application Verifier & FFI

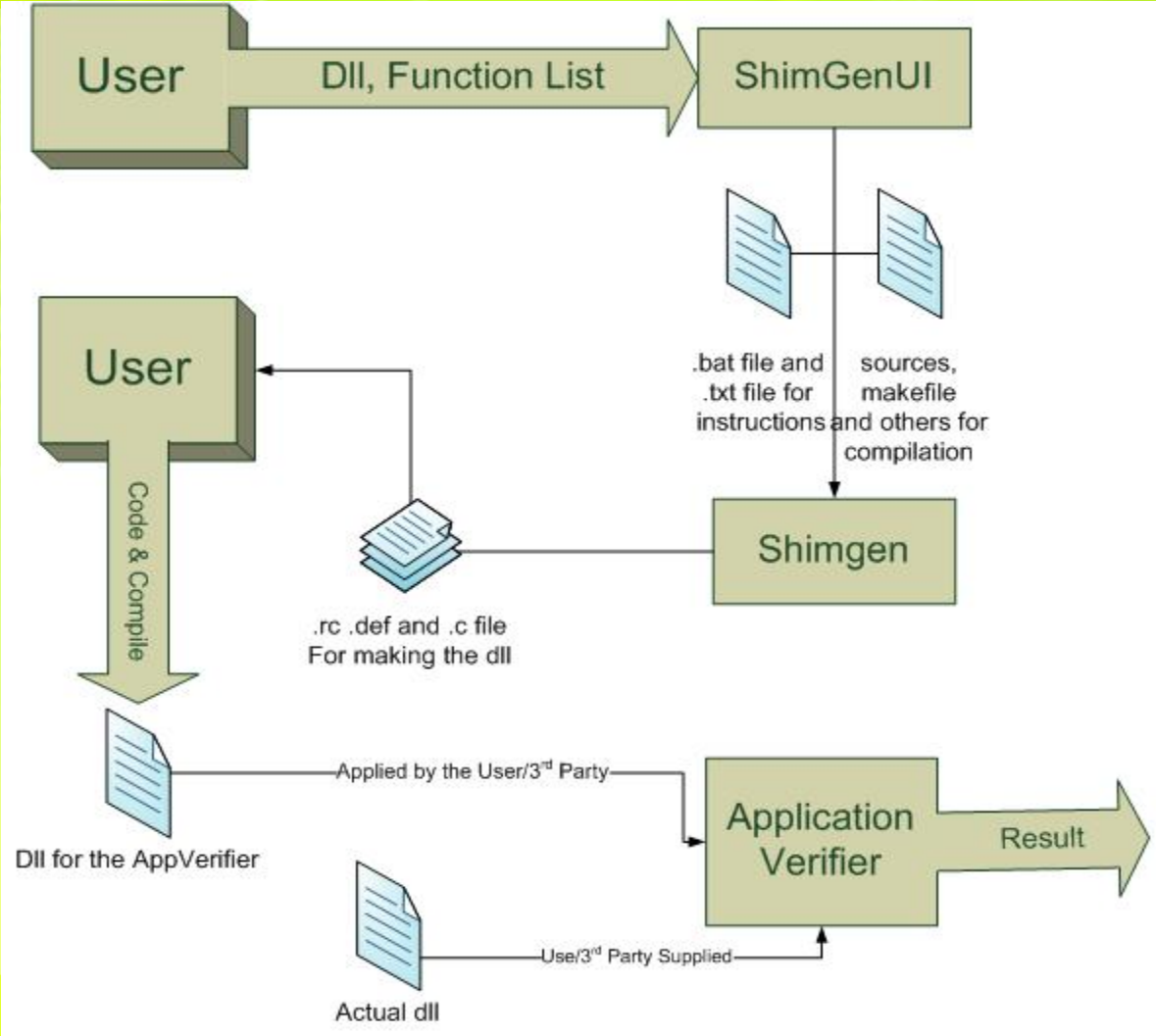


Application Verifier

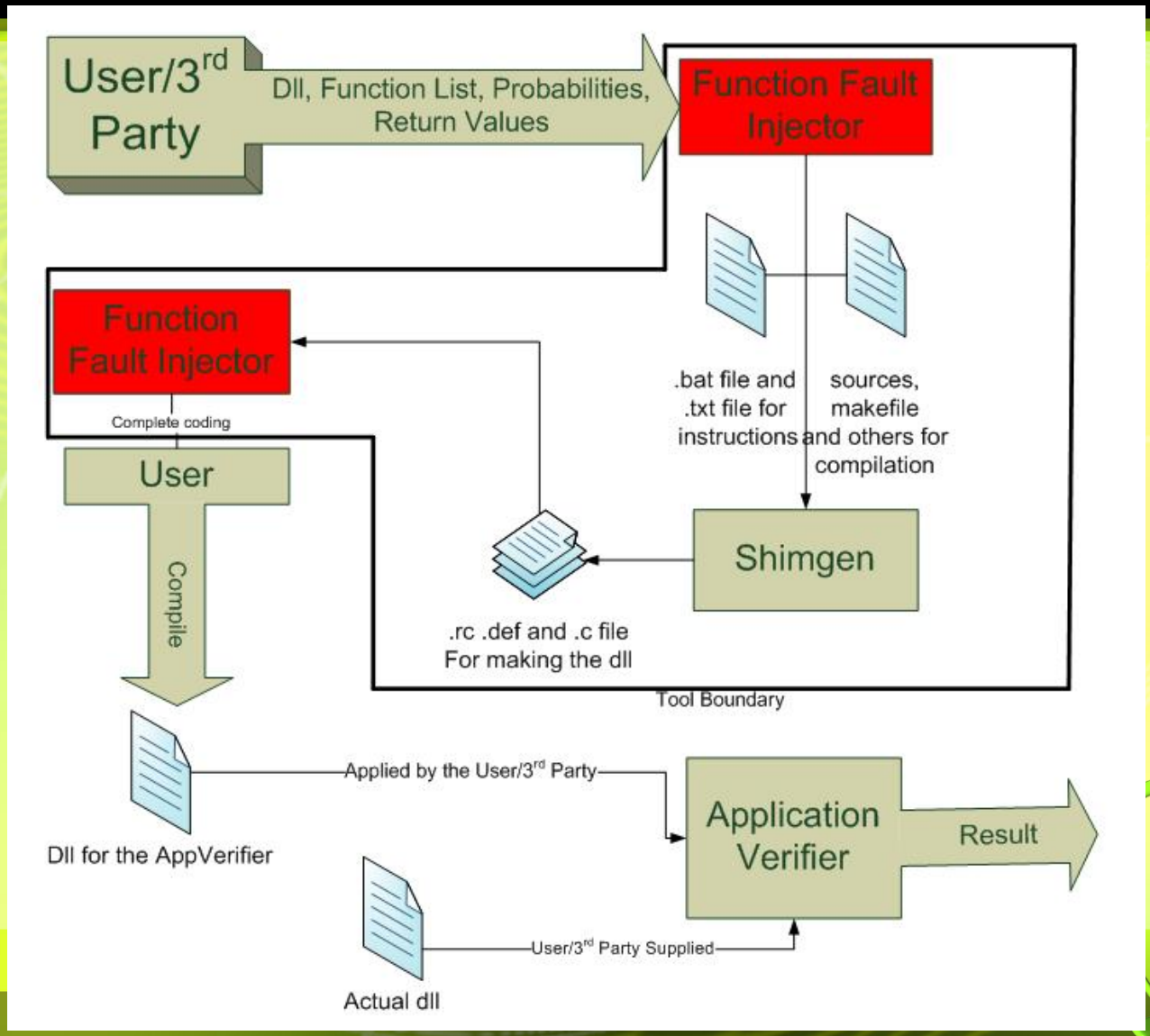


- Freeware.
- Has both x86 as well as Arm versions.
- A lot of help available online.
- Can easily alter the code and disrupt calls.
- Already used as fault injector for many type of faults.

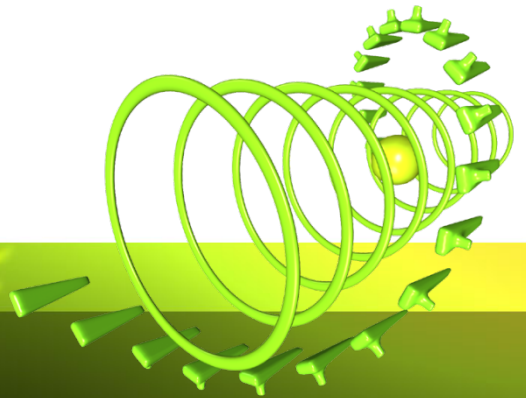
Working-Appverifier



Working-FFI

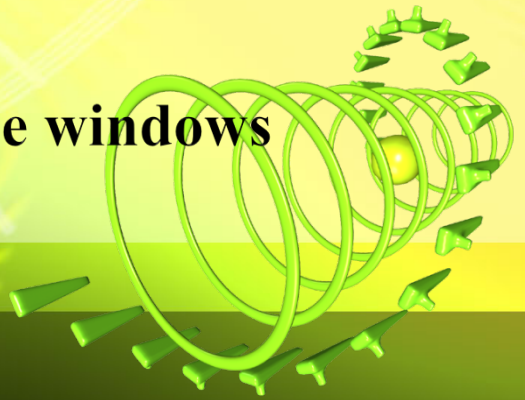

User/3rd Party
Dll, Function List, Probabilities, Return Values
Function Fault Injector
Function Fault Injector
Complete coding
User
Compile
.bat file and .txt file for instructions
sources, makefile and others for compilation
Shimgen
.rc .def and .c file For making the dll
Tool Boundary
Dll for the AppVerifier
Applied by the User/3rd Party
Application Verifier
Result
Actual dll
User/3rd Party Supplied

Features



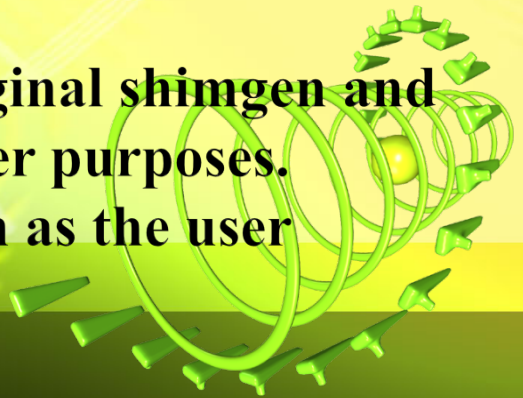
Features

1. Intercept the program flow (via the application verifier) and pass the original function to the shimmed dll where the following can be applied:
   a. Give alternate Return Value.
   b. Fill in a stub function, to replace the original one.
   c. Change the passed parameters to the original function return the result produced henceforth.
   d. Wait for the original function to complete and modify the output or the return value after the function has ended its flow.
2. Have probabilities associated with each type of return and also with the original function.
3. Modify the probabilities dynamically, through the windows registry.
4. No injection into the original dll.

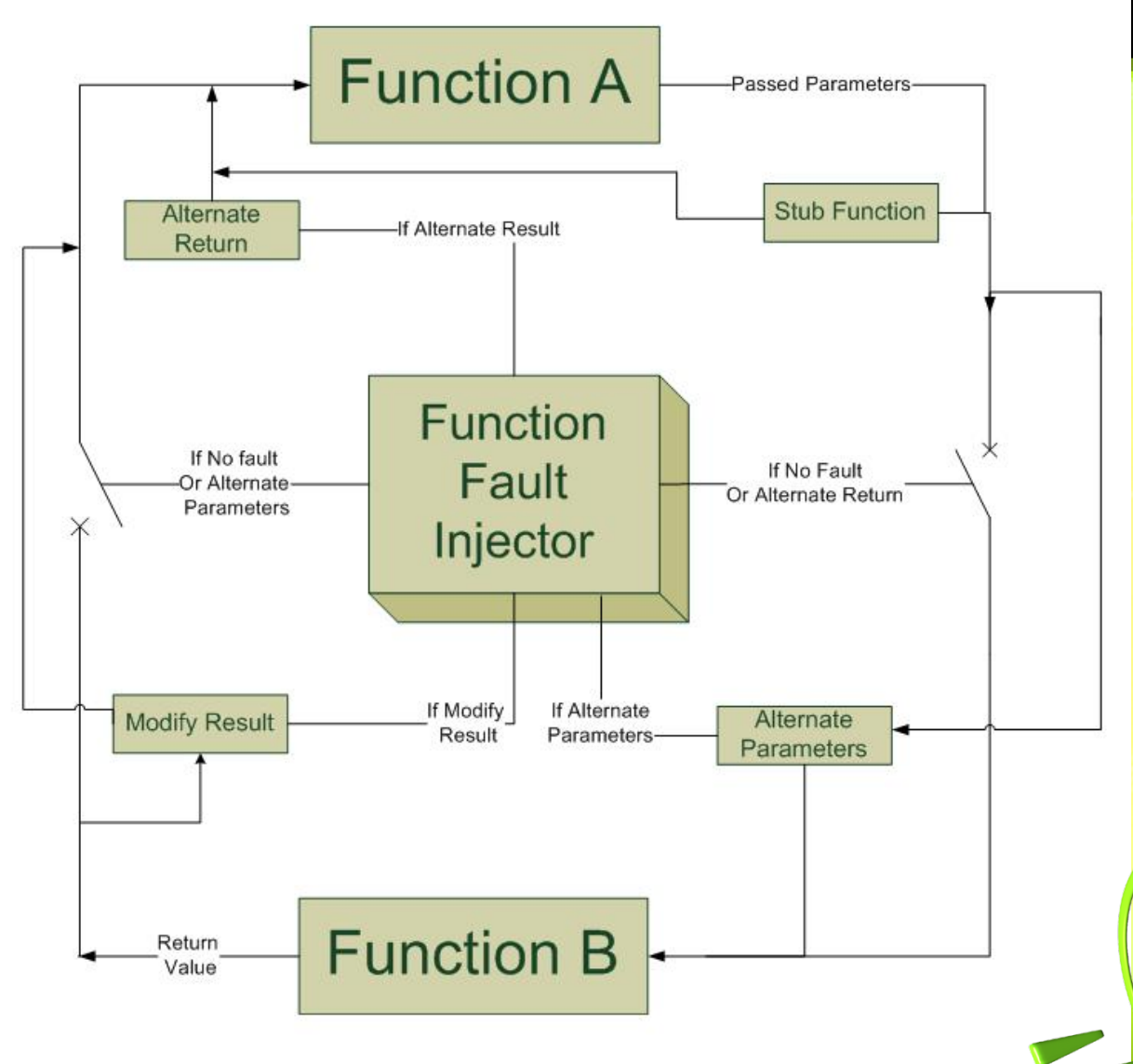


Features

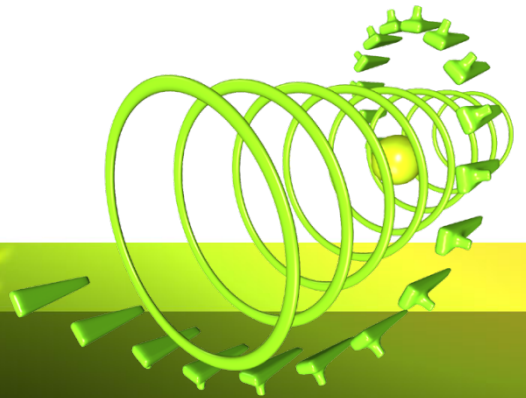
5. Can remove the shim from application via just one command.
6. Save and modify the project any time and apply the modified shim by just replacing the old one.
7. Full flexibility to write any C code and include custom headers.
8. Can also be used for API testing through the modify passed values or parameters option.
9. System dlls can be shimmed to produce other types of faults by say restricting the memory available.
10. A lot of time is saved as the tool automatically generates the code and the support files which in normal shimming process takes much more time.
11. Free form many leaks and flaws present in the original shimgen and hence a good alternative to produce new shims for other purposes.
12. Leaves the C files uncompiled for any modification as the user wants.



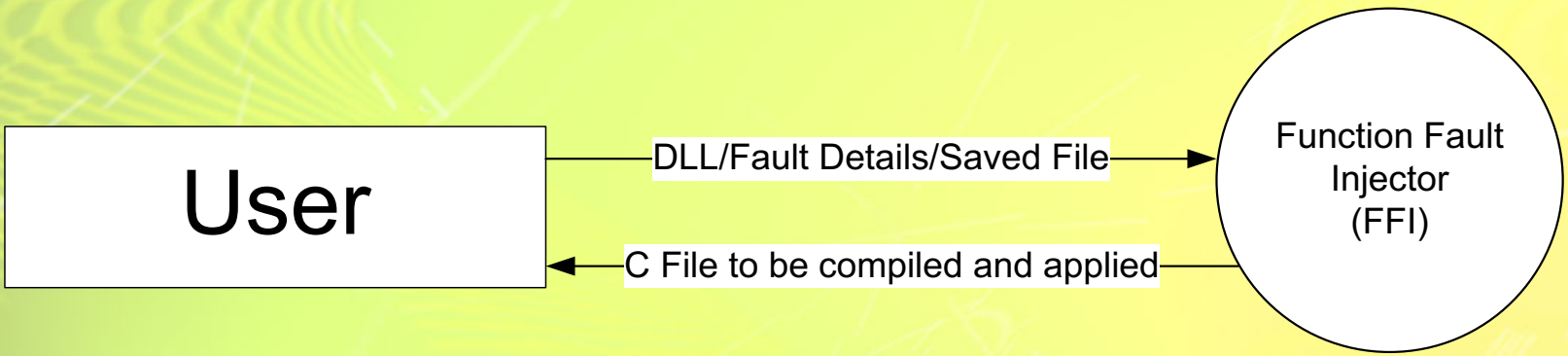
Features


Function A
Passed Parameters
Alternate Return
If Alternate Result
Stub Function
Function Fault Injector
If No fault Or Alternate Parameters
If No Fault Or Alternate Return
Modify Result
If Modify Result
If Alternate Parameters
Alternate Parameters
Return Value
Function B

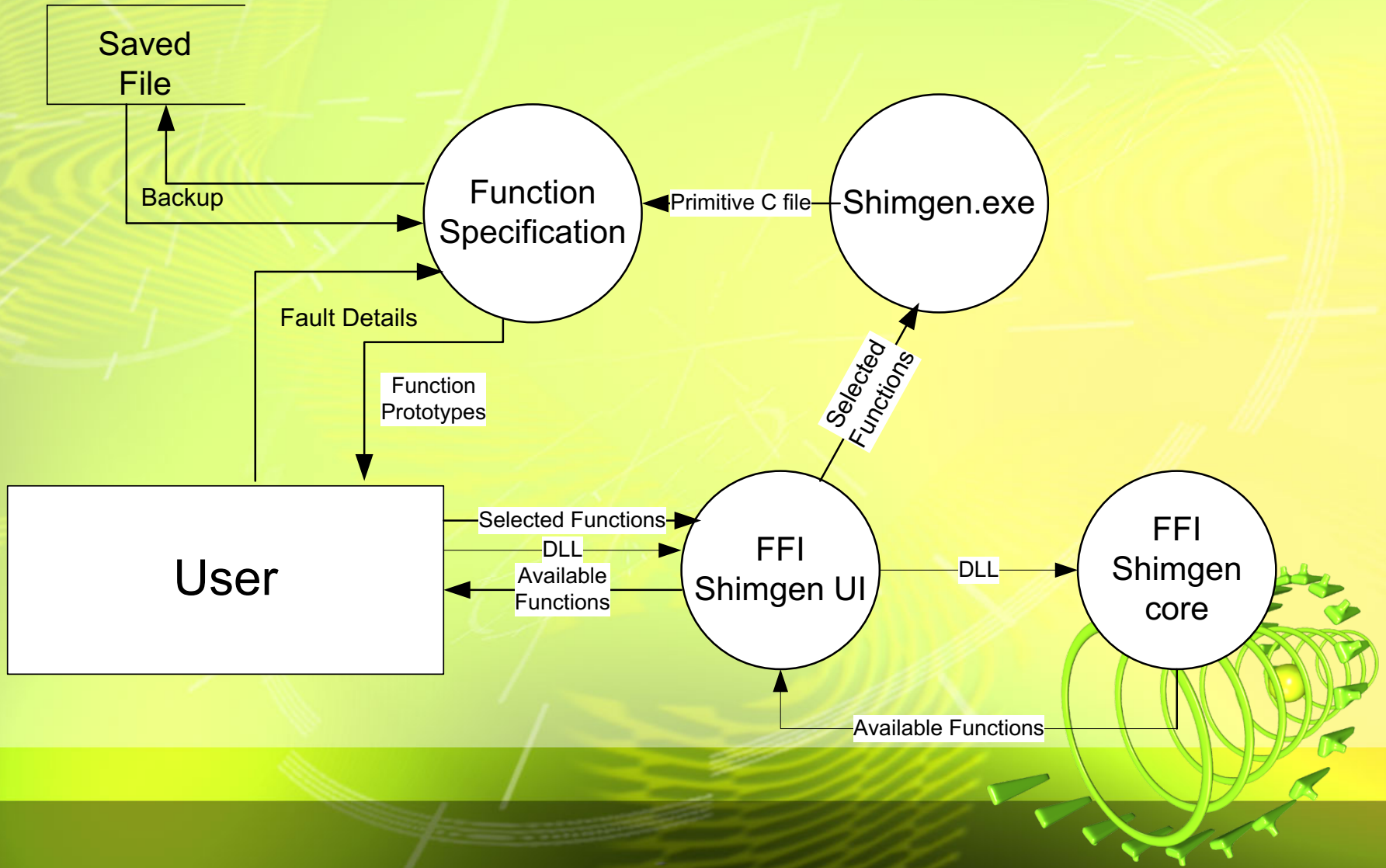
DFDs



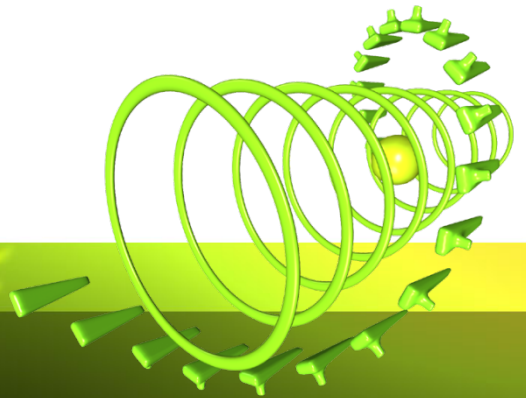
Context Diagram



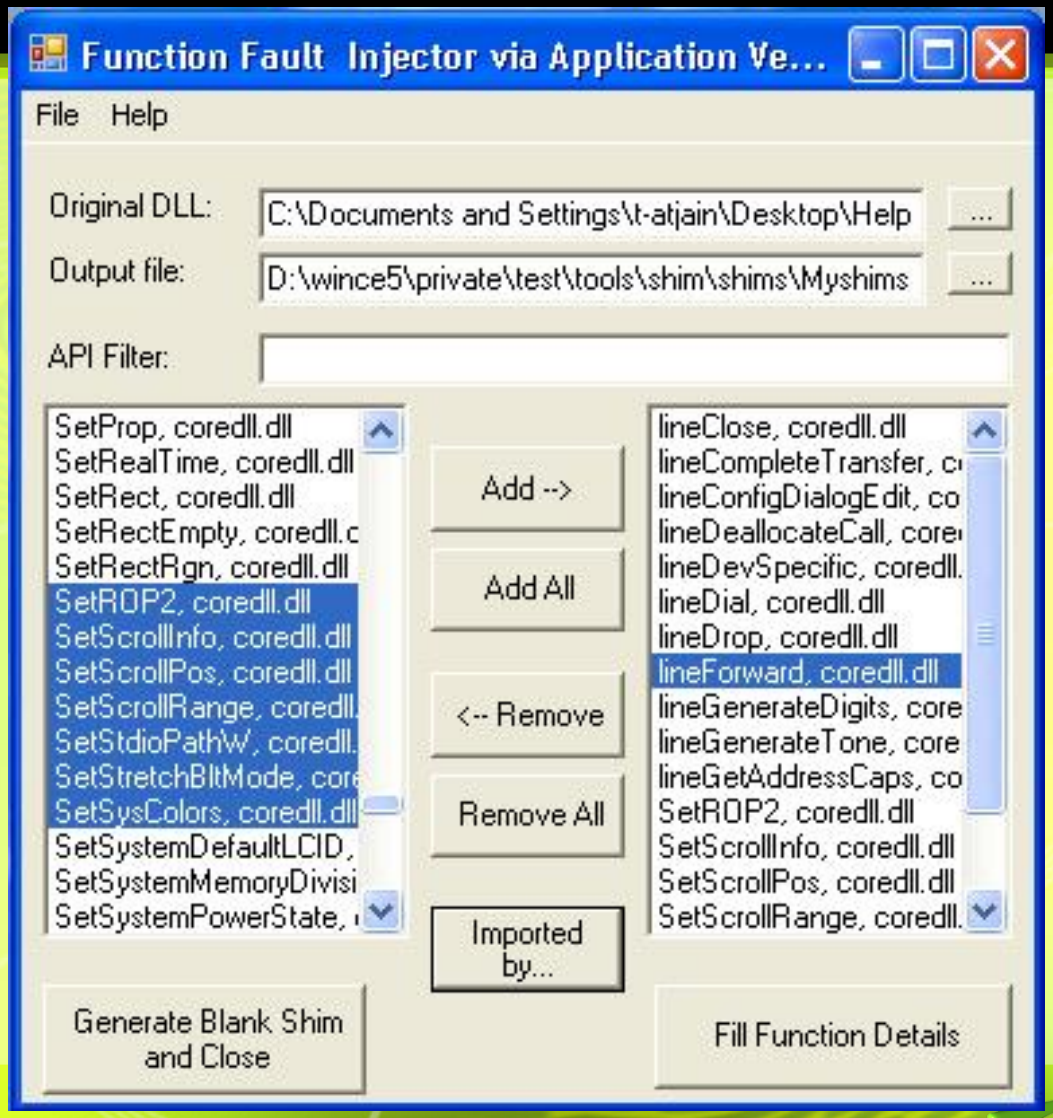
Level 1 DFD


Saved File
Backup
Function Specification
Primitive C file
Shimgen.exe
Fault Details
Function Prototypes
Selected Functions
User
Selected Functions
DLL
Available Functions
FFI Shimgen UI
DLL
FFI Shimgen core
Available Functions

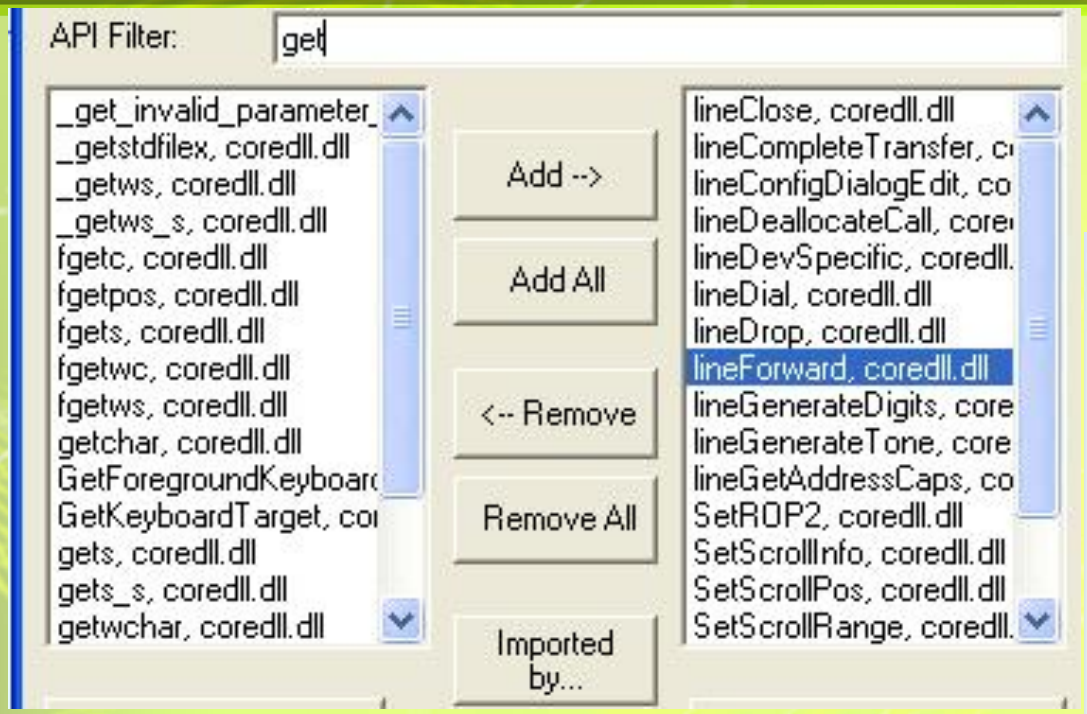
Working, Coding & Testing

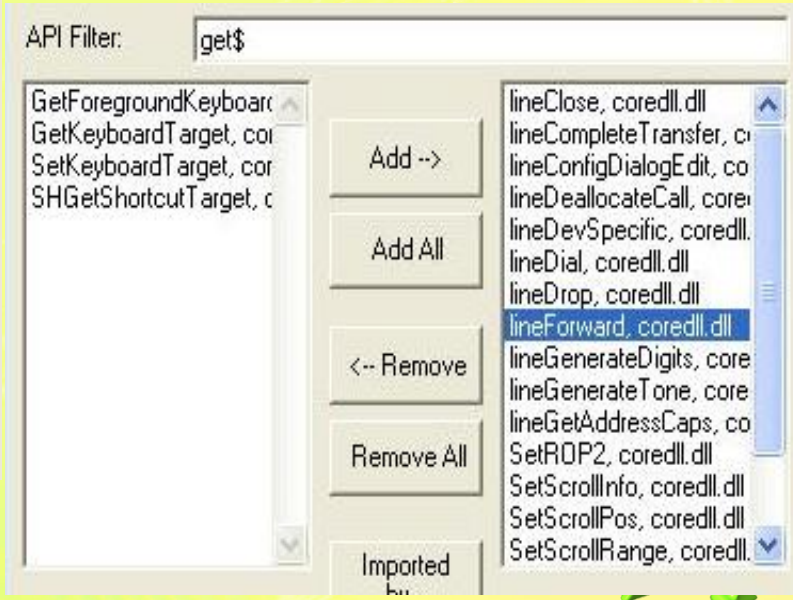


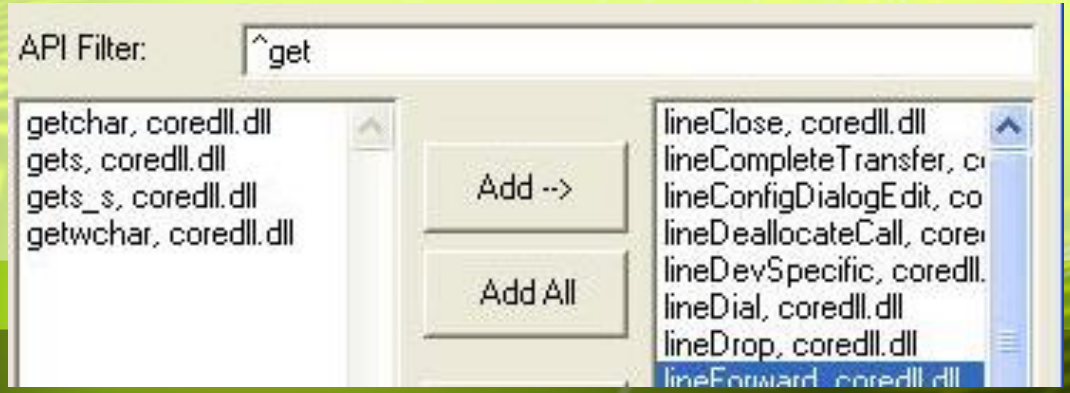
Function Selection UI



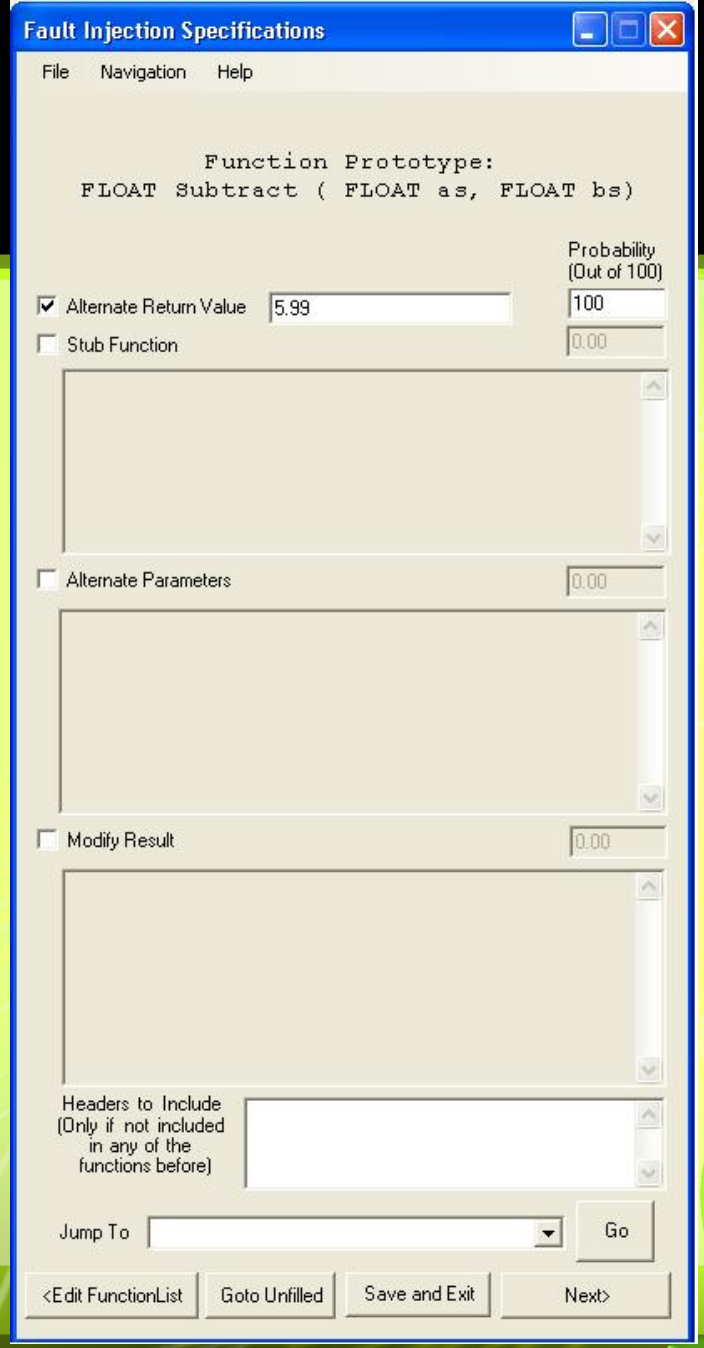
API Filter


API Filter:
get
_get_invalid_parameter_
_getstdfilex, coredll.dll
_getws, coredll.dll
_getws_s, coredll.dll
fgetc, coredll.dll
fgetpos, coredll.dll
fgets, coredll.dll
fgetwc, coredll.dll
fgetws, coredll.dll
getchar, coredll.dll
GetForegroundKeyboard
GetKeyboardTarget, cor
gets, coredll.dll
gets_s, coredll.dll
getwchar, coredll.dll
Add -->
Add All
<-- Remove
Remove All
Imported by...
lineClose, coredll.dll
lineCompleteTransfer, c
lineConfigDialogEdit, co
lineDeallocateCall, core
lineDevSpecific, coredll.
lineDial, coredll.dll
lineDrop, coredll.dll
lineForward, coredll.dll
lineGenerateDigits, core
lineGenerateTone, core
lineGetAddressCaps, co
SetROP2, coredll.dll
SetScrollInfo, coredll.dll
SetScrollPos, coredll.dll
SetScrollRange, coredll.

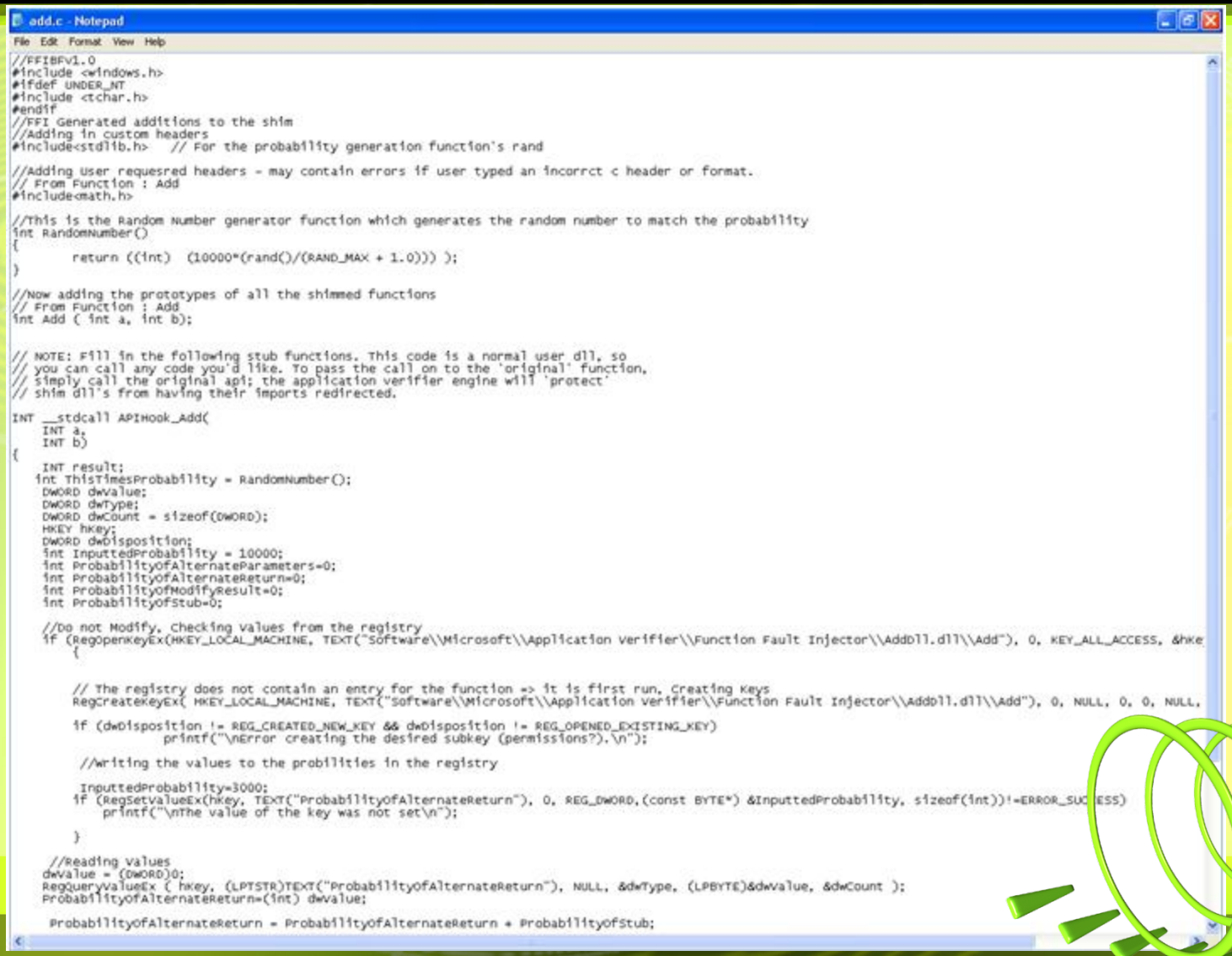

API Filter:
get$
GetForegroundKeyboard
GetKeyboardTarget, cor
SetKeyboardTarget, cor
SHGetShortcutTarget, c
Add -->
Add All
<-- Remove
Remove All
Imported
lineClose, coredll.dll
lineCompleteTransfer, c
lineConfigDialogEdit, co
lineDeallocateCall, core
lineDevSpecific, coredll.
lineDial, coredll.dll
lineDrop, coredll.dll
lineForward, coredll.dll
lineGenerateDigits, core
lineGenerateTone, core
lineGetAddressCaps, co
SetROP2, coredll.dll
SetScrollInfo, coredll.dll
SetScrollPos, coredll.dll
SetScrollRange, coredll.


API Filter:
^get
getchar, coredll.dll
gets, coredll.dll
gets_s, coredll.dll
getwchar, coredll.dll
Add -->
Add All
lineClose, coredll.dll
lineCompleteTransfer, c
lineConfigDialogEdit, co
lineDeallocateCall, core
lineDevSpecific, coredll.
lineDial, coredll.dll
lineDrop, coredll.dll

Filling Fault Details



Generated Files



add.c - Notepad

File Edit Format View Help

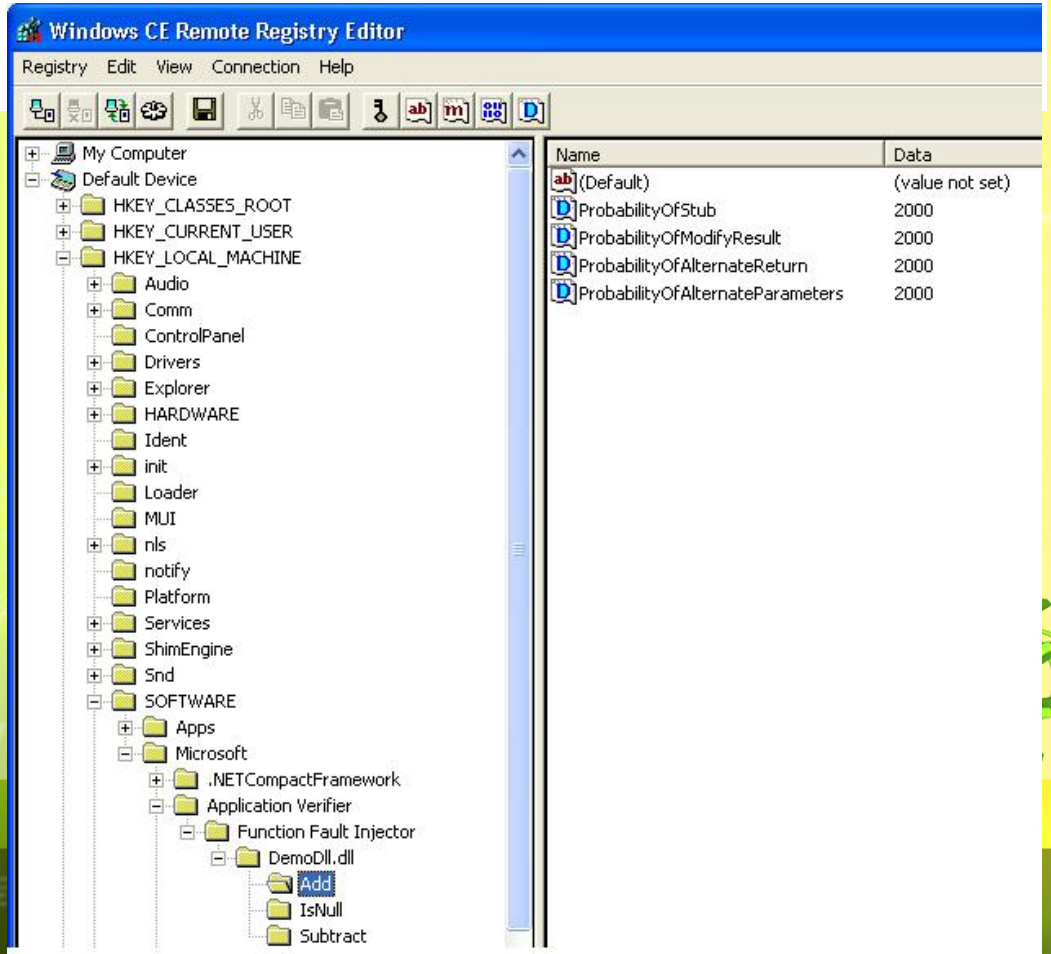
```
//FFIBFv1.0
#include <windows.h>
#ifdef UNDER_NT
#include <tchar.h>
#endif
//FFI Generated additions to the shim
//Adding in custom headers
#include<stdlib.h>   // For the probability generation function's rand

//Adding User requesred headers - may contain errors if user typed an incorrct c header or format.
// From Function : Add
#include<math.h>

//This is the Random Number generator function which generates the random number to match the probability
int RandomNumber()
{
        return ((int)  (10000*(rand()/(RAND_MAX + 1.0))) );
}

//Now adding the prototypes of all the shimmed functions
// From Function : Add
int Add ( int a, int b);


// NOTE: Fill in the following stub functions. This code is a normal user dll, so
// you can call any code you'd like. To pass the call on to the 'original' function,
// simply call the original api; the application verifier engine will 'protect'
// shim dll's from having their imports redirected.

INT __stdcall APIHook_Add(
    INT a,
    INT b)
{
    INT result;
   int ThisTimesProbability = RandomNumber();
    DWORD dwValue;
    DWORD dwType;
    DWORD dwCount = sizeof(DWORD);
    HKEY hKey;
    DWORD dwDisposition;
    int InputtedProbability = 10000;
    int ProbabilityOfAlternateParameters=0;
    int ProbabilityOfAlternateReturn=0;
    int ProbabilityOfModifyResult=0;
    int ProbabilityOfStub=0;

    //Do not Modify, Checking values from the registry
    if (RegOpenKeyEx(HKEY_LOCAL_MACHINE, TEXT("Software\\Microsoft\\Application Verifier\\Function Fault Injector\\AddDll.dll\\Add"), 0, KEY_ALL_ACCESS, &hKe
        {


        // The registry does not contain an entry for the function => it is first run, Creating Keys
        RegCreateKeyEx( HKEY_LOCAL_MACHINE, TEXT("Software\\Microsoft\\Application Verifier\\Function Fault Injector\\AddDll.dll\\Add"), 0, NULL, 0, 0, NULL,

        if (dwDisposition != REG_CREATED_NEW_KEY && dwDisposition != REG_OPENED_EXISTING_KEY)
                printf("\nError creating the desired subkey (permissions?).\n");

         //Writing the values to the probilities in the registry

         InputtedProbability=3000;
        if (RegSetValueEx(hKey, TEXT("ProbabilityOfAlternateReturn"), 0, REG_DWORD,(const BYTE*) &InputtedProbability, sizeof(int))!=ERROR_SUCCESS)
            printf("\nThe value of the key was not set\n");

        }

     //Reading Values
    dwValue = (DWORD)0;
    RegQueryValueEx ( hKey, (LPTSTR)TEXT("ProbabilityOfAlternateReturn"), NULL, &dwType, (LPBYTE)&dwValue, &dwCount );
    ProbabilityOfAlternateReturn=(int) dwValue;

     ProbabilityOfAlternateReturn = ProbabilityOfAlternateReturn + ProbabilityOfStub;
```

Compilation

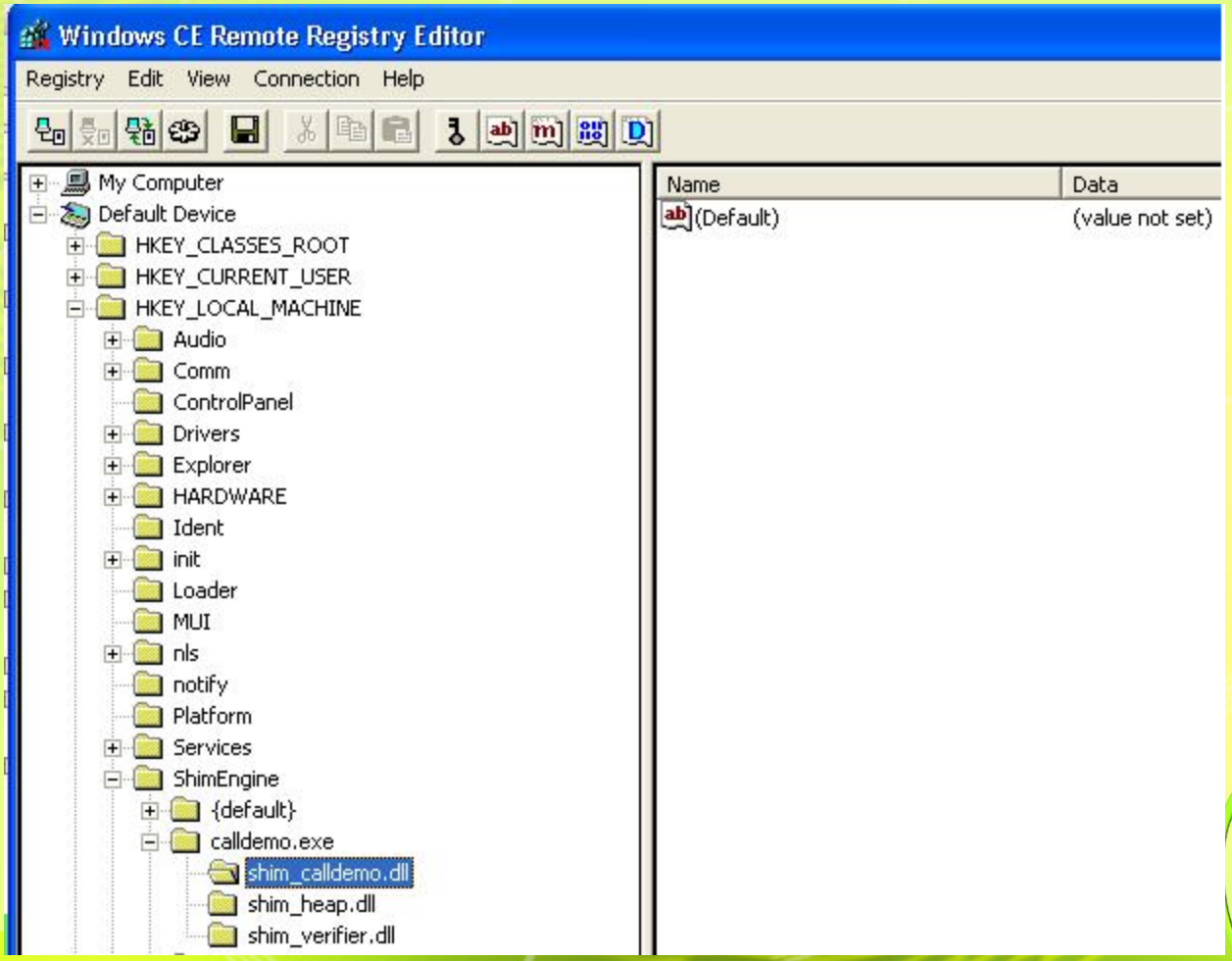

```
BUILD:  [00:0000000153:PROGC ] Saving D:\wince5\private\test\tools\Build.dat.
BUILD:  [00:0000000155:PROGC ] Done.
BUILD:  [00:0000000156:PROGC ]                          Files   Warnings   Errors
BUILD:  [00:0000000157:PROGC ] Midl                         0          0        0
BUILD:  [00:0000000158:PROGC ] Message                      0          0        0
BUILD:  [00:0000000159:PROGC ] Precomp Header               0          0        0
BUILD:  [00:0000000160:PROGC ] Resource                     3          0        0
BUILD:  [00:0000000161:PROGC ] MASM                         0          0        0
BUILD:  [00:0000000162:PROGC ] SHASM                        0          0        0
BUILD:  [00:0000000163:PROGC ] ARMASM                       0          0        0
BUILD:  [00:0000000164:PROGC ] MIPSASM                      0          0        0
BUILD:  [00:0000000165:PROGC ] C++                          0          0        0
BUILD:  [00:0000000166:PROGC ] C                            7          0        0
BUILD:  [00:0000000167:PROGC ] Static Libraries             0          0        0
BUILD:  [00:0000000168:PROGC ] Exe's                        0          0        0
BUILD:  [00:0000000169:PROGC ] Dll's                        2          0        0
BUILD:  [00:0000000170:PROGC ] Preprocess deffile           2          0        0
BUILD:  [00:0000000171:PROGC ] Resx                         0          0        0
BUILD:  [00:0000000172:PROGC ] CSharp Compile               0          0        0
BUILD:  [00:0000000173:PROGC ] Other                        0          0        0
BUILD:  [00:0000000174:PROGC ]
BUILD:  [00:0000000175:PROGC ] Total                       14          0        0
```

Execution & Testing

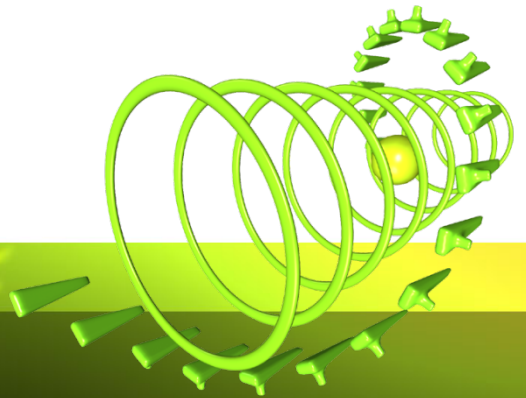
```
Windows CE>appverif -m calldemo.exe -s shim_heap.dll
Verifier loader: SUCCESS
Windows CE>appverif -m calldemo.exe -s shim_calldemo.dll -opt
Verifier loader: SUCCESS
Windows CE>s calldemo.exe
Windows CE>_
```



Execution & Testing

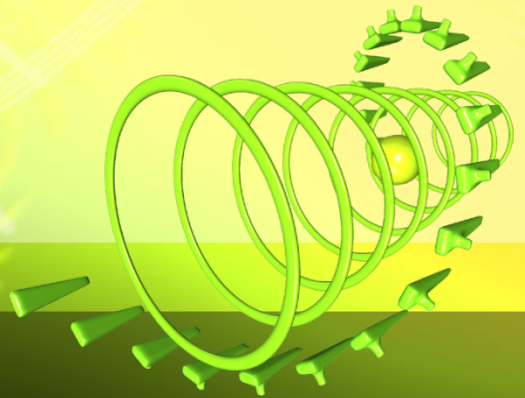


Results



Results and Future scope

- Accepted and Presently under use.
- Embedded as a part of the WinCe 6 testing tools.
- Future scope includes independent compilation of the WinCE code and hence possibly a release as a component of the next version of the Application Verifier.




Thank
You