What is the Basement? Why do I care?
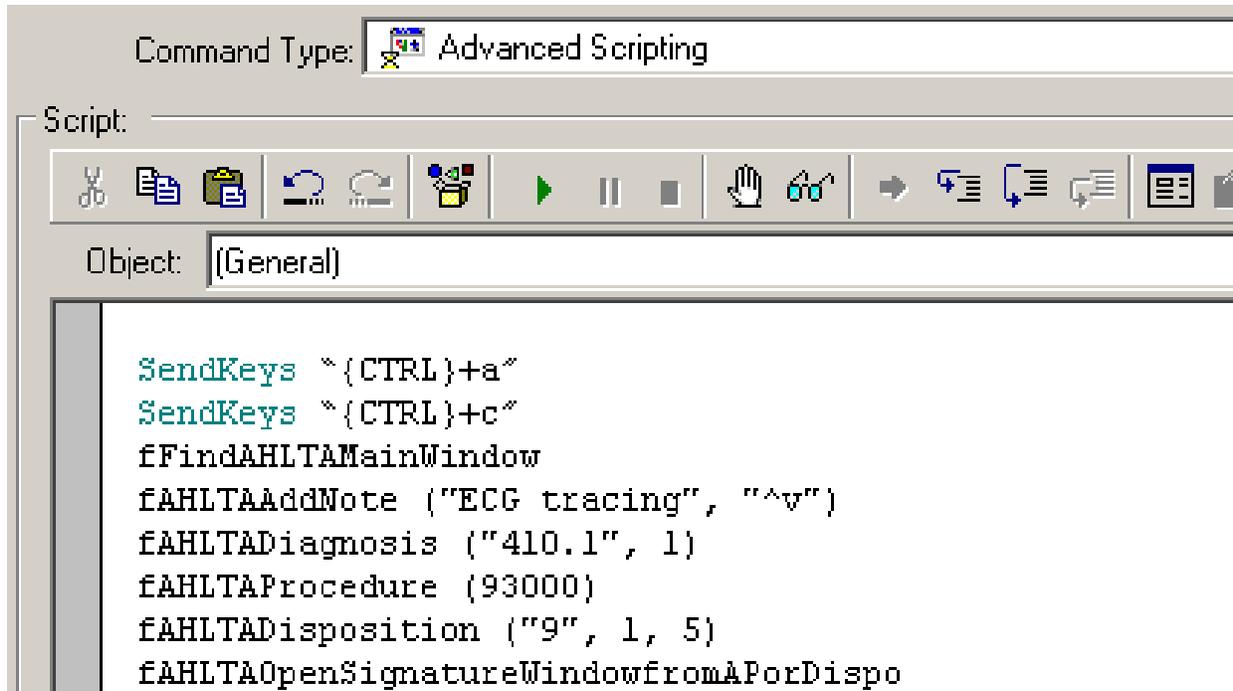
Wouldn't it be nice to perform an entire AHLTA encounter with 8 lines of Dragon script?

```
Command Type: [Advanced Scripting]
Script:
Object: [(General)]

SendKeys "{CTRL}+a"
SendKeys "{CTRL}+c"
fFindAHLTAMainWindow
fAHLTAAddNote ("ECG tracing", "^v")
fAHLTADiagnosis ("410.1", 1)
fAHLTAProcedure (93000)
fAHLTADisposition ("9", 1, 5)
fAHLTAOpenSignatureWindowfromAPorDispo
```

Now an entire encounter is done. This is the concept of the basement: Functions that do all the heavy lifting for you.

Concept/Goals:

1) Provide building block functions to allow basic programmers to access powerful commands and harness our EHRs.
2) Allow centralization of functions to simplify support and allow easy updating without reaching to each macro
3) Migrate Dragon Medical VB script functions to full VB/VBA/HTML platforms.

TECHNICAL PAPER FOR **MAPS** DRAGON FUNCTIONS

Required Set-up:

Say (or run) "**Set Up My Dragon**"

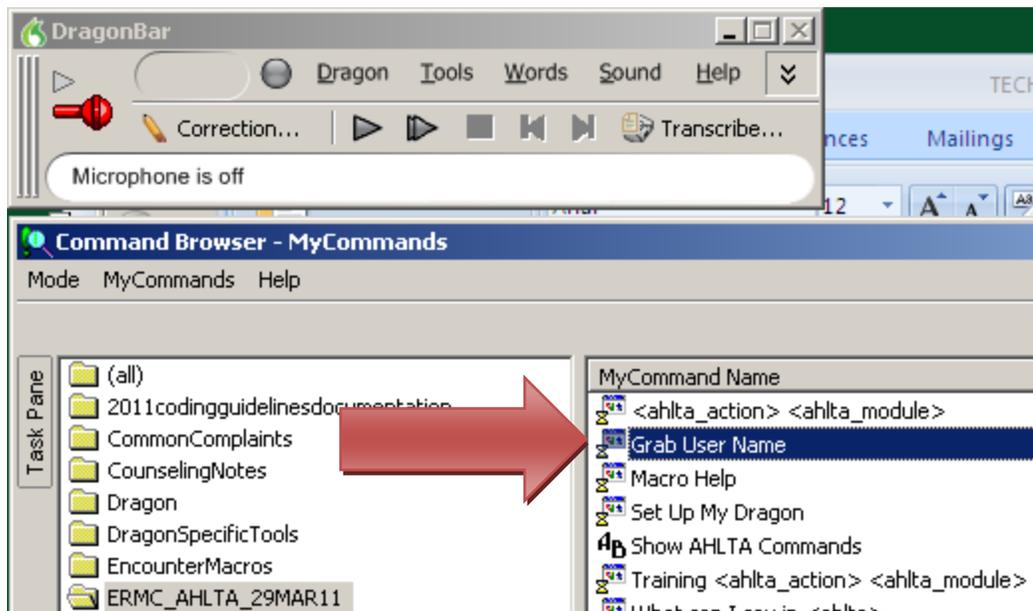**Writes a file to C: drive with commonly used functions**
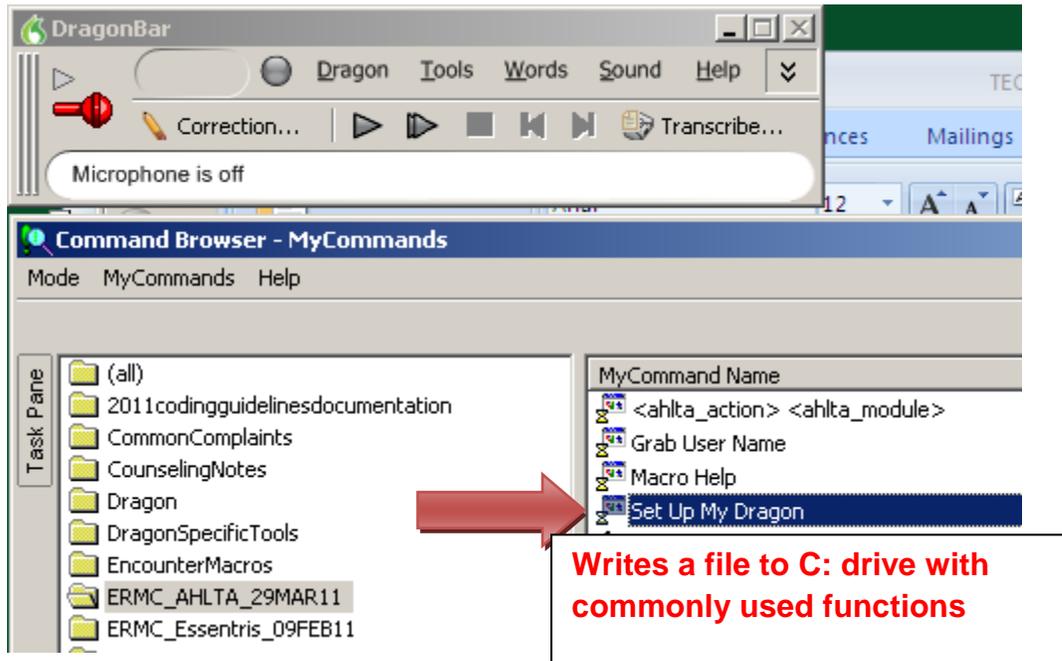
Dragon script using described functions MUST then have this before sub main:
'#Uses "C:\Documents and Settings\All Users\Documents\ERMC_GlobalDragon.VBS"
(Although appears to be a comment, it is not, it references the VBA functions here.)
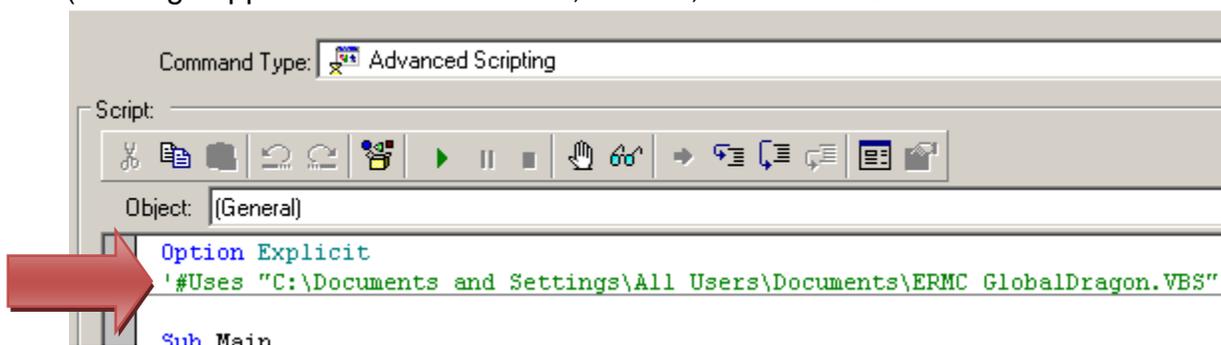
Optional:

Say (or run) "**Grab User Name**"

When:
1) If user uses same computer often (tablets, same exam room with personal log-on/CAC). NOT GENERIC KIOSK.

Instructions:
1) Must be logged into AHLTA (only 1 instance of AHLTA).
2) Run once (after "set up my dragon"), finds AHLTA on desktop and saves AHLTA user name under user's folder on C: drive.
3) fFindAHLTAMainWindow function looks for file before searching for AHLTA. Saves time for AHLTA window search with macros.
4) **Never, ever run this on kiosks with shared/generic log-ons—captured user name will prevent others from using Dragon macros.** Can use on shared computers with CAC/separate log-ins.

Functions

fFindAHLTAMainWindow -- returns true if found
> **1) MUST RUN PRIOR TO CALLING OTHER AHLTA FUNCTIONS**
> 2) If ran "Grab User Name", uses stored AHLTA username.
> 3) Otherwise, searches for "***(Privacy Act of 1974/FOUO)***" on desktop, and if 1 copy found, sets AHLTA user name to this for rest of macro.
> 4) If more than 1 copy of AHLTA open, gets last name of the users' profile (dll)
> > a. Reasons for failure of this ^{3rd} step:
> > > i. CAC person not one logged into AHLTA
> > > ii. Last name change, hyphenated/apostrophes different in user profile than in AHLTA
> > b. Gives error if unable to find AHLTA window with last name on dll

fSendMAPSKeys (**String to send, # of repeats**)
> 1) Time-saving repeating function -- fSendMAPSKeys("{TAB}",15) = SendKeys "{TAB}" 15 times with a 0.1second wait between each
> 2) Why? As a single command ({TAB 15}), not all will be executed by AHLTA or Essentris without short waits between
> 3) Greatly simplifies # of lines of programming

fActivateEssentrisMainWindow -- *returns true if found*
> 1) Activates Essentris using FindWindowsLike (searching for "***- Essentris***")
> 2) Only finds the main/root menus of Essentris, does not find pop-up windows

fAHLTAAddNote (NoteTitle, NoteText, *Optional* InitialWaitTime, *Optional* WaitTimeBetweenTabs) – *returns true if completes*
> 1) Places title and text into Add Note
> 2) Default 1.5 second initial wait after note appears prior to typing note title
> 3) Default 0.4 second wait between (2) tabs prior to putting in note text
> 4) 2&3 can be changed from default if macro too fast

Examples:

Title "Cardiology Consult", pastes contents of clipboard
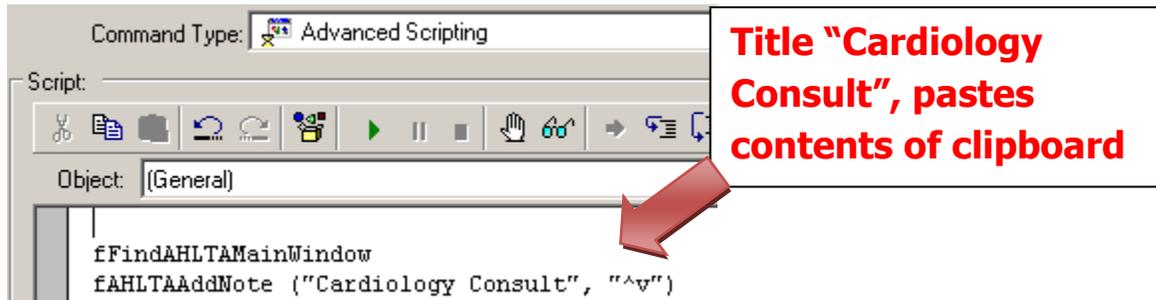
```
fFindAHLTAMainWindow
fAHLTAAddNote ("EKG Procedure", "ECG done in clinic, see tracing in clinical notes")
```

```
Object: [(General)
    fFindAHLTAMainWindow
    Dim sTitle As String
    Dim sText As String
    sTitle = "ECG Procedure"
    sText = "ECG done in clinic,   tracing
    fAHLTAAddNote (sTitle, sTex   .5)
```

fAHLTAClinicalNote (FirstLetterofMTF, #NumberofMTF (within letter), FirstLetterofReport, #OfReportType (within letter), NoteTitle, ConsultText, *Optional* InitialWaitTime) – *returns true if successful*

1) Puts Clinical Note in AHLTA and saves it (does not close clinical notes)
2) Requires first letter of MTF, choice within letter
3) First letter of report type, number within that letter
4) Default 1 second after "clinical notes" appears before entering data
5) Examples:

```
Object: [(General)
    fFindAHLTAMainWindow
    fAHLTAClinicalNote ("m", 2, "o", 3, "Test Report2", "^v",1.5)
```

| Appointments | Current Encounter | **Clinical Notes** |

New/Edit Clinical Note

Date... | 04/04/2011 | Clinician... | EASHRY3720

POC: Martin ACH, Fort Benning, GA

Note Types: Outpatient Report

Operative Report
Outpatient Consultation
Outpatient Report
Outpatient Surgery Report
Patient Notes
Physician Progress Notes
Procedure Report
Text Notes

Irwin ACH Fort Riley, KS
Jalalabad
Keller ACH West Point NY
Kirkuk
Landstuhl RMC
Lyster ACH Ft. Rucker AL
Madigan AMC-Fort Lewis
Martin ACH, Fort Benning, GA

fAHLTADiagnosis (**DiagnosisName, DiagnosisChoice#**, *Optional* WaitTimeBeforeDiagnosis, *Optional* WaitTimeAfterDiagnosis) – *returns true if successful*

1) Navigates to A/P→diagnosis
2) Enters "0" until AHLTA gives an error (this is expected)
3) Then enters first diagnosis in AHLTA
4) Accepts ICD-9 number or name, requires choice #
5) If subsequent diagnoses are needed, can just add short wait and sendkeys "{ICD-9 code}" and alt-a, no need to use this command numerous times
6) Default settings of 1 second before diagnosis and 0.6 seconds after
7) Examples:

("786.50", 1) types "786.50" then chooses first ICD-9 (786.50)

fFindAHLTAMainWindow
fAHLTADiagnosis ("786.50", 1)

CHEST PAIN 786.50

("chest pain", 2) types "chest pain" then chooses second ICD-9

fFindAHLTAMainWindow
fAHLTADiagnosis ("chest pain", 2)

ATYPICAL CHEST PAIN 786.59
CHEST PAIN 786.50
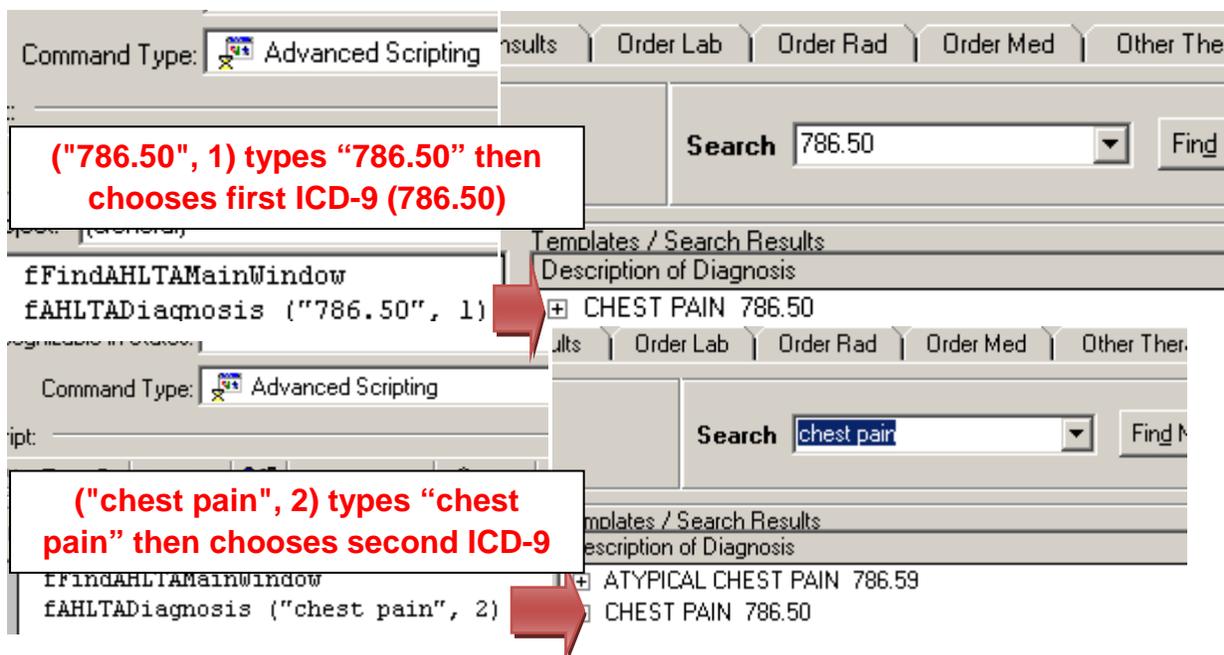
fAHLTAOrderSets
1) Opens order sets (does not do anything within)
2) Many providers like to be left here to allow entry of orders

fAHLTAProcedure (**CPTCode#**) – *returns true if successful*
1) Navigates to A/P→procedure
2) Enters "0" until AHLTA gives an error (this is expected)
3) Enters procedure CPT code (assumes this will be first code in AHLTA)
4) Example: fAHLTAProcedure (93000)

fOpenAHLTAModifers -- *returns true if successful*
1) Opens up procedure modifiers window
   a. Can enter units of service, modifiers

2) Although are many options, be careful programming too much as choices depend on the permissions of the person logged in.

fCloseAHLTAModifiers
    1) Closes the window (basically just hits alt-o)

fAHLTADisposition (**FirstLetterofCategory, #OfCategoryChoice, #ofEMCodeWithinCategory**, *Optional* WaitAdminOptionsReady, *Optional* sngWaitAfterAdminOptions) – *returns true if successful*
    1) Finds AHLTA, opens disposition
    2) Keeps hitting "%a" until "administrative options" window appears, then cancels this window (now computer is ready)
    3) Moves down to "calculated" or "select" or "modifiers" tabs (can be any), then moves left 2 (ensuring is at "calculated", then right 1, now at "selected" tab.
    4) Enters E/M Code by typing "<" which brings up "<none>" (resets the E/M).
    5) Types first letter of the category (i.e. "9" for 99499, "o" for office/outpatient visit). **NOTE: typing "o" after "<none>" may bring up a different choice than typing "o" from the default type of encounter—make sure you start at "<none>" when determining counts, as the program starts with this.**
    6) Inputs the number of choice within letter ("o", 2 would be outpatient consults)
    7) Then moves down to the # of the choice within this category
    8) Examples:

Object: (General)

```
fFindAHLTAMainWindow
fAHLTADisposition ("9", 1, 5)
```
Codes 99499

```
fFindAHLTAMainWindow
fAHLTADisposition ("o", 1, 4)
```
Est Level 4

Object: (General)

```
fFindAHLTAMainWindow
fAHLTADisposition ("o", 1, 9)
```
New Level 4

fAHLTAOpenSignatureWindowfromAPorDispo – *returns true if successful*
  1) Opens up sign encounter screen from AP or Dispo

fAHLTAOpenSignatureWindowfromSF600
  1) Opens up sign encounter screen from the SF600

General Functions

fCheckForWindow (TargetWindow, WaitMax (1sec),
              Show/HideMsgBox) – *returns true if finds window*
  1) Waits for a window to gain the focus on the desktop
  2) List # of seconds to wait for the window (cycles every second)
  3) You control if a message box is made if the window fails to appear.
  4) Typically used to wait for a window after giving a command.
  5) Example (see next page):

Command Type: 🖳 Advanced Scripting

Script:

[toolbar]

Object: (General)

```
Option Explicit
'#Uses "C:\Documents and Settings\All Users\Documents\ERMC GlobalDragon.

Sub Main

    If fActivateEssentrisMainWindow Then                        'Find Es
        Wait .1
        SendKeys "%n",1                                          'Select
        SendKeys "{ENTER}",1
    Else
    GoTo jumpout
    End If

    If fCheckForWindow ("Notes Menu Screen - Essentris",5,True) Then
        SendKeys "^n",1                                          'Wait fo
    Else
        GoTo jumpout
    End If

    If fCheckForWindow("Select A Note Type",5,True) Then
        SendKeys "consolidated consult",1
        SendKeys "{ENTER}",1                                     'Wait fo
    Else
        GoTo jumpout
    End If

    If fCheckForWindow ("Date/Time Selection",5,True) Then
        SendKeys "{ENTER}",1                                     'Wait an
    Else
        GoTo jumpout
    End If

    If fCheckForWindow ("Enter or select a new note topic",5,True) Then
        SendKeys "Cardiology Consult",1                          'Waits a
        SendKeys "{ENTER}",1
    Else
        GoTo jumpout
    End If

    If fCheckForWindow ("Consolidated Consult",5,True) Then
        SendKeys "Medicine",1                                    'Waits f
        fSendMAPSKeys ("{ENTER}",3)
```

Window Waiting for

Tries 5 Seconds

Message Box if Fails

fActivateWindow (TargetWindow, intWaitMax (0.1sec), Show/HideMsgBox) – *returns true if finds*

1) Somewhat similar to fCheckForWindow, but tries to activate a window on the desktop rather than waiting for a new one to appear
2) Use when window is already open on the desktop
3) Cycles every 0.1 seconds, (50=5 seconds)
4) Control whether you want message box to appear if fails
5) Example:

Command Type: Advanced Scripting

Script:

Object: [General]

Tries to activate Inbox for 5 seconds, shows message box if fails to find

```
fActivateWindow ("Inbox - Microsoft Outlook", 50, True)
```
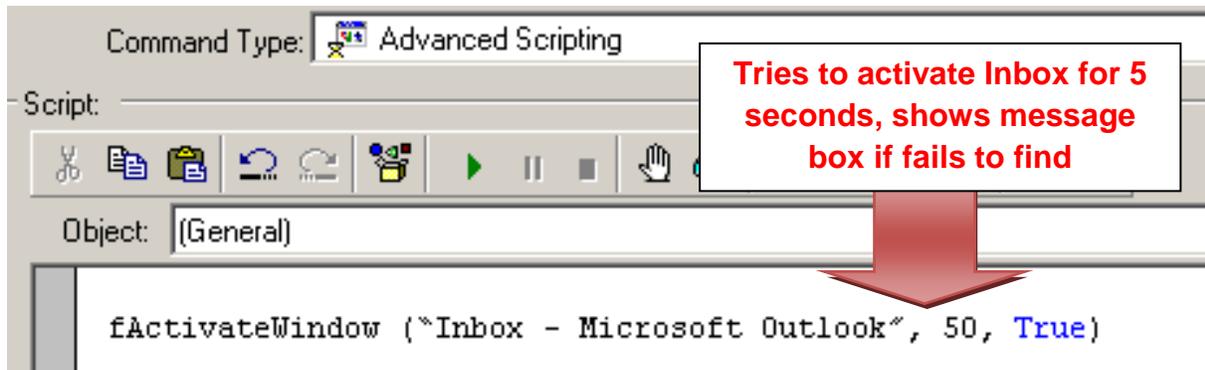
fSearchForWindow (TextToFind) – *returns value 0 if not found, 1 if found 1 window with title, 2 if more than 1 window with title*

1) Finds a window on the desktop
2) Uses "*" as wildcard
3) If only 1 copy that meets the criteria, brings this window to focus
4) Example:

Command Type: Advanced Scripting

Script:

Finds MS Word window, regardless of file name

Object: [General]

Does not work if 2 instances with same search string are open -- ("Sample Note - Microsoft Word") & ("Sample Note 2 - Microsoft Word")

```
fSearchForWindow ("* - Microsoft Word*")
```
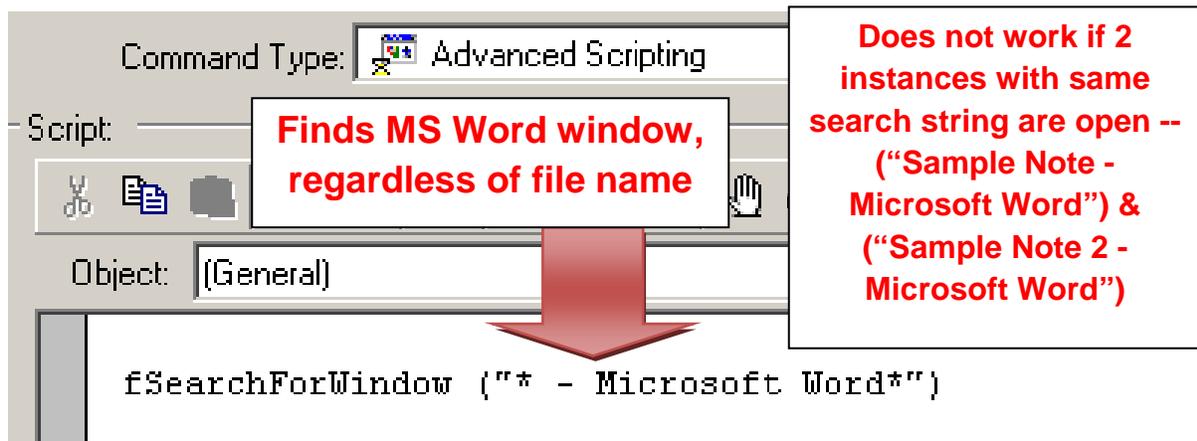
fUpdateTracer (MacroName)

1) Simple counter function, creates a file in "c:/Documents And Settings/" + sFullname + "/" (fullname from GetUserName, that is, folder of user at time).
2) Saves a file named "MAPS Macro Tracer_" + sMacroName + # times used (updated every time used with prior file deleted) .EXE
3) .EXE is critical--although there is nothing in the file other than text saying it is harmless, the DoD has robust tracking programs for .EXE files. This allows us to follow, across the Army, use of every one of these macros for every user, and how many times used.
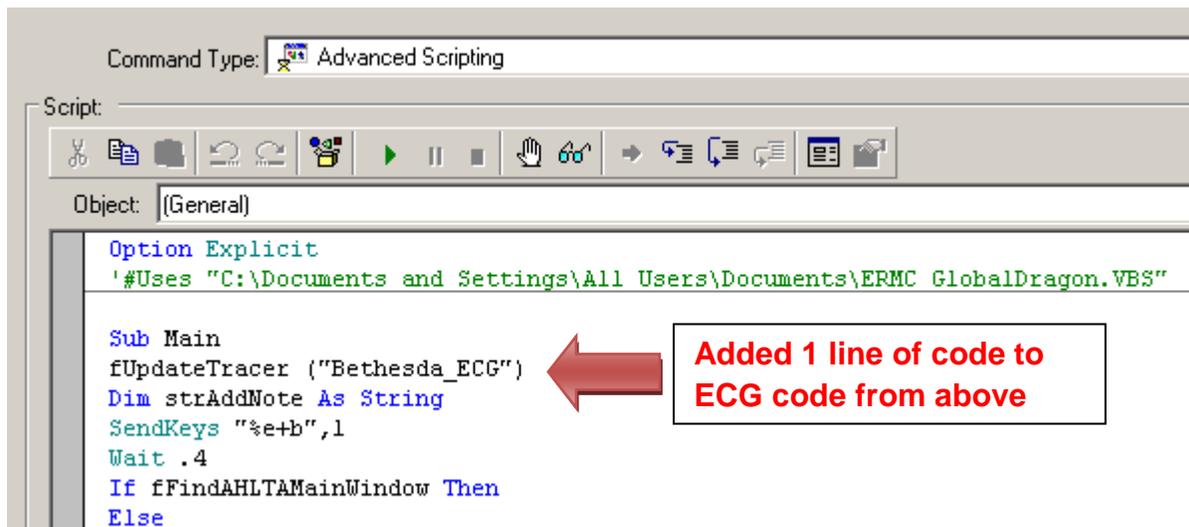
Command Type: Advanced Scripting

Script:

Object: (General)

```
Option Explicit
'#Uses "C:\Documents and Settings\All Users\Documents\ERMC GlobalDragon.VBS"

Sub Main
fUpdateTracer ("Bethesda_ECG")
Dim strAddNote As String
SendKeys "%e+b",1
Wait .4
If fFindAHLTAMainWindow Then
Else
```

Added 1 line of code to ECG code from above

C:\Documents and Settings\Eric.Shry

File   Edit   View   Favorites   Tools   Help

Back   Search   Folders

Address   C:\Documents and Settings\Eric.Shry

Name	Size	Type
NTUSER.DAT	4,864 KB	DAT File
-fanixtemp	47 KB	File
NTUSER.DAT.LOG	1 KB	Text Document
MAPS Macro Tracer.exe	1 KB	Application
MAPS Macro Tracer_Essentris_Basement(v2.0)_00012...	1 KB	Application
MAPS Macro Tracer_Bethesda_ECG_00001.exe	1 KB	Application
MAPS Macro Tracer_AHLTA_Basement(v2.0)_00019.exe	1 KB	Application

Tracer (Bethesda_ECG) shows up under the user's folder and use can be tracked. Note other tracers.

Complete Encounter Programming Examples

To start in acrobat reader, find AHLTA, paste in tracing in clinical notes, put a generic statement into add note, put in chest pain as diagnosis, code ECG, and add 99499:
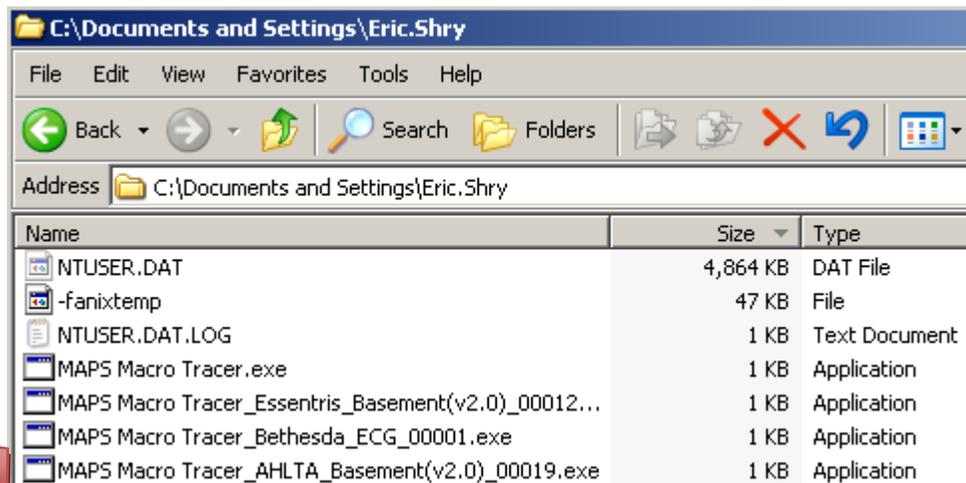
```
Command Type:  Advanced Scripting

ject: (General)

Option Explicit
'#Uses "C:\Documents and Settings\All Users\Documents\ERMC GlobalDragon.VBS"

Sub Main
    Dim strAddNote As String
    SendKeys "%e+b",1                    'Pastes file to clipboard in PDF reader
    Wait .4
    fFindAHLTAMainWindow
    fAHLTAClinicalNote ("n", 20, "p", 2, "ECG Image", "^v")      'Bethesda
    strAddNote = "ECG done in clinic, see tracing in clinical notes"
    Clipboard strAddNote
    fAHLTAAddNote ("Cardiology Consult", "^v")       'Pastes clipboard (faster than typing
    fAHLTADiagnosis ("chest pain", 2)                'Chooses second chest pain ICD-9
    fAHLTAProcedure (93000)                          'code for ECG
    fAHLTADisposition ("9", 1, 5)                    '99499
    fAHLTAOpenSignatureWindowfromAPorDispo
End Sub
```

This is great as long as there are never "hiccups", however, if the first step breaks (tier 1 or the user hits the mouse), it will try to execute 50 more steps.

Therefore, isolating each step in an *if …then…else jumpout* loop is very helpful.

Note that all the functions are Boolean (true or false). If false, then end the macro.

So the above code becomes (next page):

Command Type: ⚙ Advanced Scripting

Script:

✂ 🖹 🖹 ↺ ↻ 🖳 ▶ ▌▌ ■ ✋ 👓 ➡ ⮡ ⮢ ⮤ 🔳 🖳

Object: (General)

```vbscript
Option Explicit
'#Uses "C:\Documents and Settings\All Users\Documents\ERMC GlobalDragon.VBS"

Sub Main
Dim strAddNote As String
SendKeys "%e+b",1
Wait .4
If fFindAHLTAMainWindow Then
Else
      GoTo jumpout
End If
If fAHLTAClinicalNote ("n", 20, "p", 2, "ECG Image", "^v") Then
Else
      GoTo jumpout
End If
strAddNote = "ECG done in clinic, see tracing in clinical notes"
Clipboard strAddNote
If fAHLTAAddNote ("Cardiology Consult", "^v") Then
Else
      GoTo Jumpout
End If
If fAHLTADiagnosis ("chest pain", 2) Then
Else
      GoTo jumpout
End If
If fAHLTAProcedure (93000) Then
Else
      GoTo jumpout
End If
fAHLTADisposition ("9", 1, 5) '99499
fAHLTAOpenSignatureWindowfromAPorDispo
jumpout:
End Sub
```

Can add waits between functions (at arrows) if needed depending on local speed. Also, add note and clinical note have default waits built in that can be increased.

MAPS Functions Improvement/Sustainment

1) Currently, for Dragon Medical users, requires them to "Set Up My Dragon" each time there is an upgrade
2) "Set Up My Dragon" VBA reference could be placed on share drive for those who use them only at work and have constant access, but requires reference at beginning of each macro to be changed to share drive.
3) HTML or VB-integration is better for big, highly used macros
   a. Dragon has large memory overhead and significantly slows down macros.
   b. Dragon script is less stable, lock-ups in VB can be broken with function break, but lock-ups in Dragon require task manager "end process".
   c. Obviously easier to maintain a HTML page that references an executable for common macros.
   d. Talk with your IMD
4) However, for individual providers who dictate and then paste in notes with their signature blocks, specialized content, etc. continued Dragon Medical Scripting will be important over next several years.