Speed Learning

2015 Survox Summit

Speed Learning – Time Savers, Quick & Useful Tips on how to get this done fast and efficiently

What is Speed Learning?

- Speed Learning is a collection of methods of learning which attempts to attain higher rates of learning without unacceptable reduction of comprehension or retention.
- Speed Learning is closely related to speed reading, but encompasses other methods of learning, such as observation, listening, conversation, questioning and reflection.

Speed Learning Rules!

- Each presenter/group will have 5 minutes to present
- Then each presenter will introduce themselves
- There will be a brief transition between each presenter/group
- There will be a few minutes at the end for Q&A with all presenters
- Overall goal is that everyone at least learns something new

Speed Learning Presenters

Presenters	Company	Topic
Pat Costello	LHK Partners Inc.	**Automating Processes**
Frank Barney, Cora Coleman & Taylor Zigo	Venture Data	**Project Setup Project Management Data Delivery**
Jason Bentrup	MaritzCX	**Jquery / JSON To Populate Data**
Jamie Jurgaitis	Opinion America Group	**Cronjob / Crontab For Customers**
Don Ludemann	Thoroughbred Research	**File Comparison**
Mike Blair	MaritzCX	**Recovering Overwritten Data**

LHK PARTNERS INC.

Marketing Research Worldwide

Pat Costello

LHK Partners Inc.

Automating processes using Survox, PowerShell and Task Scheduler

mentorRun_ps1.txt

```
# PowerShell example of running a mentor program

$program = "C:\CFMC\GO\mentor.exe"
$arguments = "logit.spx -error"

start-process $program $arguments
```

mailnotify_ps1.txt

```
# Example of sending an email with attachement if an error found in output file and another message if no errors.

# Can be modified to if file exists, if no error proceed, or for more advanced error catching could use try/catch logic.


#email information

$EmailTo = "Send@email.com"

$EmailFrom = "From@email.net"

$SmtpServer = "relay.name.net"


#error file and the error key (text in file to determine error)

$ErrorFile = "C:\MyScripts\Error.prt"

$ErrorText = "errors!"


#if fail

$Subject1 = "Hey, your imporant job failed."

$Body1 = "Your stuff failed real bad."

#if complete

$Subject2 = "Step1 is complete"

$Body2 = "Good job, step 1 is done."

$fail = Get-Content $ErrorFile | Select-String $ErrorText -quiet -casesensitive


if ($fail -eq "True")

{

Send-MailMessage `

-To $EmailTo `

-From $EmailFrom `

-Body $Body1 `

-Subject $Subject1 `

-Attachments $ErrorFile `

-smtpServer $SmtpServer

}

else

{

Send-MailMessage `

-To $EmailTo `

-From $EmailFrom `

-Body $Body2 `

-Subject $Subject2 `

#-Attachments $ErrorFile `   # commented out to not send an attachment

-smtpServer $SmtpServer

}
```

AddWorksheetHTM_ps1.txt

```
# Power Shell example of importing a .htm file into a new Excel worksheet

# The new worksheet is given the run date as the worksheet name

$ExcelFile = ìC:\example\Book1.xlsxî

$WEBFile = "C:\example\report.htm"

$sheetname = Get-Date -format M.d.yyyy


#create Excel object

$Excel = New-Object -ComObject Excel.Application

$Excel.visible = $false       # change to $true for debugging else $false

$Excel.DisplayAlerts = $false


#Open workbook

$wb = $excel.Workbooks.Open($ExcelFile)

# add new worksheet

$worksheet = $wb.WorkSheets.Add()

$worksheet.Name = "$sheetname"

#Define the connection string and where the data is supposed to go

$TxtConnector = ("URL;" + $WEBfile)

$CellRef = $worksheet.Range("A1")


#Build, use and remove the text file connector

$Connector = $worksheet.QueryTables.add($TxtConnector,$CellRef)

$worksheet.QueryTables.item($Connector.name).Refresh()

$worksheet.QueryTables.item($Connector.name).delete()


$worksheet.UsedRange.EntireColumn.AutoFit()


#Save Excel file

#51 = xlOpenXMLWorkbook (without macro's in 2007-2013, xlsx)

#52 = xlOpenXMLWorkbookMacroEnabled (with or without macro's in 2007-2013, xlsm)

$worksheet.SaveAs($ExcelFile,51)

$Excel.Quit()


#Make sure ComObjects close

[System.Runtime.Interopservices.Marshal]::ReleaseComObject($Excel)
```

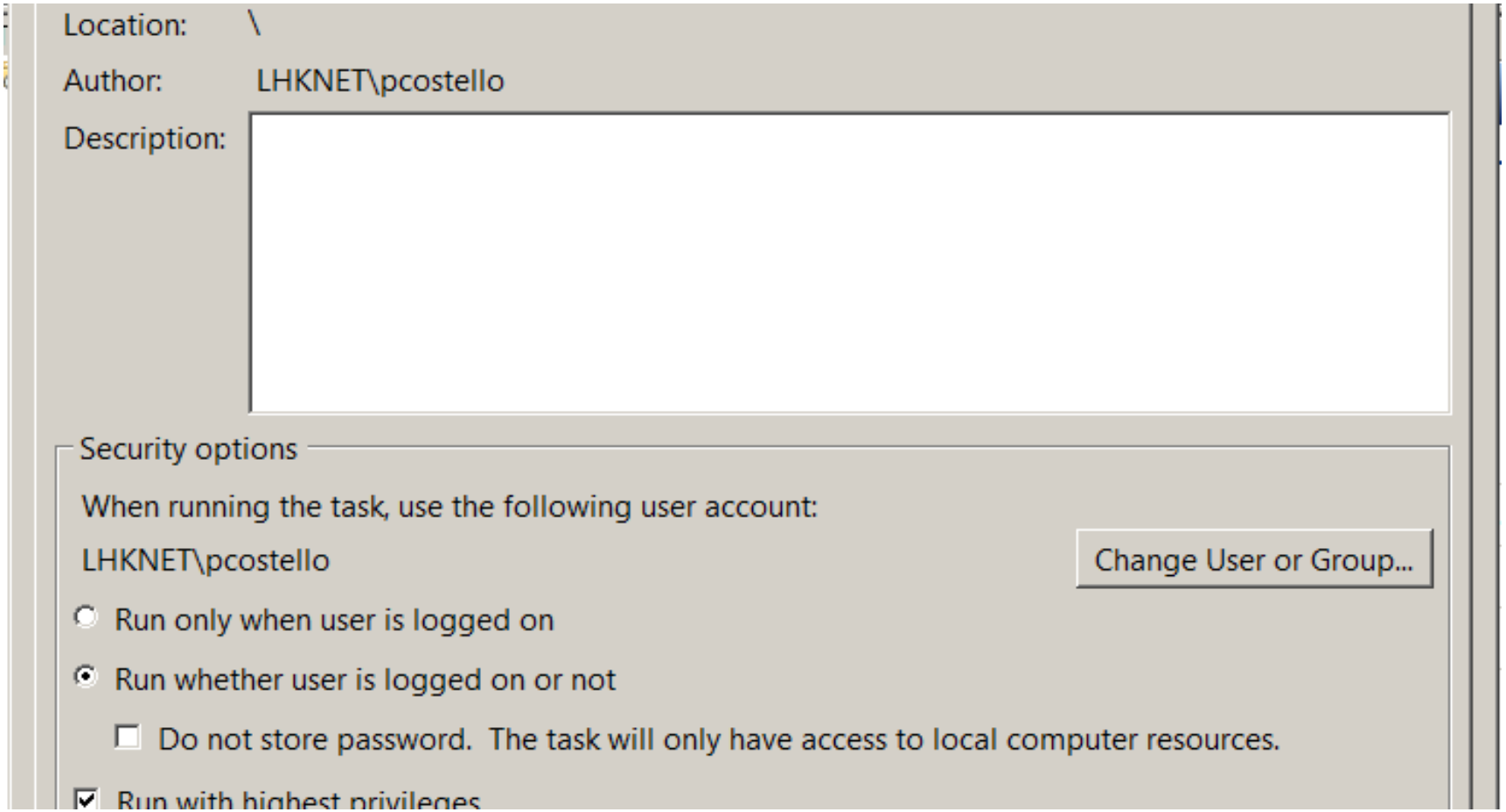
ApendToWorksheet_ps1.txt

```
# PowerShell example of appending a tab delimited file to the end of an Excel worksheet

#Input files and sheet name
$ExcelFile = "C:\example\W73report_request.xls"
$SheetName = "w73report_request"
$TextFile = "C:\example\testfile.txt”

#Open Excel object
$objExcel = New-Object -ComObject Excel.Application
$objExcel.Visible = $false     # change to $true for debugging else $false
#Open workbook
$objWorkbook = $objExcel.Workbooks.Open($ExcelFile)
$objWorksheet = $objWorkbook.Worksheets.item($SheetName)
$objWorksheet.Activate()

#determine last row
$LastRow = $objWorksheet.UsedRange.Rows.Count + 1
$LastRow
$objExcel.Range("A" + $LastRow).Activate()

#Define the connection string and where the data is supposed to go
$TxtConnector = ("TEXT;" + $Textfile)
$CellRef = $objWorksheet.Range("A" + $LastRow)

#Build, use and remove the text file connector
$Connector = $objWorksheet.QueryTables.add($TxtConnector,$CellRef)
$objWorksheet.QueryTables.item($Connector.name).Refresh()
$objWorksheet.QueryTables.item($Connector.name).delete()

$objWorksheet.UsedRange.EntireColumn.AutoFit()

#Save and close Excel
$objWorkbook.Save()
$objWorkbook.Close()
#Closing COM object 3 ways, It has a habit of the process hanging otherwise
$objExcel.Quit()
[System.Runtime.Interopservices.Marshal]::ReleaseComObject($objExcel)
Stop-Process -Name EXCEL -Force
```

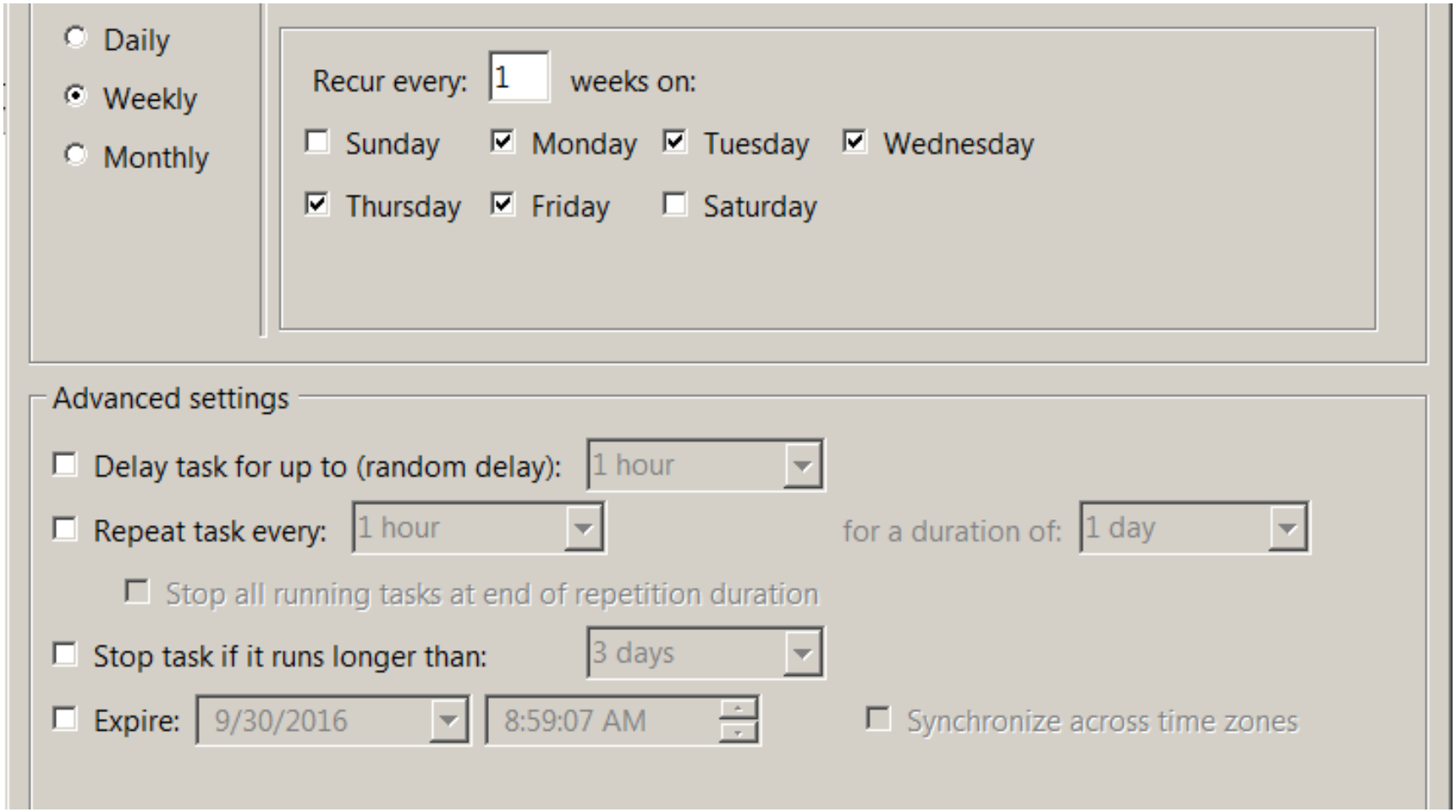
RunProg_bat.txt

```
REM Bat file to run a PowerShell program. Advance settings allows runningwith
Windows task monitor when the user is not logged onto the machine.


@ECHO OFF

PowerShell -NoProfile -ExecutionPolicy Bypass -Command "& {Start-Process
PowerShell -ArgumentList '-NoProfile -ExecutionPolicy Bypass -File
""C:\File\directory\on\the\PC\RunProg.ps1""' -Verb RunAs}"
```

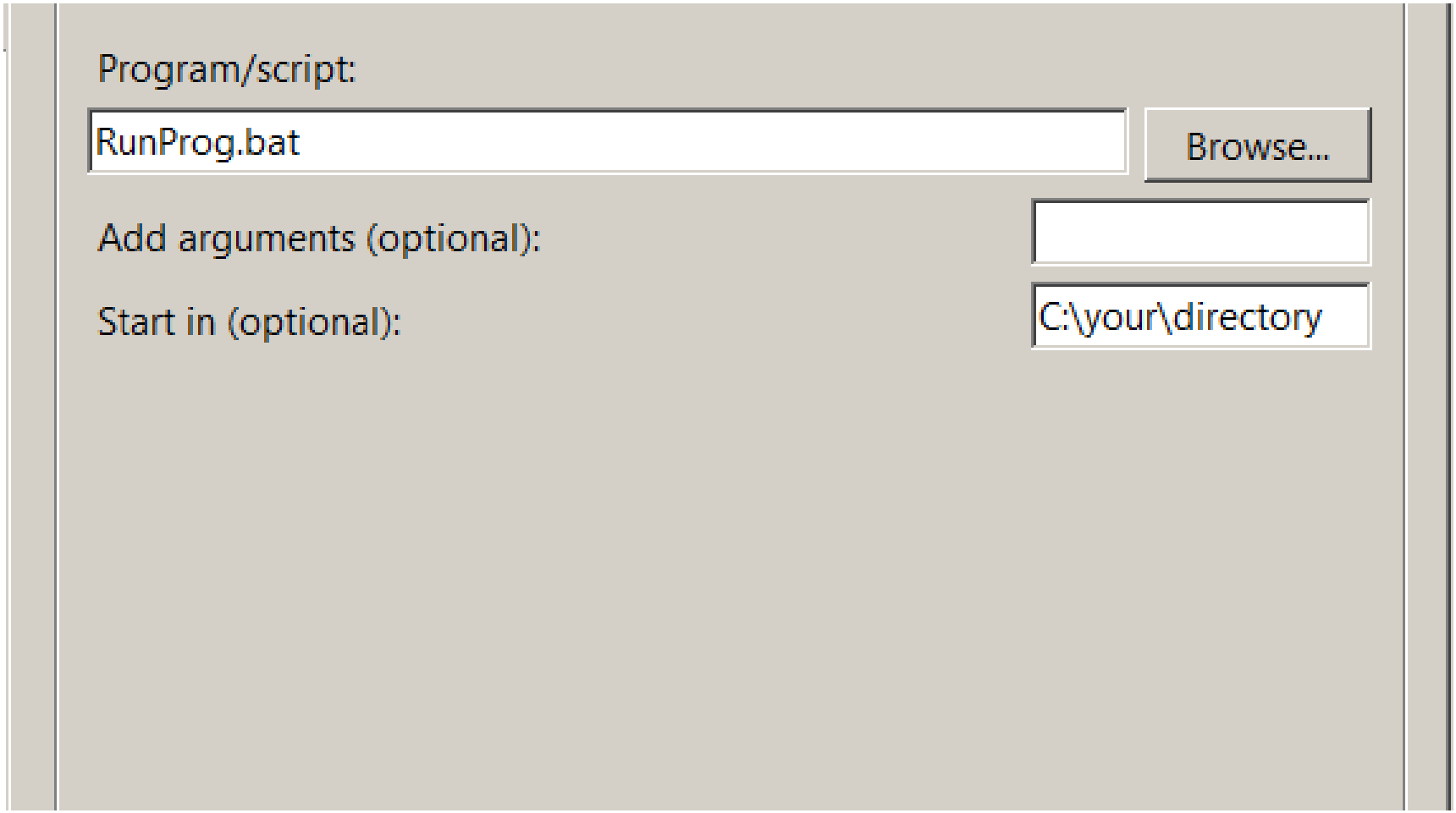
Task schedule 1



Task schedule 2



Task schedule 3



Frank Barney, Cora Coleman & Taylor Zigo
Venture Data

Project Setup, Project Management & Data Delivery

Jason Bentrup
MaritzCX

Using JSON and jQuery instead of a DBR

Benefits

- Does not require a recompile
- Can be hot swapped
- UTF-8 compliant
- Fully customizable
- No size limit

JSON

```
var data = [
    {
     "listid": "it_0000272",
     "dealers": [
        {"name": “ROMA AUTOS, 1 VIA ROMA,", "code": "00005902"}
      ]},
    {
     "listid": "it_0026718",
     "dealers": [
        {"name": "EUROCAR, VIA NAPOLI", "code": "00003710"},
        {"name": "EUROCAR, PIAZZA ROMANI", "code": "00003711"},
        {"name": "EUROCAR, VIA LEONARDO DAVINCI,", "code":
"00003712"}
    ]}
]
```

Survent

{D_1_A:
<script type=“TEXT/JAVASCRIPT” SRC=“json.js”>
<script type=“TEXT/JAVASCRIPT” SRC=“load_dealer.js”>

<input type=“hidden” name=“regioncode” value=“0026718”>

<select name=“dealer”>
<option>Select</option>
</select>

!disp
}

{Q_1: 5001.8
!VAR,H}

jQuery

```
(function($) {
   $.fn.changeType = function() {
       //bring in json file and format it for select box
   }
}
$(document).ready(function() {
   var inputbox = $("input[type='hidden'][name^='var_']");
   $("form#CFMC").changeType();
   $("select#dealer").change(function() {
       var selected = $("#dealer option:selected").text();
       var selectedID = $("#dealer option:selected").val();
       if (selected != "Select") {
           $(inputbox).val(selectedID);
       }
   });
})
```

Jamie Jurgaitis
Opinion America Group

Take some the mundane tasks out of your day

Take some the mundane tasks out of your day.

- Using your linux Cron
- Edit crontab -e
- Set up to run at any time during the day, week, year
  - Minute
  - Hours
  - Day
  - Month
  - Weekday
- Moving files or email files when you need them
- Maintain your server by archiving log files
- Running reports and placing them into the shared files area for supervisors and clients

#Daily Callbacks

30 05 * * 1-5 csh -c "setenv CFMC /cfmc/;/cfmc/websurv/studies/reports/scripts/jobs/daycbam.csh >& /cfmc/websurv/studies/reports/scripts/jobs/daycbam.log"

Don Ludemann
Thoroughbred Research

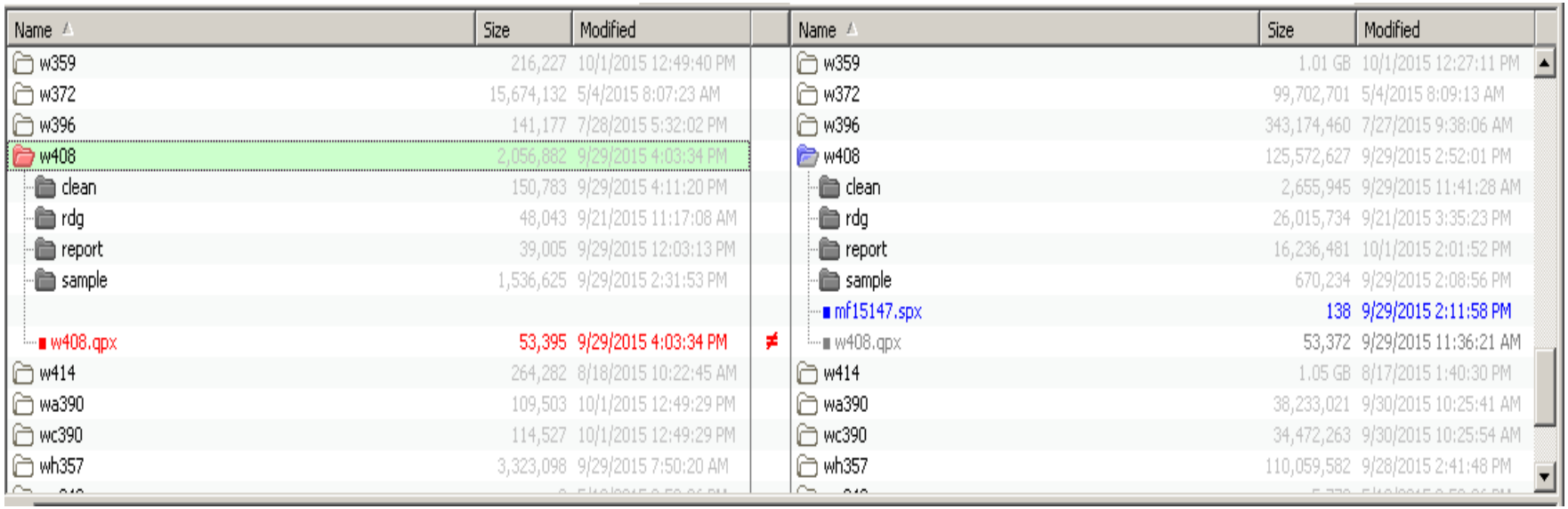
File comparison software to figure out what changes were made in a program

I have it working.
Now what did I do to get here?

- Ever lost track of code changes?
  - Programmers have to try lots of things to make a
  - File comparison can save your bacon!
- File comparison software
  - Find the files that changed
  - Look at the changed lines of code

Comparison software

- Folder

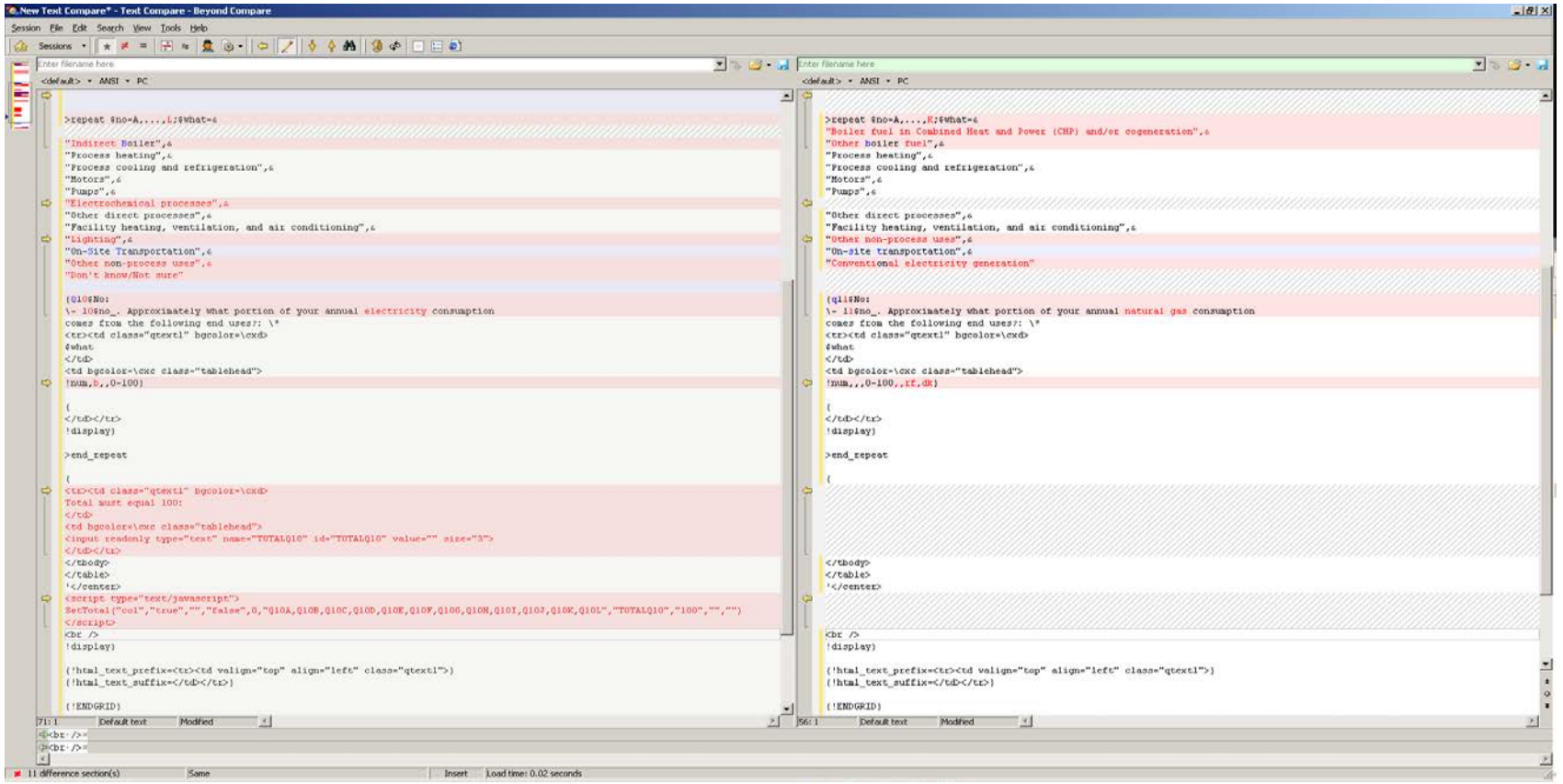


Name	Size	Modified
w359	216,227	10/1/2015 12:49:40 PM
w372	15,674,132	5/4/2015 8:07:23 AM
w396	141,177	7/28/2015 5:32:02 PM
w408	2,056,882	9/29/2015 4:03:34 PM
clean	150,783	9/29/2015 4:11:20 PM
rdg	48,043	9/21/2015 11:17:08 AM
report	39,005	9/29/2015 12:03:13 PM
sample	1,536,625	9/29/2015 2:31:53 PM
w408.qpx	53,395	9/29/2015 4:03:34 PM
w414	264,282	8/18/2015 10:22:45 AM
wa390	109,503	10/1/2015 12:49:29 PM
wc390	114,527	10/1/2015 12:49:29 PM
wh357	3,323,098	9/29/2015 7:50:20 AM

Name	Size	Modified
w359	1.01 GB	10/1/2015 12:27:11 PM
w372	99,702,701	5/4/2015 8:09:13 AM
w396	343,174,460	7/27/2015 9:38:06 AM
w408	125,572,627	9/29/2015 2:52:01 PM
clean	2,655,945	9/29/2015 11:41:28 AM
rdg	26,015,734	9/21/2015 3:35:23 PM
report	16,236,481	10/1/2015 2:01:52 PM
sample	670,234	9/29/2015 2:08:56 PM
mf15147.spx	138	9/29/2015 2:11:58 PM
w408.qpx	53,372	9/29/2015 11:36:21 AM
w414	1.05 GB	8/17/2015 1:40:30 PM
wa390	38,233,021	9/30/2015 10:25:41 AM
wc390	34,472,263	9/30/2015 10:25:54 AM
wh357	110,059,582	9/28/2015 2:41:48 PM

Comparison software

- File



Next Steps

- Beyond Compare
  - Win Mac Linux
  - Folder compare
  - File compare
  - ftp/sftp
  - 3 way compare
  - $30/$50 per seat
- Others
  - TFS
  - Google “file comparison tool” (some are free)

Mike Blair
MaritzCX

Recovering Overwritten Data From Questions in a WebSurvent project

Recovering Overwritten Data From Questions in a WebSurvent project

Htmlresp log file info:

File name structure:

<studycode>.htmlresp/<studycode>_<password or ivid>.htmlresp

(uses password if websurvent and Interviewer ID if webcati)

Eg: /cfmc/intvr_logs/abcde.htmlresp/abcde_ercqsm4wnmu84.htmlresp

Survent concatenates to the file if it is there when it starts up, so you don't have to concatenate files. The most recent response is always after any previous responses in the file in the case of backups and suspends.

Htmlresp record formats:

1. Question Response:

mm:dd hh:mm:ss Label:Data

eg. answer to S1B:, label starts in column 16 of record

```
09:08 09:46:11 S1B:01,10,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,
```

2. Data modification:

mm:dd hh:mm:ss NO QQ:mod data Location data

The data location always starts in column 31, eg. Case id (last line in file):

```
09:08 11:08:47 NO QQ:mod data 1.10 1530000762
```

Recovering Overwritten Data From Questions in a WebSurvent project (cont)

Process:

1. Gather the files you need to a directory. Use the ~input files=" statement to read them all, and set up a memory area to save data to, here is the flow reading through the file:

```
rec1: |1.10000 data buffer              |1000jkkkkj1.200 memory buffer      |
recn: |1.10000 data buffer cleared      |10001.200 memory buffer not cleared|
last: |1.10000 data buffer cleared      |10001.200 memory ready to retrieve |
```

2. Use the '~merge' command set to replace the missing data

SurVox wishes:

1. Put the filename at the top of any new file so can know when the file starts, and label the caseid record
2. Be able to know when you are at the beginning or end of each file in a "files=" statement, eg.

```
If FirstCaseInCurrentFile
ExeofCurrentfile
If LastcaseInCurrentFile
```

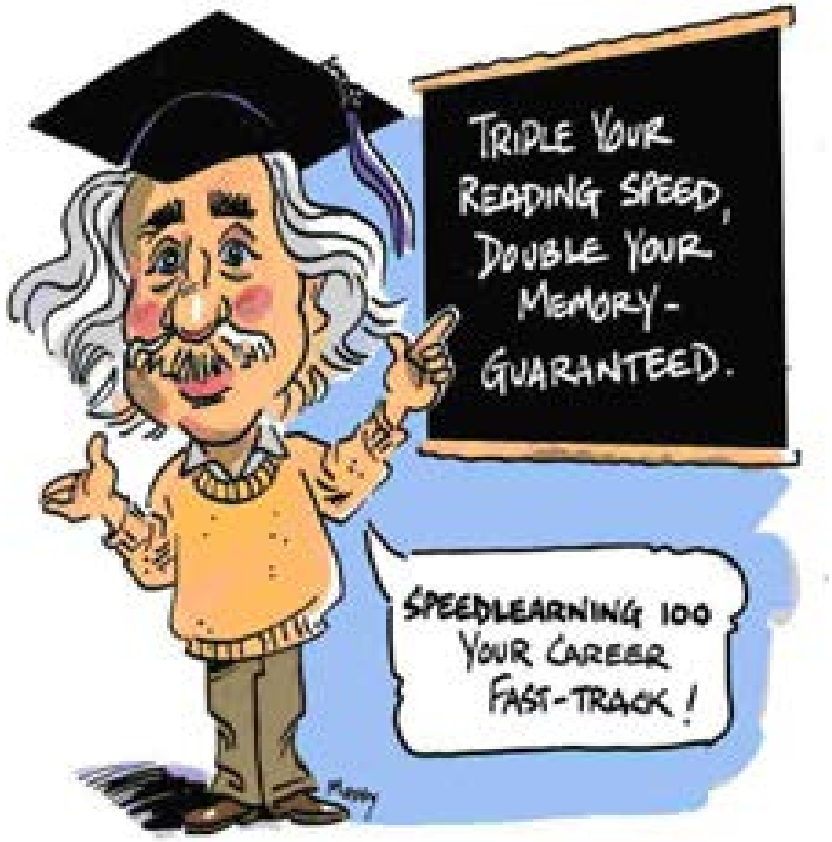
3. An option in ~merge to write the records not written in the first file to a second file

GetHtmlRespData.spx

```
~comment
"Memory area:
10001.10 caseid
10012.3 41.3=001
10021.20 S1B
>define @LABEL S1B
>define @LABELLENGTH 3
>define @STUDY mystudy
~define proc=p1:
"S1B
If [16.4#"S1B:"]
  modify [10021.30$]=substitute([20.3$],",","")
Endif
If [22.14#"mod data 41.3 "]
  copy status[10012.3]=[36.3]
Endif
"If we found a Caseid
If [22.13#"mod data 1.10 "]
  If [10012.3#001] "if it was a complete
    modify [10001.10$]=[36.10$]
    writecase [10001.100]
    blank [10001.100]
  Else blank [10001.100] "was a terminate, remove data
Endif
}
```

```
~in files="*/*.htmlresp" ascii length=10000 total_length=10100
~out @STUDY~_htmlresp_merge.txt ascii length=100
~execute proc=p1
~in;
"Append data to study data file
>purgesame
>usedb ../study
~in mystudy.tr study=study1 numbuffers=2
~in @STUDY~htmlresp_merge.txt ascii,length=100, study=study2,newbuf
~out study_merged.tr
~merge
primary=study1
primary_key=[1.10$]
secondary=study2
secondary_key=[1.10$]
sort_primary
sort_secondary
disallowed_duplicates=ok
write_disallowed_duplicates
write_matched_primary
write_unmatched_primary

mcopy [S1B]=[21.50]
~end
```

Speed Learning



Questions?

WWW.SURVOXINC.COM