

# Login VSI 4.0 |

Workload Language Reference Guide

## Disclosure and Warranty

The information, concepts, and ideas contained in this document are the property of Login VSI. Any product descriptions or representations in this document are for identification purposes only and are not to be construed as a warranty of specific properties or guarantee or warranty of any other type.

Login VSI shall assume no liability, either explicit or implied, for the documentation. Information in this document, including URL and other Internet Web site references, is subject to change without notice.

All sample code described in this document is provided by Login VSI for illustrative purposes only. These examples have not been thoroughly tested under all conditions. Login VSI, therefore, cannot guarantee or imply reliability, serviceability, or functionality of these programs or code examples. All brand names and product names used in this document are trademarks of their respective holders and are recognized as such.

© 2013 Login VSI. All rights reserved.

## Contents

1	Introduction.....	3
2	Philosophy .....	4
3	Language options.....	5
4	Predefined variables .....	6
5	Adding an application to a workload.....	8
6	Removing an application from the workload .....	9
7	Functions .....	10
	App_Start.....	10
	App_Close.....	11
	App_Focus.....	12
	PDF_Print .....	13
	Segment.....	14
	VSI_Buffer .....	15
	Set_DefaultPrinter .....	16
	Web_Start.....	17
	VSI_Browse.....	18
	Web_Quit.....	19
	Web_GoTo.....	20
	VSI_Type_fixed.....	21
	VSI_Type_Time.....	22
	VSI_File_Copy.....	23
	VSI_Random_File_Copy.....	24
	VSI_File_Delete .....	25
	VSI_Reg_Write .....	26
	VSI_RegDelete .....	27
	VSI_Sleep .....	28
	Workload_Idle.....	29
	Workload_RandomIdle.....	30
	VSI_ShellExecute.....	31
	VSI_ShellExecuteWait .....	32
	VSI_FileWriteToLine.....	33
	VSI_Timer.....	34

# 1 Introduction

This document describes the Login VSI Meta language for benchmarking 1.0 used by Login VSI 4.0. This language has been designed keeping in mind best practices to do performance tests and allows for easy customization of existing workloads as well as creating new workloads from scratch. All Login VSI Pro licenses can customize and build workloads. The Login VSI Express version of Login VSI does not support customization of the workloads.

## 2 Philosophy

The default workloads shipped with Login VSI 4.0 are designed to stress and performance test a system in a realistic way not only during the project phase of a VDI or SBC implementation but also to safeguard and improve the performance in an existing environment. To do this a few design choices have been made, it's not mandatory to obey these design decisions but it is strongly recommended when building workloads from scratch or extending/changing a large part of them.

- A Workload should take 48 minutes
  - A 48 minute workload should be divided in 4 segments
  - Every segment ends with 2 minutes idle buffer
    - As the workload slows down because of degradation of performance the buffer will be smaller to keep the segments 12 minutes in time
  - Segments reflect user activities, for example the medium workload consists of a segment that does:
    - Multimedia & Relaxation
    - Productivity
    - Consuming information (Reading PDF's, Word docs etc.)
  - Every segment contains 3 timers
    - When a timer is initiated no other instance of Microsoft Word is running
- Workloads should be compatible with all operating systems more recent than XP
  - In both 32 bit and 64 bit versions
- Workloads should be compatible with all office versions more recent than 2003
- Workloads should be able to run in loops/continuously
  - Make sure documents that are created are being cleaned up
  - Make sure all applications are closed at the end of a workload
- Applications should be started using a document and their File Type Association when possible
- The first timer executed by a user will be ignored

When testing the segment and loop length of your newly created workloads it is recommended to use a machine with at least 2vCPU's and 1GB of memory as these days that's the industry standard for VDI environments.

## 3 Language options

Login VSI workloads can be designed so that they can work on operating systems and applications that are multi-language. Before execution the engine will parse the workload in the language configured for the test. It will replace everything between the lang: .. :lang tags. For example lang:Outlook:lang will result in "Inbox" when English is selected, but it will return "Postvak" when Dutch is selected as language in the Management console.

Login VSI uses a dictionary file to provide the engine with the replacements needed for every specific language the workload has to support. The dictionary file can be found in the VSIShare at \\server\VSIShare\VSI\_Configuration\_CurrentTest\LanguageSettings.csv

Example LanguageSettings.csv

ID	English	Dutch	Japanese
Print	Print	Afdrukken	印刷
ExcelGoto	Go To	Ga Naar	ジャンプ

The language file can be extended to 200 languages (columns) and a virtually unlimited amount of items (rows).

## 4 Predefined variables

The Login VSI workload files support the use of Predefined variables. These variables make the workloads easier to use on different platforms and by different users as values are automatically updated to reflect the current environment.

Replacements supported by the Login VSI 4.0.2 engine relevant for the meta language:

All standard Windows environment variables	%APPDATA%, %USERNAME%, etc. See <a href="http://ss64.com/nt/syntax-variables.html">http://ss64.com/nt/syntax-variables.html</a> for more.
%VSI_Share%	Path to VSIshare
%VSI_Random_Recipient%	A random email recipient chosen from a predetermined pool. The predetermined pool cannot be changed.
%VSI_Random_Name%	A random name chosen from a predetermined pool. The predetermined pool cannot be changed.
%VSI_Random_doc%	A random Microsoft Word document chosen from the content pool.
%VSI_Random_docx%	A random Microsoft Word (docx format) document chosen from the content pool.
%VSI_Random_jpg%	A random JPG file chosen from the content pool.
%VSI_Random_mm%	A random Freemind MindMap document chosen from the content pool.
%VSI_Random_pdf%	A random PDF file chosen from the content pool.
%VSI_Random_ppt%	A random Microsoft Powerpoint document chosen from the content pool.
%VSI_Random_pptx%	A random Microsoft Powerpoint (pptx format) document chosen from the content pool.
%VSI_Random_pst%	A random PST file chosen from the content pool.
%VSI_Random_txt%	A random text file chosen from the content pool.
%VSI_Random_xlsx%	A random Microsoft Excel worksheet (xlsx format) chosen from the content pool.
%VSI_DataLocation%	The path to the content share used by this user.
%VSI_WebLocation%	The path to the websites used by this user.
%VSI_Web_BBC%	The number of versions available for the BBC website in the web content pool.
%VSI_Web_Engadget%	The number of versions available for the Engadget website in the web content pool.
%VSI_Web_Pinterest%	The number of versions available for the Pinterest website in the web

	content pool.
%VSI_Web_Player%	The number of versions available for the video content in the web content pool. Please note that VSI currently does not distinguish between different formats of the content (480p, 720p, 1080p).
%VSI_Web_TheVerge%	The number of versions available for the Verge website in the web content pool.
%VSI_Web_Wired%	The number of versions available for the Wired website in the web content pool.
%VSI_GroupDrive%	The path to the VSI group drive.
%VSI_UserHome%	The path to the VSI home drive.
%VSI_Workload_Language%	The language chosen for the workload.

#### Code Example

This code example shows how variables can be used to open a document with a variable name e.g. the username and automatically adjust it for every user.

```
App_Start("Logname", "c:\windows\system32", "notepad.exe", "Title",  
"Notepad", "", "%VSI_UserHome%\%USERNAME%.txt")
```



## 5 Adding an application to a workload

If you are thinking about extending the workload with an extra application follow these steps to add it to the workload. In this example you can see how to add a simple application to the workload, to load a document, read and manipulate the document and also close the document.

```
# Open a document. The default FileType Association (FTA) will open it in
the application associated with that file type.
App_Start("MyCustomApplication", %VSI_Share%\ ", "MyFileType.fta",
"Title", "FTA editor", "", "")
# wait for the window to appear and make sure it is maximized
App_Focus("MyCustomApplication", "Title", "FTA editor", "", "Maximize")
# Edit the newly opened document
VSI_Type_Time("MyCustomApplication", 25)
# Save the newly created document
VSI_Type_Fixed("MyCustomApplication ", "\s")
# Close the saved document
App_Close("MyCustomApplication ", "Title", "FTA editor")
```

Below is an example of adding an application to the workload without using file type association.

```
# Open new instance of Chrome and browse to gmail.
App_Start("Gmail", "C:\Program Files (x86)\Google\Chrome\Application\",
"chrome.exe", "Title", "Inbox", "", "mail.google.com")
# wait for the window to appear and make sure it is maximized
App_Focus("Gmail", "Title", "Inbox", "", "Maximize")
# Browse messages in Gmail
VSI_Type_Fixed("Gmail", "{down}{enter}")
# wait for the actual email to be loaded. Please note that this mailbox
was prepared to have only emails starting VSIMAIL.
App_Focus("Gmail", "Title", "VSIMAIL", "", "Maximize")
VSI_Browse("Gmail", 24, 22, 10, 5, 6000, 10000, 2500)
# Move back to inbox
VSI_Type_Fixed("Gmail", "{left}{enter}")
App_Focus("Gmail", "Title", "Inbox", "", "Maximize")
# Close Chrome
App_Close("Gmail", "Title", "Gmail")
```

## 6 Removing an application from the workload

If you want to remove an application that is part to one of our default workloads simply follow the steps below to remove it.

- Note the logname of the application you want to remove, the logname is always the first parameter of every function in the workload language e.g. Outlook
- Identify all lines that contain this logname and remove them
- Save the workload
- Run a test

# 7 Functions

## App\_Start

The App\_Start function can start applications and open documents leveraging the shell execute capabilities of the operating system. When you want to start an application it's recommended to use the file type association this way a document will automatically be opened and application virtualization techniques are supported out of the box.

### Syntax

```
App_Start("Logname ", " Working directory", " Executable or full path", " Window search mode", " Window title ", ["Window content "], ["Command line parameters"])
```

### Parameters

Logname	String		
Working directory	String		
Executable or full path	String		
Window search mode	Selection	Title,Class	
Window title	String		
[Window content]	String		
[Command line parameters]	String		

### Code Example

```
App_Start("Logname", "c:\windows\system32", "notepad.exe", "Title", "Notepad", "Some text found in the document or app", "H:\document.txt")
```

### Related functions

[App\\_Close](#), [App\\_Focus](#)

## App\_Close

This function closes applications or windows

### Syntax

```
App_Close("Logname", "Searchmode", "Searchstring", ["SearchContent"])
```

### Parameters

Logname	String		
Searchmode	Selection	Title,Class	
Searchstring	String		
[SearchContent]	String		

### Code Example

```
App_Close("MyLogName", "Title", "Notepad", "Text in document or app")
```

### Related functions

[App\\_Start](#), [App\\_Focus](#)

## App\_Focus

This function makes sure an application window is on top but can also be used to minimize or maximize an application. If an application window is already on top it will be ignored.

### Syntax

```
App_Focus("Logname", "Searchmode", "Searchstring", ["SearchContent"], ["Maximize"])
```

### Parameters

Logname	String		
Searchmode	Selection	Title,Class	
Searchstring	String		
[SearchContent]	String		
[Action]	Selection	Maximize,Minimize	

### Code Example

```
App_Focus("MyLogName", "Title", "Notepad", "Text in document or app", "Maximize")
```

### Related functions

[App\\_Start](#), [App\\_Close](#)

## PDF\_Print

Will start a print action on the currently active window and generate a PDF file using DoroPDF writer. When "Office" is used in the Logname the print sequence will automatically be adapted to the correct office version.

### Syntax

Normal:

```
PDF_Print("PDFWriter", "PDF Printer Windowname", ["Filename"])
```

From a Microsoft Office application:

```
PDF_Print("PDFWriterOffice", "PDF Printer Windowname", ["Filename"])
```

### Parameters

Logname	String		
PDF Printer Windowname	String	"Doro PDF Writer"	Do not use a different printer window name. This parameter is in preparation for supporting other PDF printers
Filename	String		Optional, prints to the specified file name. If this parameter is not specified it will default to "%VSI_UserHome%\Output\PDFPrint.pdf"

### Code Example

```
PDF_Print("PDFWriter", "Doro PDF writer",
"%VSI_Userhome%\Output\PWP1Print.pdf")
```

### Related functions

[App\\_Focus](#)

## Segment

Will split the workload in an extra segment. A typical segment is 12 minutes in length of which 10 minutes are active and 2 minutes are idle users initiating their Login VSI test will automatically be assigned the segment they start in therefore load balancing application usage.

### Syntax

Segment("Logname",Segment ID)

### Parameters

Logname	String		
Segment ID	Integer		

### Code Example

```
segment("workload",1)
```

### Related functions

[VSI Buffer](#)

## VSI\_Buffer

As it's recommended to make segments 10 minutes long finishing with a 2 minute idle, this function can make sure that when a segment is too short or executed too slow it is stretched or shortened to fit back in the 12 minutes timeframe. The goal of this buffer is to smooth the test results and allow for easier analysis of the data.

### Syntax

```
Buffer("Logname",Minimum wait)
```

### Parameters

Logname	String		
Minimum wait	Integer		Minimum time to wait in seconds

### Code Example

```
Buffer("workload", 30)
```

### Related functions

[Segment](#)



## Set\_DefaultPrinter

The Set\_DefaultPrinter function can be used to change the default printer within a test session.

### Syntax

```
Set_DefaultPrinter("LogName", "Printername")
```

### Parameters

Logname	String		
Printername	String		

### Code Example

```
Set_DefaultPrinter("LogName", "DoroPDF Writer")
```

### Related functions

[PDF Print](#)

## Web\_Start

Starts a new instance of Microsoft Internet Explorer and navigates to the defined URL

### Syntax

```
Web_Start("logname","URL")
```

### Parameters

Logname	String		
URL	String		

### Code Example

```
web_start("Internet","http://www.LoginVSI.com")
```

### Notes

Please note that every instance of Internet Explorer opened through Web\_Start should also be closed by Web\_Quit. Failure to do so may cause unexpected results.

### Related functions

[Web\\_Quit](#), [Web GoTo](#)

## VSI\_Browse

Browses "Scrolls" through the currently active window acting as a real user

### Syntax

VSI\_Browse("LogName", Time, Maxdown, AmountDown, Upamount, ReadInterval, DownInterval)

### Parameters

Logname	String		
Time	Integer		Time in seconds to scroll through the currently active window.
MaxDown	Integer		How many times the down key can be pressed before the end of the document is reached.
AmountDown	Integer		How many times the down key should be pressed per scroll operation.
UpAmount	Integer		How many times the up key should be pressed per scroll operation.
ReadInterval	Integer		How long to pause between scroll operations.
DownInterval	Integer		How much time in milliseconds between down keystrokes.
UpInterval	Integer		How much time in milliseconds between up keystrokes.
Direction	Selection	1,2,3,4	1: Up/Down 2: Right/Left 3: Down / up 4: Left / Right

### Code Example

```
VSI_Browse("Internet", 120, 90, 8, 4, 3000, 500)
```

### Related functions

[VSI Type Fixed](#), [VSI Type Time](#)

## Web\_Quit

Closes the defined browser instance using the Logname provided when starting Internet Explorer

### Syntax

```
Web_Quit("logname")
```

### Parameters

Logname	String		
---------	--------	--	--

### Code Example

```
web_Quit("Internet")
```

### Related functions

[Web\\_Start](#), [Web\\_GoTo](#)

## Web\_GoTo

Navigate an existing Internet Explorer instance to another website

### Syntax

```
Web_GoTo("logname","URL")
```

### Parameters

Logname	String		
URL	String		

### Code Example

```
web_goto("Internet", "http://www.loginvsi.nl")
```

### Related functions

[Web\\_Quit](#), [Web\\_Start](#)

## VSI\_Type\_fixed

Type the defined text

### Syntax

```
VSI_Type_fixed("Logname","Text",SendKeyDelay)
```

### Parameters

Logname	String		
Text	String		
[SendKeyDelay]	Integer		The delay in milliseconds between keystrokes.

### Code Example

```
VSI_Type_fixed("Logname","Text to type",150)
```

### Related functions

[VSI\\_Type\\_Time](#)

## VSI\_Type\_Time

Types random text from the input file limited by a number of seconds

### Syntax

```
VSI_TYPE_Time("Logname","Time",[SendKeyDelay])
```

### Parameters

Logname	String		
Time	Integer	seconds	
[SendKeyDelay]	Integer		The delay in milliseconds between keystrokes.

### Code Example

```
VSI_Type_Time("Logname", 60, 150)
```

### Related functions

[VSI\\_Type\\_Fixed](#)

## VSI\_File\_Copy

Copy a file from source to target.

### Syntax

```
VSI_File_Copy("logname","Source","Target")
```

### Parameters

Logname	String		
Source	String		
Target	String		

### Code Example

```
VSI_File_Copy("logname","c:\temp\document.doc","c:\temp\document2.doc")
```

### Related functions

[VSI\\_File\\_Delete](#), [VSI\\_Random\\_File\\_Copy](#)



## VSI\_Random\_File\_Copy

Copy a random file from the file library to specified location

### Syntax

```
VSI_Random_File_Copy("logname","Filetype","Target")
```

### Parameters

Logname	String		
Filetype	String		
Target	String		

### Code Example

```
VSI_Random_File_Copy("logname","doc","h:\document.doc")
```

### Related functions

[VSI File Delete](#), [VSI Random File Copy](#)

## VSI\_File\_Delete

Deletes the specified file

### Syntax

```
VSI_File_Delete("logname","FilePath")
```

### Parameters

Logname	String		
FilePath	String		

### Code Example

```
VSI_File_Delete("logname","c:\temp\document.doc")
```

### Related functions

[VSI File Copy](#)

## VSI\_Reg\_Write

Write a registry value

### Syntax

```
vsi_regwrite("Logname","KeyName","ValueName","Type")
```

### Parameters

Logname	String		
KeyName	String		
ValueName	String		
Type	Selection	"REG_SZ", "REG_MULTI_SZ", "REG_EXPAND_SZ", "REG_DWORD", "REG_QWORD", "REG_BINARY"	

### Code Example

```
vsi_regwrite("Appname", "HKEY_CURRENT_USER\Software", "VSI", "REG_SZ", "VSI40T  
EST")
```

### Related functions

[VSI\\_RegDelete](#)

## VSI\_RegDelete

Delete a registry value

### Syntax

```
VSI_Regdelete("Logname","KeyName",["ValueName"])
```

### Parameters

Logname	String		
KeyName	String		
[ValueName]	String		

### Code Example

```
VSI_Regdelete("AppName", "HKEY_CURRENT_USER\Software", "VSI")
```

### Related functions

[VSI\\_Reg\\_Write](#)

## VSI\_Sleep

Wait for specified number of seconds before continuing next action

### Syntax

VSI\_Sleep(Time)

### Parameters

Time	Integer		In seconds
------	---------	--	------------

### Code Example

```
VSI_Sleep(30)
```

### Related functions

[Workload\\_Idle](#), [Workload\\_RandomIdle](#)

## Workload\_Idle

Wait for specified number of seconds before continuing next action and allows the user to specify a message that will be shown on screen during the idle time.

### Syntax

```
Workload_IDLE("Logname", Time, "Message")
```

### Parameters

Logname	String		
Time	Integer		In seconds
Message	String		

### Code Example

```
workload_IDLE("workload", 30, "%USERNAME% is IDLE")
```

### Related functions

[VSI Sleep](#), [Workload RandomIdle](#)

## Workload\_RandomIdle

Wait for random amount of time within the timeframes specified before continuing next action and allows the user to specify a message that will be shown on screen during the idle time.

### Syntax

```
Workload_RandomIdle("Logname", MinIdle, MaxIdle, "Message")
```

### Parameters

Logname	String		
MinIdle	Integer		In seconds
MaxIdle	Integer		In seconds
Message	String		

### Code Example

```
workload_RandomIdle("Logname", 5, 15, "User is on the phone")
```

### Related functions

[VSI Sleep](#), [Workload Idle](#)

## VSI\_ShellExecute

Perform a shellexecute action.

### Syntax

```
VSI_ShellExecute("Logname", "filename", "parameters", "workingdir", "verb", "showflag")
```

### Parameters

Logname	String		
Filename	String		
Parameters	String		
Workingdir	String		
Verb	String		
Showflag	Selection	Minimize/Maximize	

### Code Example

```
VSI_ShellExecute("MyApp", "c:\temp\document.txt", "", "", "", "Maximize")
```

### Related functions

[VSI\\_ShellExecuteWait](#)



## VSI\_ShellExecuteWait

Perform a shellexecute action and wait for it to complete before continuing the workload. This function is meant to replace the way Login VSI 3 handled customization. Please note that if the action never finishes than the workload will be paused forever.

### Syntax

```
VSI_ShellExecuteWait("Logname", "filename", "parameters", "workingdir", "verb", "showflag")
```

### Parameters

Logname	String		
Filename	String		
Parameters	String		
Workingdir	String		
Verb	String		
Showflag	Selection	Minimize/Maximize	

### Code Example

```
VSI_ShellExecutewait("MyApp", "%VSI_Share%\MyCustomApp.exe", "", "", "", "Maximize")
```

### Related functions

[VSI\\_ShellExecute](#)

## VSI\_FileWriteToLine

Write text to a file

### Syntax

```
VSI_FileWriteToLine("Logname", "filename", [line number], ["text"])
```

### Parameters

Logname	String		
Filename	String		
Line Number	Integer		
Text	String		

### Code Example

```
VSI_FileWriteToLine("Logname", "c:\temp\example.txt", 1, "Hello world")
```

### Related functions

-

## VSI\_Timer

Perform a timer event. The timer event will perform the response time measurements required for the VSI analyzer to be able to determine if the environment is saturated.

### Syntax

```
VSI_Timer()
```

### Parameters

--	--	--	--

### Code Example

```
VSI_Timer()
```

### Related functions

-