



SHUNRA VE Desktop for HP Software and for Microsoft Visual Studio

USER MANUAL

Professional Edition
Version 5.0



Copyright Notice

© 2008 Shunra Software Ltd. Shunra is a trademark of Shunra Software Ltd. All rights reserved.

This document is for information purposes only. Shunra Software Ltd. makes no warranties, expressed or implied. Shunra Virtual Enterprise, Shunra Virtual Enterprise Suite, Shunra VE, Shunra VE Suite, VE Network Appliance, VE Modeler, VE User Automation, VE User Automation for Lab Automation, VE User Automation for Web Users, VE Automation Agent, VE Reporter, VE Predictor, VE Profiler, VE Network Catcher, VE Network Catcher lite, VE Endpoint, VE Endpoints, VE MCast, VE QoS, NetworX, and SPEL are trademarks of Shunra Software Ltd.

Microsoft, Visio and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other brand and product names are trademarks or registered trademarks of their respective holders.

Information in this document is subject to change without notice and does not represent a commitment on the part of Shunra Software Ltd. The software described in this document is furnished under license agreement. The software may be used only in accordance with the terms of this agreement.

No part of this manual may be reproduced or transmitted in any form or by any means, for any purposes other than the purchaser's personal use, without the express written permission of Shunra Software Ltd.

US Headquarters
Shunra Software Ltd.
1800 JFK Blvd.
Philadelphia
PA 19103
USA
Tel: (215) 564 4046

International Headquarters
Shunra Software Ltd.
20 Hata'as Street
Kfar Saba, 44425
Israel
Tel: +972 9 764 3743
Fax: +972 9 764 3754

Toll Free: 1 877 474 8672
Fax: (215) 564 4047

Email: info@shunra.com
Website: www.shunra.com

Table of Contents

Chapter 1: VE Desktop Professional

Introduction	.1 - 1
Integration with HP Software	.1 - 2
VE Desktop Integration with HP LoadRunner	.1 - 4
VE Desktop Integration with HP Performance Center	.1 - 5
VE Desktop for MS Visual Studio	.1 - 6
VE Desktop client	.1 - 6
VE Desktop for HP LoadRunner	.1 - 7
VE Desktop for HP Performance Center	.1 - 8
VE Network Editor	.1 - 8
VE Desktop server	.1 - 8
VE Analyzer	.1 - 9
VE Network Catcher	.1 - 9
Licensing for HP Software	.1 - 9
Evaluation License	1 - 10
Licensing for HP LoadRunner Integration	1 - 10
Licensing for HP Performance Center Integration	1 - 10
Licensing for MS Visual Studio	1 - 11
Integrated WAN Emulation Process for HP Software	1 - 11
WAN Emulation on HP LoadRunner	1 - 11
WAN Emulation on HP Performance Center	1 - 12

Chapter 2: Installing and Uninstalling VE Desktop

Components	.2 - 1
System Requirements	.2 - 2
Configuring VE Desktop client with RDP	.2 - 4
Installing VE Desktop on HP Software	.2 - 6
Installing VE Desktop on HP Performance Center	.2 - 6
Installing the VE Desktop for HP LoadRunner	.2 - 9

Installing the VE Desktop client on the HP Load Generator	2 - 12
Installing VE Desktop for MS Visual Studio	2 - 15
Uninstalling VE Desktop on HP Software	2 - 16
Uninstalling VE Desktop for MS Visual Studio	2 - 16
Silent Installation	2 - 17
Silent Uninstall	2 - 18
VE Desktop client Licenses Overview	2 - 19
Server-Based Licensing	2 - 19
Node-Locked VE Desktop client Licenses	2 - 19
Requesting a License Key	2 - 19
Installing a New License	2 - 21

Chapter 3: Conducting WAN Emulation in HP Software and MS Visual Studio 3 - 1

Incorporating WAN Emulation in HP Performance Center	.3 - 1
Incorporating WAN Emulation in HP LoadRunner	.3 - 2
Incorporating WAN Emulation in MS Visual Studio	.3 - 4
Conducting Tests in Basic Mode	.3 - 6
Conducting Tests in Advanced Mode	.3 - 8
Conducting Tests in Professional Mode	.3 - 9
Configuring Additional Settings	3 - 14
Including Additional Test Result Statistics	3 - 16
File Types	3 - 19

Chapter 4: Monitoring Performance 4 - 1

The PerfMon Counters	.4 - 1
Viewing Statistics in the HP LoadRunner Analysis	.4 - 4
Viewing Statistics in the VE Reporter	.4 - 4

Chapter 5 VE Desktop API 5 - 1

Guidelines for Working with the API	.5 - 1
General Guidelines and Limitations	.5 - 2

Notes about VE Analyzer Reports	.5 - 2
Transaction Markers vs. Time Stamp	.5 - 3
Working with the External DLL Interface	.5 - 4
Process Overview	.5 - 4
Automation Script Requirements	.5 - 5
Supporting Files	.5 - 5
Working with the COM Interface	.5 - 6
VE Desktop COM Interface Definitions	.5 - 6
Working with the CLI	.5 - 7
API Commands and Parameters List	.5 - 7
API Commands and Parameters Details	5 - 10
StartAdvanced	5 - 11
StartProfessional	5 - 13
StopTest	5 - 15
StartTransaction	5 - 16
StartNextTransaction	5 - 18
NextNetwork	5 - 19
RestartBusinessProcess	5 - 20
NextBusinessProcess	5 - 21
StopTransaction	5 - 22
SaveRun	5 - 24
UploadRun	5 - 26
SetLogFile	5 - 28
ActivatePacketList	5 - 30
CancelPacketList	5 - 32
LicenseCheckout	5 - 33
LicenseCheckin	5 - 35
GetRunningTestDescription	5 - 36
GetErrorString	5 - 38
IsEmulationRunning	5 - 40
SetAutoAnalysisParams	5 - 41
SetKeepRunHistory	5 - 43
GetKeepRunHistory	5 - 45
SetIPFilterParameters	5 - 47

GetIPFilterParameters	5 - 49
GetIPFilterItem	5 - 51
GetIPFilterCount	5 - 53
AddIPFilterItem	5 - 55
RemoveIPFilterItem	5 - 57
ClearIPFilter	5 - 59
SetPacketListParameters	5 - 60
GetPacketListParameters	5 - 63
SetAutoSaveTestResults	5 - 66
SetResultFolderPrefix	5 - 68
GetAutoSaveTestResults	5 - 70
SetCalculateTopTalkers	5 - 72
GetCalculateTopTalkers	5 - 74
VEDStats_GetCurrentStatistics	5 - 76
VEDStats_GetErrorString	5 - 79
VEDStats_GetLastError	5 - 81
GetHostID	5 - 83
SetLicenseKey	5 - 85

Appendix A: Single User Script Modification A - 1

Appendix B: Presets.xml file B - 1

Index i - 1

1

VE Desktop Professional Introduction

Shunra VE Desktop is a software-based testing environment that enables developers and QA teams to understand exactly how an application will perform on the production network. Tests can incorporate multiple network configurations to accurately assess performance that will be observed by end-users in multiple geographic locations, all from within a single test lab environment.

VE Desktop is packaged into several versions. This section describes the use of VE Desktop with HP LoadRunner and HP Performance Center. Refer to **VE Desktop for MS Visual Studio** [page 1-6](#), for information about the integration with MS Visual Studio.

Other versions of VE Desktop provide support for:

- ◆ Testers doing manual testing
- ◆ Testers using automated functional testing using tools such as HP QuickTest Pro

In addition, VE Desktop includes an optional server component that can be used to centralize storage of network emulation configurations and test results. Shunra recommends that new users of VE Desktop for HP Software begin without using the VE Desktop server.

WAN emulation tests consist of a number of pre-defined parameters including the virtual origin and destination of traffic, connec-

tion quality and connection type. This data can be captured from production networks using an additional Shunra product, Network Catcher, or can be defined based on data provided by network engineers. Other test data such as a set of network conditions to be tested within a test plan, and definitions of transactions to be tested can also be defined in advance.

Shunra VE Desktop enables users to see exactly how their application will perform over different network configurations, from different geographic locations—without ever leaving their workstations. WAN emulation tests consist of a number of pre-defined parameters including the virtual origin and destination of traffic, connection quality, connection type, and business processes, or a list of transactions that end-users need to conduct.

The Shunra VE Desktop client component can be configured:

- ◆ With the VE Desktop server
- ◆ As a standalone application
- ◆ Integrated with the HP LoadRunner and the HP Performance Center
- ◆ Integrated with MS Visual Studio

Integration with HP Software

The integration of Shunra's VE Desktop with the HP LoadRunner and HP Performance Center applications provides a software-based solution that simulates a wide area network link from each HP Load Generator to the server(s) of the application undergoing the performance test. Each HP Load Generator can be configured to represent a single remote location. All Virtual Users (VUsers) on that HP Load Generator (subject to filtering rules) are routed through the emulated network. HP LoadRunner and HP Performance Center generate the application transaction load while the VE Desktop component augments the performance test by applying the anticipated production network conditions to the test. As

a result, the performance test presents a realistic emulation of load and network conditions.

Test statistics such as HP LoadRunner-measured transaction response times include the impact of the network. In addition, Web Page Diagnostics reflect the network cost of complex transactions. Additional Shunra-provided WAN emulation runtime statistics can be included as part of the results obtained from the application performance test, providing an inclusive representation of the network load caused by a group of users at each location. The statistics obtained from the performance monitors on the HP Load Generators are presented as PerfMon counters and can be added to the set of Windows Performance Monitors for the load test. On the HP Performance Center, the statistics are visible on the Windows Performance Monitors on the HP Performance Center User Site.

Normally, all data from each test will be stored as a standard part of the load test results. Optionally, the test can be configured to generate a set of Shunra analysis reports (one set for each load generator) after each test. In addition, a straightforward set of APIs can be added to existing scripts to cause Shunra to include a more detailed analysis for the key transactions within a test, to explain why certain key transactions are excessively slow or are timing out.

VE Desktop can be configured generate reports after each test, and automatically post HTML reports and test results to a shared network drive for additional analysis. The VE Analyzer provides the additional analysis and the ability to export data and charts to CSV or MS-Word.

Optionally, the test can be configured to generate a set of Shunra analysis reports per HP Load Generator after each test. In addition, a straightforward set of APIs can be added to existing scripts to cause VE Desktop to include a more detailed analysis of key

transactions within a test, and to explain why specific transactions are excessively slow or are timing out.

VE Desktop Integration with HP LoadRunner

The WAN emulation parameters can be defined within the HP LoadRunner Controller user interface for each HP Load Generator during configuration of the performance test. Shunra VE Desktop 5.0 includes specialized integration interfaces to work seamlessly within a HP LoadRunner 9.5 environment. For HP LoadRunner 9.0 and 9.1 environments, a simple API can be added to existing HP LoadRunner scripts to enable testing. See **VE Desktop API** and the sample script located in the Samples directory of your installation.

VE Desktop for HP LoadRunner includes the following components:

- ✦ **VE Desktop client** ([page 1-6](#))
- ✦ **VE Desktop for HP LoadRunner** ([page 1-7](#))

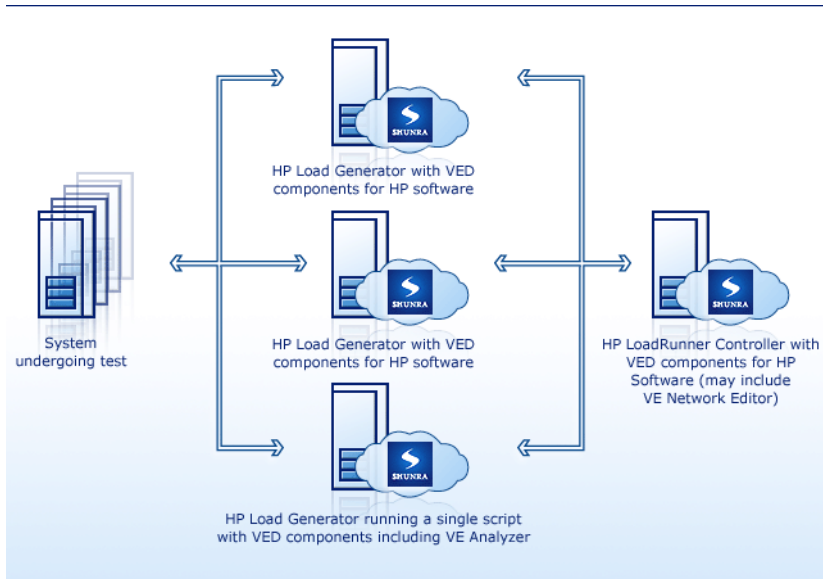
The following optional components are also included with your Shunra license for use with HP Software:

- ✦ **VE Network Editor** ([page 1-8](#))
- ✦ **VE Analyzer** ([page 1-9](#))

These components requires a separate license:

- ✦ **VE Desktop server** ([page 1-8](#))
- ✦ **VE Network Catcher** ([page 1-9](#))

The following diagram shows typical configuration of the HP LoadRunner with the VE Desktop components:



VE Desktop Integration with HP Performance Center

The VE Desktop for HP Performance Center provides an integrated user interface that controls the Shunra network emulation capabilities from within HP Performance Center User Site. The the VE Desktop for HP Performance Center ([page 1-7](#)) component must be installed on each User Site.

In addition, every HP LoadRunner Controller and the HP Load Generators that are configured with it per HP Performance Center must be configured with the components listed in **VE Desktop Integration with HP LoadRunner**([page 1-4](#)). The flow is the same on the HP Performance Center as on the HP LoadRunner Controller, except that the test is configured on the User Site.

VE Desktop for MS Visual Studio

Using the integration with MS Visual Studio, you can integrate WAN emulation into your development environment, to test the reaction of your software in real network conditions, in both release and debugging mode, refer to **Incorporating WAN Emulation in MS Visual Studio** ([page 1-6](#)).

VE Desktop client

VE Desktop client for HP LoadRunner is the component that runs on each HP Load Generator. During a load test, all configuration is done centrally via the HP LoadRunner Controller or HP Performance Center. During a load test you do not access the VE Desktop Client user interface on the HP Load Generator, as all configuration is done centrally via the HP LoadRunner Controller or HP Performance Center. The VE Desktop Client includes a Shunra network emulation driver that applies selective traffic impairments for traffic flows between the load generator and the application servers being tested. The VE Desktop client also includes a windows user interface which is not normally accessed during load tests.

VE Desktop client can also be installed independently from HP LoadRunner for manual testing, and is included as part of VE Desktop Professional. Refer to the VE Desktop Professional Client User Manual for additional information.

Emulation is configured via the HP LoadRunner Controller or HP Performance Center for each HP Load Generator. Three modes of emulation are supported:

- ◆ **Basic Mode:** enables users to select the virtual origin and destination of traffic, connection quality, and connection type, then apply these impairments to all traffic entering and leaving the local network card.

- ◆ **Advanced Mode:** enables users to manually enter Bandwidth, Latency and Packet Loss parameters, then apply these impairments to all traffic entering and leaving the local network card.
- ◆ **Professional Mode:** enables users to import pre-defined WAN emulation files from a file system (which can be located on the HP LoadRunner Controller or the HP Performance Center) or from the VE Desktop server. These files can contain more complex network conditions configured in the Network Editor or data recorded by the VE Network Catcher.

Note: Professional mode also supports use of a test plan file. This is required when using the Shunra API and VE Analyzer to include a more detailed assessment of performance. See **Including Additional Test Result Statistics** ([page 3-16](#)) for more information.

VE Desktop for HP LoadRunner

This component provides the user interface is used to define the configuration for network emulation. The user interface is launched directly within the HP LoadRunner 9.5 controller to configure the test environment. This component also provides an integrated license management mechanism required to include emulation within HP LoadRunner 9.5. For HP LoadRunner 9.0 and 9.1, this component is not used, and manual script changes and licensing are required. For more information on API level integration see VE Desktop API (page 5-1).

Note: After your evaluation trial period ends, you must install a valid license on each Controller in order to include WAN emulation in your tests.

VE Desktop for HP Performance Center

This component is installed on HP Performance Center 9.5 and above, and provides the web user interface that is used to configure network emulation within HP Performance Center, and the HP LoadRunner controllers that are integrated with the HP Performance Center. For HP Performance Center 9.0 and 9.1, Shunra network emulation is available via the API, see **VE Desktop API** ([page 5-1](#)).

VE Network Editor

Shunra NetworX files are required when tests are run using Professional Mode for emulation. You can use the VE Network Editor to define complex network scenarios by modifying scripts to emulate your specific impairments. These files can be the sample network files, or the files obtained from recordings by the VE Network Catcher. This optional component can be installed on any of the HP LoadRunner or HP Performance Center components, or on a separate computer, refer to the VE Network Editor User Manual.

VE Desktop server

VE Desktop server is used to create or edit WAN emulation files, which define the parameters of tests that need to be conducted by users, houses WAN emulation files which can be downloaded by users, and acts as a central repository for the emulation results. Additionally, VE Desktop server can manage licenses depending upon the license scheme. For more information, refer to the VE Desktop server User Manual.

VE Analyzer

Shunra VE Analyzer conducts an analysis based on emulation results which assists in pinpointing factors that negatively impact an application's operation across a network. Resulting data is displayed in informative reports that provide insight into an application's operation. It can be installed on any of the HP LoadRunner components, for more details refer to the VE Reporter User Manual.

VE Network Catcher

VE Network Catcher is a web-based application that captures and records production network parameters including latency, jitter, packet loss and router changes, directly from the enterprise environment. These network parameters can then be replayed on the VE Desktop or on the VE Network Appliance, for more details refer to the VE Network Catcher User Manual.

Licensing for HP Software

HP LoadRunner seamlessly integrates Shunra licensing for WAN Emulation. Licensing is managed by the HP LoadRunner Controllers. In the HP Performance Center, neither the User Site nor the Server requires license components, but each HP Performance Center host in the configuration requires its own Shunra License Manager.

The node-locked license is enabled per usage, so that for each emulation played by the VE Desktop client, the HP LoadRunner Controller provides a license for that emulation. The license is revoked once the play has completed. License for Analysis with the VE Analyzer for up to 24 hours after the emulation has completed may also be included. The license status and type can be

seen in the License Manager on the HP LoadRunner Controller or HP Performance Center host.

Evaluation License

A trial license for the HP LoadRunner integration is provided once per VE Desktop client host, permitting 10 emulations for a duration of up to 30 minutes each, including 24 hours of analysis after the emulation has completed. A message indicates the remaining number of plays at each trial WAN emulation.

If your configuration includes a Shunra VE Desktop server, licensing may be handled in an alternate manner, refer to the VE Desktop server User Manual.

Licensing for HP LoadRunner Integration

Shunra for HP LoadRunner is licensed per HP LoadRunner Controller. Each HP LoadRunner Controller requires a node locked license that controls the number of Virtual Users that can be impacted by the VE Desktop network simulation. The VE Desktop for HP LoadRunner enables the licensing of the VE Desktop client on the HP Load Generator during the test.

VE Desktop can be licensed by:

- ◆ VE Desktop for HP LoadRunner components installed on the HP LoadRunner Controller
- ◆ Node locked to the HP Load Generator machine
- ◆ VE Desktop server

Licensing for HP Performance Center Integration

To license the HP Performance Center to include network simulation, install the appropriate license for each HP Performance Cen-

ter host. VE Desktop for HP Performance Center doesn't require a license.

Licensing for MS Visual Studio

Once the Evaluation license has expired, a node-locked license is required, see **VE Desktop client Licenses Overview** on [page 2-19](#).

Integrated WAN Emulation Process for HP Software

You can add WAN emulation parameters to the test environment on both the HP LoadRunner and the HP Performance Center; the same functionality can be obtained on both systems.

WAN Emulation on HP LoadRunner

- 1 Install the VE Desktop for HP LoadRunner on the HP LoadRunner Controller
- 2 Install the VE Desktop client on each HP Load Generator
- 3 On the HP LoadRunner Controller create a new test including WAN emulation settings
- 4 The HP LoadRunner Controller provides a license and starts the test on each HP Load Generator; during the test, transactions are sent and received from the HP Load Generators back and forth to the Application Server (of the application being tested)
- 5 During the test, on the HP LoadRunner Controller view runtime statistics that are automatically collected from the Shunra monitors on each HP Load Generator
- 6 The HP LoadRunner Controller stops the load test on the HP Load Generators
- 7 In the HP LoadRunner Analysis, view the reaction of the HP LoadRunner transactions with the WAN emulation conditions

WAN Emulation on HP Performance Center

- 1** Install the VE Desktop for HP Performance Center on the HP Performance Center User Site.
- 2** Install the Shunra VE Desktop components for each HP LoadRunner Controller and HP Load Generator in the configuration.
- 3** On the HP Performance Center create a new test including WAN emulation settings.
- 4** The HP LoadRunner Controller provides a license and starts the test on each HP Load Generator; during the test, transactions are sent and received from the HP Load Generators back and forth to the Application Server (of the application being tested).
- 5** During the test, on the HP Performance Center User Site, view runtime statistics that are automatically collected from the Shunra monitors
- 6** The HP LoadRunner Controller stops the load test on the Load Generators
- 7** In the HP Performance Center Analysis, view the reaction of the test transactions with the WAN emulation conditions.

2

Installing and Uninstalling VE Desktop Components

The installation process for HP Software comprises the following steps :

- 1** If your configuration includes the HP Performance Center 9.5 and above, install the VE Desktop for HP Performance Center (see [page 2-6](#)) on the HP Performance Center User Site, and the VE Desktop for HP LoadRunner on the HP Performance Center Hosts (see [page 2-9](#)). Install the VE Desktop client on each HP Load Generator and HP Performance Center Host which emulates network impairments (see [page 2-12](#)).
- 2** Install and configure the VE Desktop for HP LoadRunner on the HP LoadRunner Controller (see [page 2-9](#)).
- 3** Install the VE Desktop client on each HP Load Generator, either on each machine, or by silent installation to multiple HP Load Generators (see [page 2-12](#)). The VE Analyzer, VE Desktop Server and Network Editor are optional components that can be installed on any of the machines that meet the system requirements.

System Requirements

The minimum requirements for VE Desktop on HP Performance Center User Site and HP LoadRunner Controller are as follows:

Processor	Intel Pentium IV Class or AMD Athlon XP (32 bit)
Memory	1GB RAM or more
Free Hard Disk Space	5 GB of free disk space (Space on disk is needed for storing the test configurations and user uploaded results.)
Network Adapter	Ethernet or Fast Ethernet Network Interface Card
Browser	Internet Explorer 7.0 or later (recommended) Internet Explorer 6.0 or later (supported)
Operating System (English versions only)	Windows XP Professional (SP2, SP3) Windows 2003 Server Enterprise Edition, Standard Edition SP2, SP1 (Recommended)
HP LoadRunner	HP LoadRunner Controller 9.1 (emulation via API only) HP LoadRunner Controller 9.5 and above (includes Shunra WAN emulation UI)
HP Performance Center	HP Performance Center 9.1 (emulation via API only) HP Performance Center 9.5 and above (includes Shunra WAN emulation UI)
Virtual Machine	VMware ESX 6.0 or later (3.0 or later supported)

The minimum requirements for VE Desktop client, VE Network Editor and VE Analyzer are as follows:

Processor	Intel Pentium IV Class or AMD Athlon XP (32 bit; multiple processors supported)
Memory	512 MB RAM or more, 1GB RAM recommended
Free Hard Disk Space	5 GB of free disk space
Network Adapter	Ethernet or Fast Ethernet Network Interface Card
Browser	Internet Explorer 6.0 or later
Operating System (English versions only)	Windows XP Professional (SP2, SP3) Windows 2003 Server (Standard or Enterprise) Windows Vista (32 bit)
HP LoadRunner	HP LoadRunner Controller 9.1 (emulation via API only) HP LoadRunner Controller 9.5 and above (includes Shunra WAN emulation UI)
HP Performance Center	HP Performance Center 9.1 (emulation via API only) HP Performance Center 9.5 and above (includes Shunra WAN emulation UI)
Virtual Machine	VMware ESX 6.0 or later (recommended) VMware ESX 3.0 or later (supported)

Configuring VE Desktop client with RDP

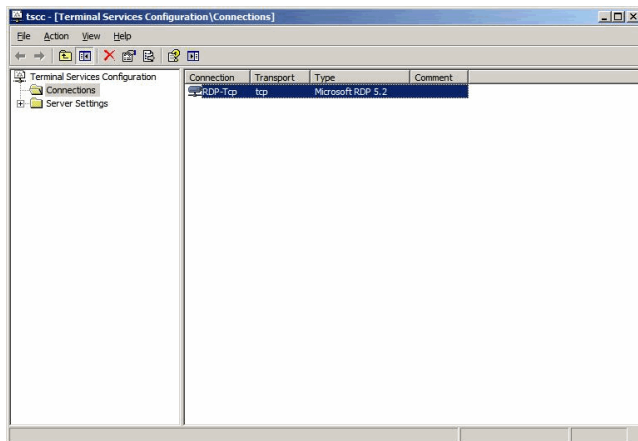
Note: This section is only applicable when installing with Windows 2003 Server.

RDP only needs to be configured if you want to install VE Desktop client over RDP, and must be configured on relevant client hosts **before** installation.

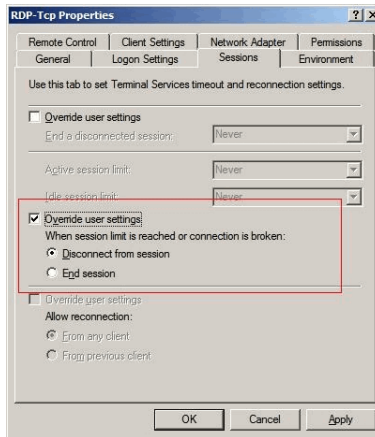
The central aspect to configuring RDP for this scenario is to change the machine policy regarding remote session disconnections. During the installation of VE Desktop client a driver is installed which results in a short network disconnection. This creates a problem with RDP and causes the installation to fail. The following configures RDP so that this short network disconnection does not impact installation.

To configure RDP:

- 1 On the host onto which the VE Desktop client will be installed, from the Start Menu, choose Administrative Tools/ Terminal Services Configuration; the Terminal Services Console opens:



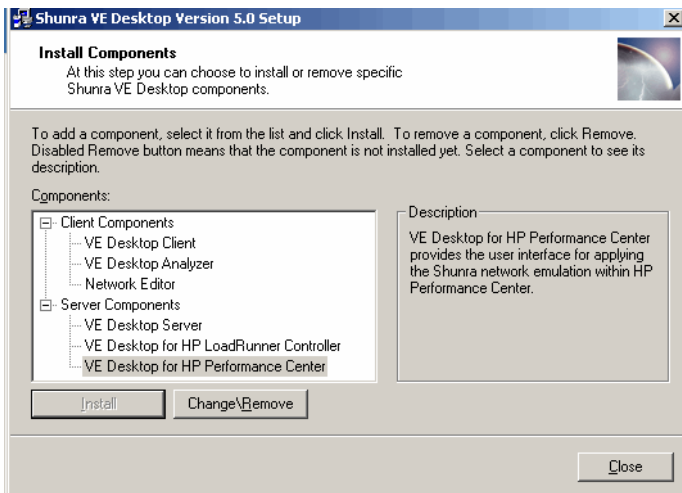
- 2 Make sure the Override user settings option is selected, together with the Disconnect from session radio button, as shown below.



- 3 Close RDP session and then re-start it; RDP has been configured.

Installing VE Desktop on HP Software

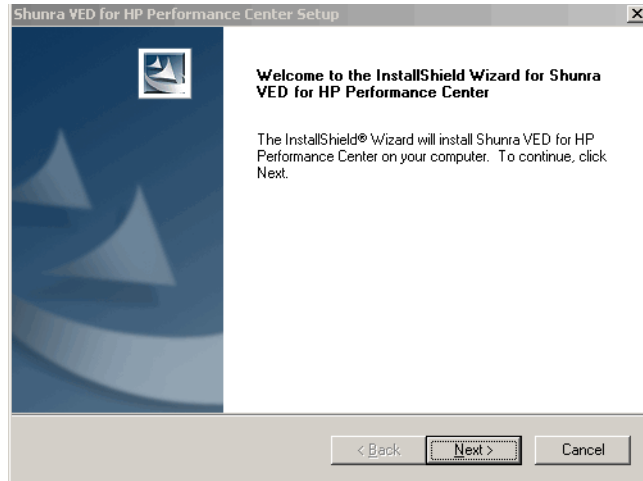
The VE Desktop components are contained in the Shunra VE Desktop Setup, which opens from VEDSuiteInstallerForHP.exe. Use this tool to install or remove any of the VE Desktop components. The standalone VE Desktop client, which can be installed on HP Software, or independently, is also included.



Installing VE Desktop on HP Performance Center

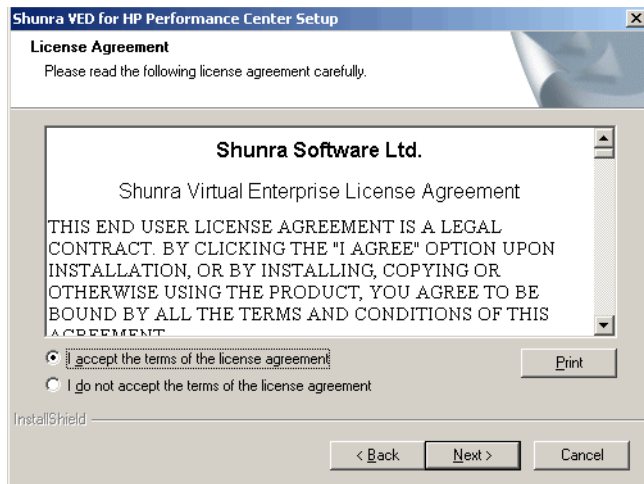
 **To install the VE Desktop for HP Performance Center:**

- 1 On the HP Performance Center User Site, open the VED-forHPPC.exe.

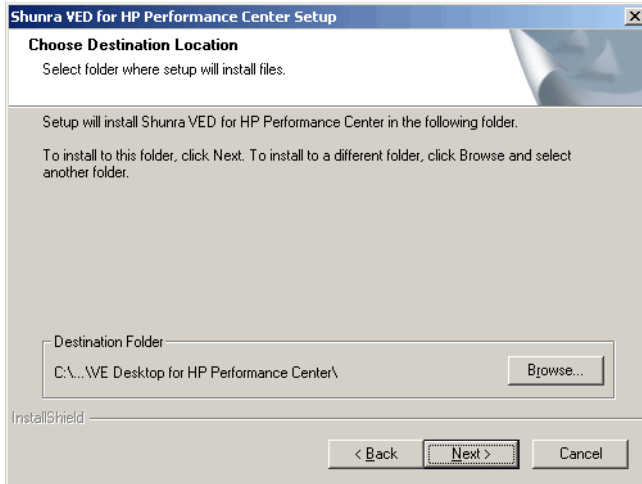


- 2 Click **[Next]**, the License Agreement window appears. Read through the license agreement and click **[Yes]** to accept it and **[Next]** to continue.

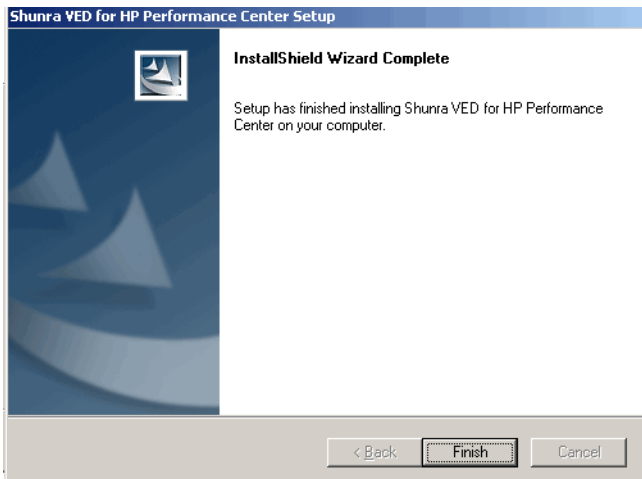
Note: If you decide not to accept the license agreement, click **[No]** to quit the installation process.



- 3 Click **[Next]**, and in the next window specify a destination folder, or accept the default folder, VE Desktop client begins installation.



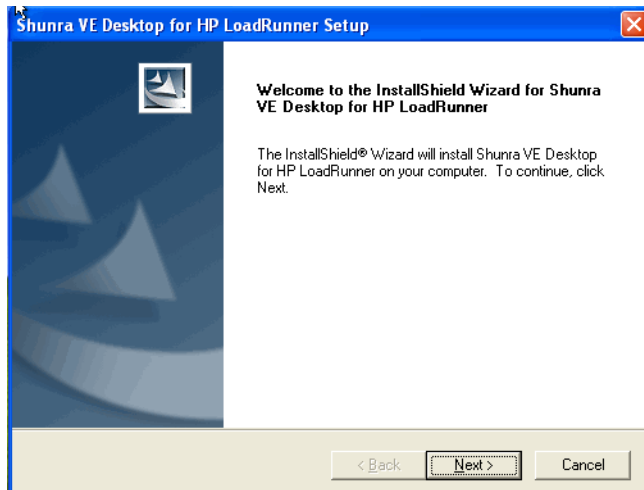
- 4 Click **[Next]**, and in the next window specify a destination folder, or accept the default folder, VE Desktop client begins installation. After the installation has completed, click **[Finish]** to complete the installation process.



Installing the VE Desktop for HP LoadRunner

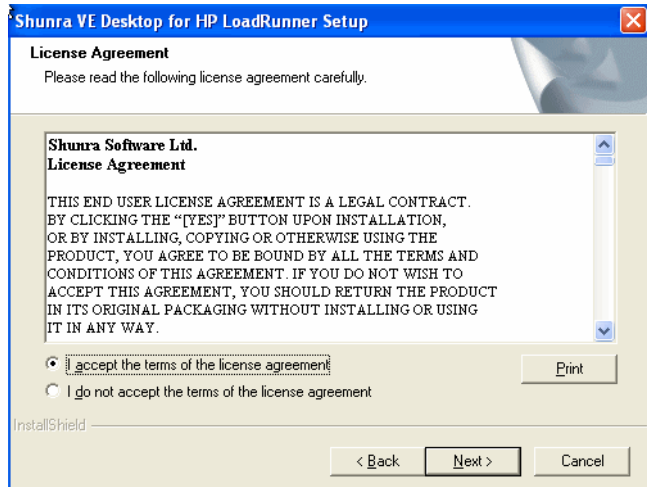
 **To install the VE Desktop for HP LoadRunner:**

- 1 On the HP LoadRunner Controller, open the VED-forHPLR.exe.

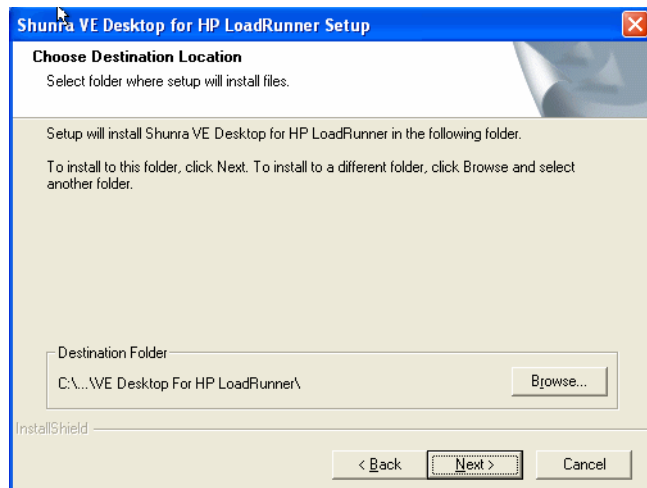


- 2 In the Welcome window, click **[Next]**, the License Agreement window appears. Read through the license agreement and click **[Yes]** to accept it and **[Next]** to continue.

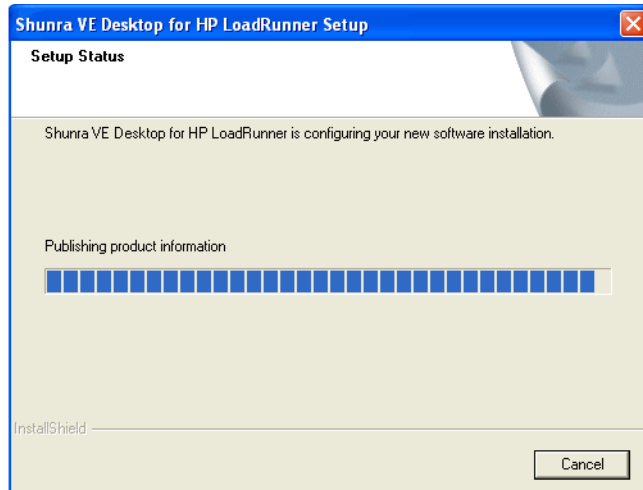
Note: If you decide not to accept the license agreement, click **[No]** to quit the installation process.



- 3 Click **[Next]** and in the next window specify a destination folder, or accept the default folder; the VE Desktop component begins installation.



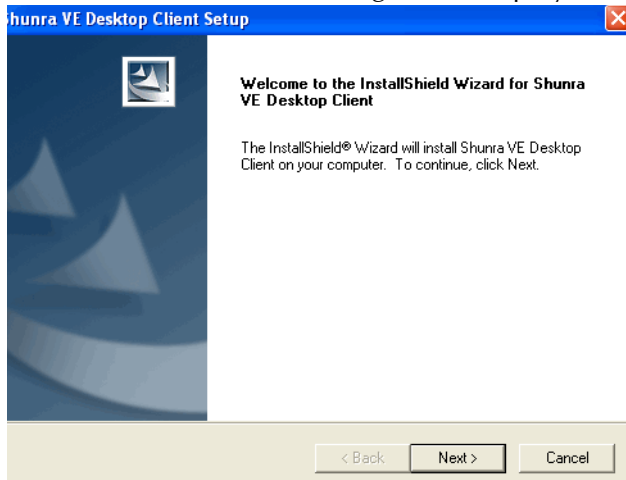
- 4 After the installation has completed, click [**Finish**] to complete the installation process.



Installing the VE Desktop client on the HP Load Generator

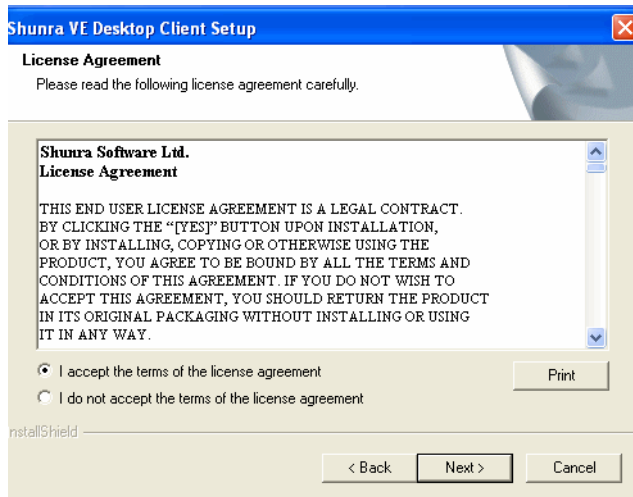
 **To install the VE Desktop client on the HP Load Generator:**

- 1 From the installation CD, , or from the Shunra VE Desktop Setup, open the VEDesktopStandalone.exe. After a few moments the **Welcome** dialog box is displayed.

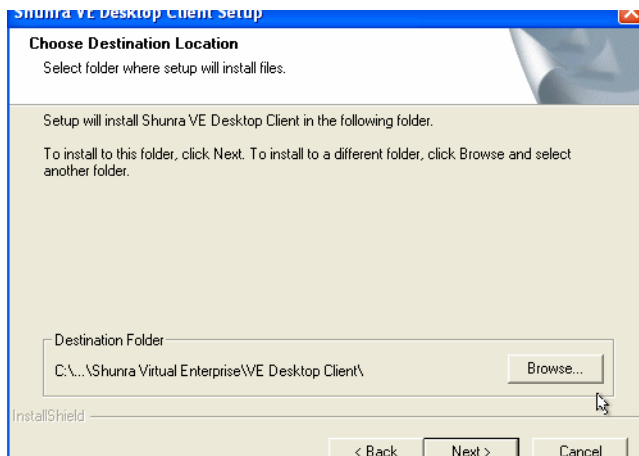


- 2 Click [**Next**]; the **License Agreement** dialog box is displayed. Read through the license agreement and click [**Yes**] to accept it; the **VE Desktop Readme** dialog box is displayed.

Note: If you decide not to accept the license agreement, click [**No**] to quit the installation process.

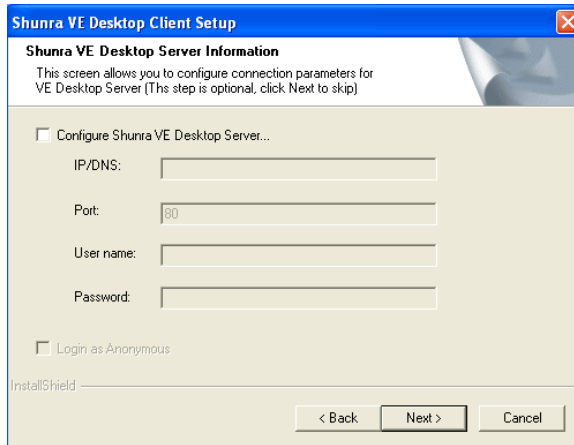


- 3 Click [Next]; then **Choose Destination Location** dialog box is displayed.

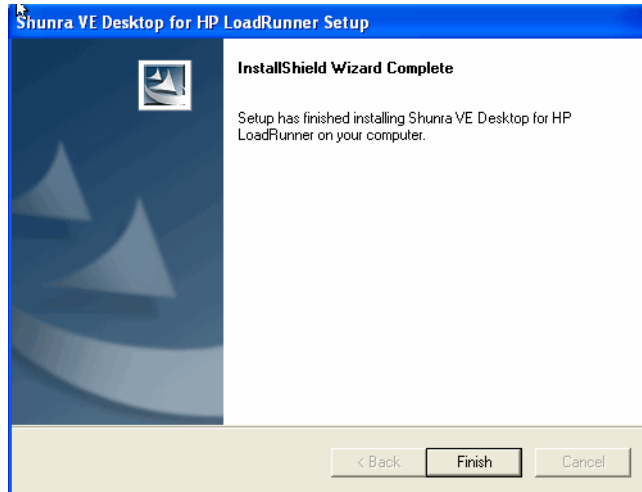


- 4 Specify a destination folder, or accept the default folder. To configure the settings so that the VE Desktop client can communicate with the VE Desktop Server, click the Configure the Shunra VE Desktop Server checkbox and set the parameters. You can skip this step during installation and configure these settings later in the Tools > Options > Login tab. For informa-

tion about the Anonymous user, see **Logging in as an Anonymous User** on [page 4-32](#). VE Desktop client begins installation.



- 5** During installation, various component **installation** messages are displayed.
- 6** After the various components have been installed, you may be asked if you want to configure the firewall.
- 7** Click [**Yes**] to automatically configure the firewall, or click [**No**] so you can manually configure the firewall after installation; the **Installation Complete** dialog box is displayed.



- 8 Click **[Finish]** to complete the installation process.

Note: You must restart your computer for the emulation driver to work correctly.

Installing VE Desktop for MS Visual Studio

The VE Desktop integration with MS Visual Studio can be installed for either MS Visual Studio 2005 or 2008.

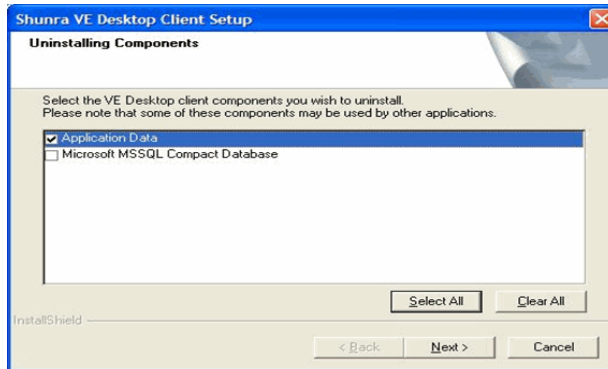
Note: Before you install VE Desktop for MS Visual Studio, the MS Visual Studio application must be closed.

To install VE Desktop for MS Visual Studio:

- 1 On a machine that meets the system requirements, from the installer, in the VED for Visual Studio folder, open the VED-forVS.exe and follow the instructions.
- 2 After restart, open the MS Visual Studio. The WAN Emulation menu is present in the MS Visual Studio main window.

Uninstalling VE Desktop on HP Software

Use the Shunra VE Desktop Setup, which opens from VEDSuiteInstallerForHP.exe to remove any of the VE Desktop components. If you choose to uninstall the database, all your test results will be deleted .



Uninstalling VE Desktop for MS Visual Studio

Note: MS Visual Studio must be closed when removing the VE Desktop component.

To remove the VE Desktop component:

- 1 In the Control Panel of the machine on which VE Desktop is installed, select Add or Remove Programs.
- 2 Select Shunra VE Desktop for MS Visual Studio and click Remove.

Silent Installation

VE Desktop client can be installed (and uninstalled) silently.

Note: When performing silent installation of VE Desktop client, you must be logged in as the Administrator or use a local system account. Verify that all system requirements are met before beginning installation. You will not be notified if silent installation fails.

To silently install VE Desktop client:

- 1** Copy the file VEDesktopStandalone.exe to a convenient location.
- 2** From the Start menu, choose **[Run]**; the Run dialog box opens.
- 3** Type CMD, the Command window opens.
- 4** In the Command window, navigate to the location of the file copied in step 1.
- 5** Enter the file name VEDesktopClient.exe followed by one of the following options:
 - ❖ boot: conducts a silent install and reboots the computer upon completion. Syntax: `VEDesktopClient.exe boot`
 - ❖ noboot: conducts a silent install without rebooting the computer. Syntax: `VEDesktopClient.exe noboot`

Note: Even if the no boot option is used during installation, the PC will need to be rebooted before VE Desktop client is operational.

Silent Uninstall

VE Desktop client can be uninstalled silently.

To silently uninstall VE Desktop client:

- 1** From the Start menu, choose Run; the Run dialog box opens.
- 2** Type CMD, the Command window opens.
- 3** In the Command window, navigate to the location of the file VEDesktopClient.exe.
- 4** Enter the file name VEDesktopClient.exe followed by one of the following options:
 - ❖ **u_boot**: conducts a silent install and reboots the computer upon completion. Syntax: `VEDesktopClient.exe u_boot`
 - ❖ **u_noboot**: conducts a silent install without rebooting the computer upon completion. Syntax: `VEDesktopClient.exe u_noboot`

Note: Even if the no boot option is used during installation, the PC will need to be rebooted before VE Desktop client is completely uninstalled.

-Or-

Click **New user** if you do *not* have a user name and password. In the User name and e-mail address boxes type a user name and your e-mail address. A message appears indicating that your user details were added to the VE Desktop Server database and that an e-mail was sent to you containing your user name and password. In the User name and Password boxes type your user name and password and click **[OK]**.

Note: If your e-mail address does *not* belong to a Trusted domain, you will *not* be able to log in as a new user. To resolve this problem, please contact the VE Desktop Administrator.

VE Desktop client Licenses Overview

The VE Desktop comes with an evaluation license of 10 plays for up to 100 Virtual Users, which also includes the integration for MS Visual Studio. When this license has completed, contact Shunra via the website at www.shunra.com.

VE Desktop can be licensed by:

- ◆ VE Desktop for HP LoadRunner components installed on the HP LoadRunner Controller or Performance Center Host, in which case the license information is available via the [VE Desktop License Manager](#)
- ◆ Node-locked to the machine on which it is running; this license is also this license is also used when integrating with MS Visual Studio
- ◆ VE Desktop Server, refer to the VE Desktop Server User Manual

Server-Based Licensing

If your configuration includes the VE Desktop Server, the server is the license manager for your client, refer to the VE Desktop Server user manual.

Node-Locked VE Desktop client Licenses

If you are using VE Desktop without the VE Desktop Server you will need to install a node-locked license.

Requesting a License Key

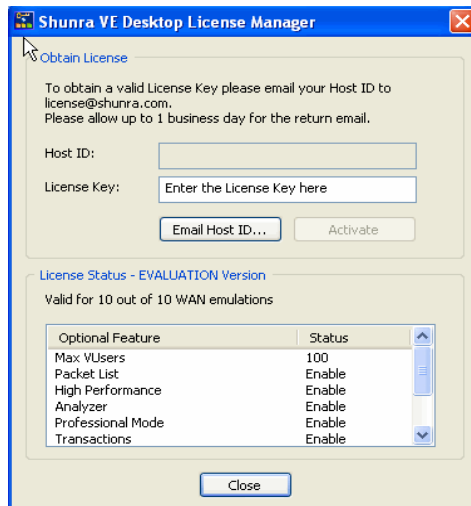
Licensing VE Desktop involves sending to your Host ID, located in the VE Desktop License Manager to Shunra. This Host ID is used by Shunra used to generate a License Key. The License Key

is then sent back to you by e-mail, and you enter it in the VE Desktop Standard License Manager.

Note: It may take up to 24 hours to receive a new License Key.

 To request a License Key:

- 1 From the Start menu, choose **Programs > Shunra Virtual Enterprise > VE Desktop License Manager**; the Shunra VE Desktop License Manager opens.



Optional Feature	Status
Max VUsers	100
Packet List	Enable
High Performance	Enable
Analyzer	Enable
Professional Mode	Enable
Transactions	Enable

- 2 Enter the Serial Number for your copy of VE Desktop; the **[Email Host ID...]** button becomes active.
- 3 Click **[Email Host ID...]**; a pre-formatted and addressed e-mail opens; or you can request a license by sending an email to: license@shunra.com.
- 4 In the body of the email, complete the information as requested, including:
 - ❖ Product Type and version (VE Desktop 5.0 for HP Software)
 - ❖ Product Serial Number
 - ❖ Host ID

❖ Company name

5 Click [Send]. The e-mail is sent.

Installing a New License

Once you have received a License Key for VE Desktop client, you need to install it.

To install a new License:

- 1** Open the **VE Desktop client** License Manager, either from the Start menu, choose **Programs > Shunra Virtual Enterprise > VE Desktop License Manager**; or,
From the System Tray, in the lower right-hand corner of your screen, double-click the VE Desktop client icon; from **Tools** menu choose **License Manager**.
- 2** In the **VE Desktop client License Manager** window, in the License Key text field, enter the License Key provided to you by Shunra; the [Activate] button becomes active.
- 3** Click [Activate]; your product has been activated.
- 4** Click [Close].

3

Conducting WAN Emulation in HP Software and MS Visual Studio

WAN emulation can be incorporated into the HP Software tests by:

- ◆ **Incorporating WAN Emulation in HP Performance Center** ([page 3-1](#))
- ◆ **Incorporating WAN Emulation in HP LoadRunner** ([page 3-2](#))
- ◆ **Including Additional Test Result Statistics** ([page 3-16](#))

WAN emulation can be incorporated into the MS Visual Studio tests by:

- ◆ **Incorporating WAN Emulation in MS Visual Studio** ([page 3-4](#))

Incorporating WAN Emulation in HP Performance Center

 **To conduct a WAN emulation within the HP Performance Center configuration:**

- 1 In the Load Test, Manage page, Workload tab, click the WAN Emulation Settings button.  The WAN Emulation Settings window opens.

- 2 In the WAN Emulation Settings, select each required HP Load Generator and click Enable WAN Emulation. Click the WAN Emulation Settings box and follow the instructions as for the HP LoadRunner, with a few noted exceptions, from **Conducting Tests in Basic Mode**.

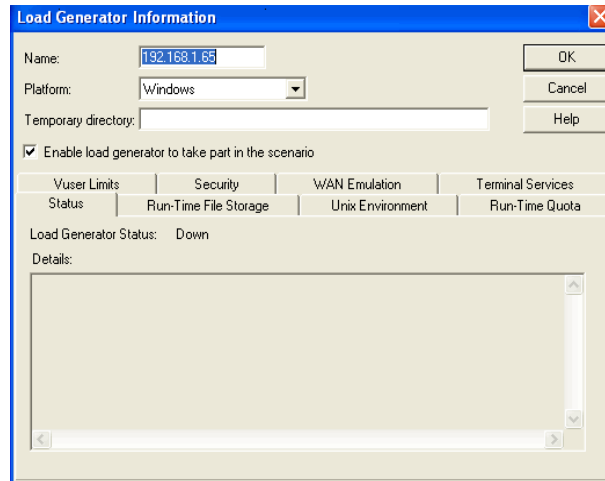
Incorporating WAN Emulation in HP LoadRunner

Within the HP LoadRunner integration you can use one of these modes of operation to define WAN emulation impairments:

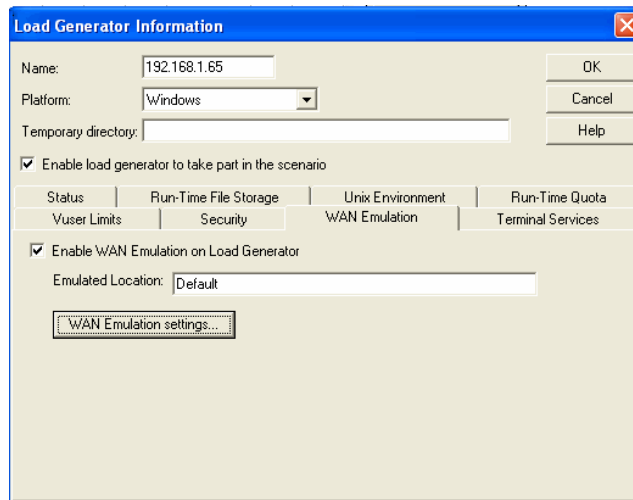
- ✦ **Conducting Tests in Basic Mode:** ([page 3-6](#)) enables selection of a pre-configured source and destination for traffic, connection quality and connection type, thereby configuring VE Desktop client to reflect the desired WAN conditions
- ✦ **Conducting Tests in Advanced Mode:** ([page 3-8](#)) enables entry of specific WAN parameters for packet loss, latency and bandwidth limitation to reflect the desired WAN conditions
- ✦ **Conducting Tests in Professional Mode:** ([page 3-19](#)) enables import of a test plan or network scenario file (.ntx). This file contains gateway configurations, IP filters and WAN configuration settings.

To define WAN emulation parameters:

- 1 In the HP LoadRunner Controller, for each the HP Load Generator, click **Details**. In the Load Generator Information window, select WAN Emulation.



- 2 In the WAN Emulation tab, select "Enable WAN Emulation on Load Generators" and define a new location for the emulation, or retain the default.



- ✦ Click WAN Emulation Settings; the Shunra WAN Emulation Setting window displays in Basic Mode.

Note: To configure the WAN emulation in Advanced mode, skip to **Conducting Tests in Advanced Mode**. Advanced Mode is designed for more technical users and enables the manual configuration of impairments that include latency, bandwidth and packet loss. To conduct tests in Professional mode, which enables importing pre-defined emulation parameters and transactions, see **Conducting Tests in Professional Mode**.

Incorporating WAN Emulation in MS Visual Studio

With the VE Desktop component installed in MS Visual Studio, you can add WAN impairments directly within your projects to determine how the application you are developing will react when deployed over the network. Use these modes of operation to define WAN emulation impairments:

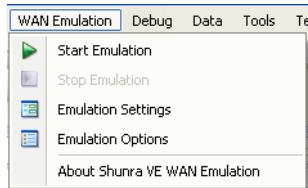
- ✦ **Conducting Tests in Basic Mode:** ([page 3-6](#)) enables selection of a pre-configured source and destination for traffic, connection quality and connection type, thereby configuring VE Desktop to reflect the desired WAN conditions
- ✦ **Conducting Tests in Advanced Mode:** ([page 3-8](#)) enables entry of specific WAN parameters for packet loss, latency and bandwidth limitation to reflect the desired WAN conditions

To display the WAN Emulation toolbar:

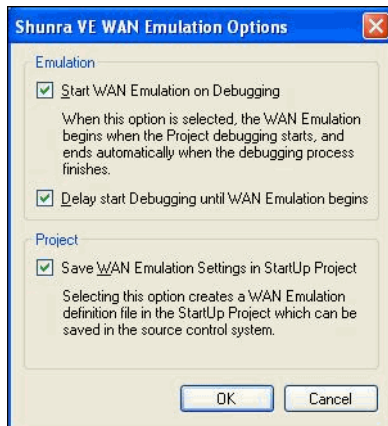
- 1 As in standard MS Windows applications, click the Add or Remove Buttons at the right side of the toolbar(s), and select or deselect **WAN Emulation**; or from Add or Remove Buttons, click **Customize** and in the Toolbars tab, select or deselect **WAN Emulation**.

To define WAN emulation parameters:

When you open a MS Visual Studio project with the VE Desktop integration installed, the WAN Emulation dropdown menu is located in the main window.



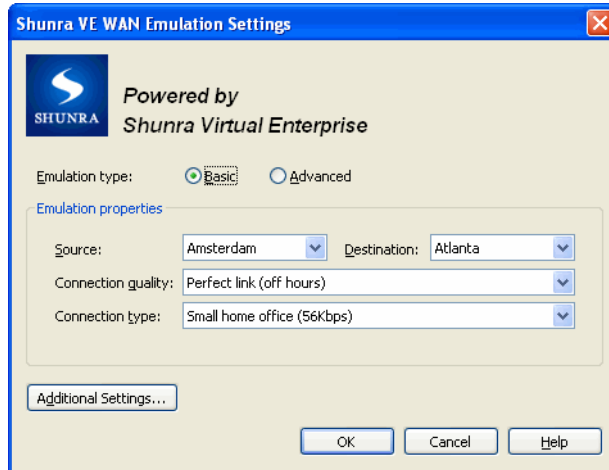
- 1 Select **Start Emulation** to enable WAN emulation for a specific project. For WAN emulation to always start automatically when debugging starts, select **WAN Emulation > Emulation Options** (menu option or button). In the **Shunra VE WAN Emulation Options** dialog box, select **Start WAN Emulation on Debugging**.



- 2 To delay the debugging process until the WAN Emulation is running, select **Delay start Debugging until WAN Emulation begins**.
- 3 To automatically save the parameters of the WAN emulation to the StartUp Project, select **WAN Emulation** (menu option or button) > **Emulation Options**. In the **Shunra VE WAN Emu-**

lation Options dialog box, select **Save WAN Emulation Settings in StartUp Project**.

- 4 To configure specific parameters for WAN emulation, select **Emulation Settings**; the Shunra VE WAN Emulation Settings dialog box displays in Basic Mode.



Note: To configure the WAN emulation in Advanced mode, skip to **Conducting Tests in Advanced Mode**. Advanced Mode is designed for more technical users and enables the manual configuration of impairments that include latency, bandwidth and packet loss.

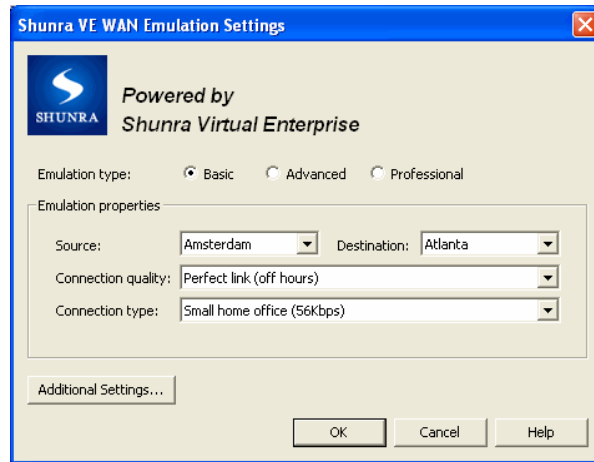
- 5 To end the WAN emulation, select **Stop Emulation**. WAN emulation stops automatically when debugging finishes if **Start WAN Emulation on Debugging** is selected.

Conducting Tests in Basic Mode

Basic Mode provides an easy to use interface that enables configuration of a source and destination location, connection quality and connection type.

Test parameters are configured, a test is begun, transactions are conducted to see how they react in real-time, and reports regarding a test can be viewed.

- 1 To define more advanced options, see **Configuring Additional Settings** (page 3-14).



Note: As a starting point for tests, basic latency for each emulated location can provide substantial value, even if packet loss and bandwidth constraints are unknown. This can be obtained from the predefined locations listed in the Basic Mode, or from a file obtained in the VE Network Catcher.

You can also use a simple ping to or from the remote location to the server location. The approximate latency to enter will be half of the total round trip time as measured by the ping.

- 2 Configure the following:
 - ❖ **Source and Destination:** select a source and destination that represent the two points between which you want to emulate WAN conditions.
 - ❖ **Connection Quality:** select the time of day that best reflects what conditions you want to test. Options include:
 - Perfect Link (off hours)

- Normal business hours
- Busy business hours
- Worst case
- Wireless
- Cellular
- Satellite
- DS3/T3

❖ **Connection Type:** select the type of connection you want to emulate. Options include:

- Small home office (56Kbps)
- DSL home office (256 Kbps)
- Cable modem home office (384 Kbps)
- Small branch office - (512 Kbps)
- Medium branch office - (768 Kbps)
- Regional branch office - (T1-1544 Kbps)
- Large branch office (3 Mbps)
- Unrestricted (no bandwidth limitation)

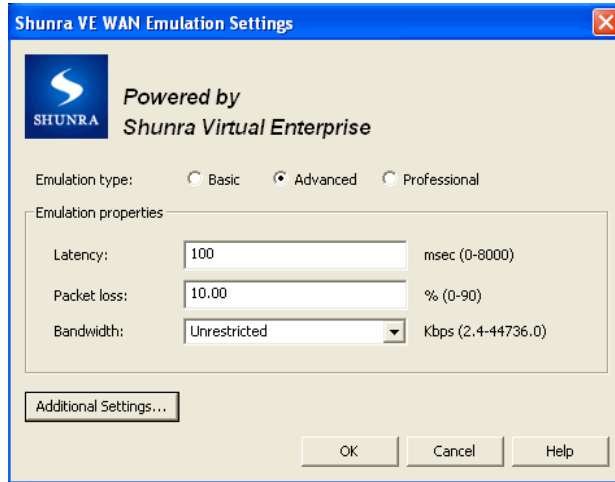
The following values reflect the usual latency for the following conditions:

Wireless	Packet loss = 0.47%
Cellular	Packet loss = 5.12
Satellite	Packet loss = 6.32%

Note: For instructions on how to exclude specific IP addressees that may interfere with the WAN emulation results, see **Configuring IP Filters** on [page 3-17](#).

Conducting Tests in Advanced Mode

- 1 To run the test in Advanced mode, in the Shunra WAN Emulation Setting window, select **Advanced**.



2 Configure the following:

- ❖ **Latency:** defines the time it takes an IP packet to cross the WAN. The valid range is 0 — 1000 msec.
- ❖ **Packet Loss:** defines the chance of losing IP packets while data travels though the WAN. The valid range is 0 — 10%
- ❖ **Bandwidth Limitation:** for the HP Load Generator to limit the aggregate bandwidth to the capacity of a specific WAN link, select a bandwidth from the Bandwidth Limitation drop-down list, or type a range from 2.4 — 44736 Kbps. You can run a test without Bandwidth limitation by selecting Unrestricted; however the overall throughput may be higher than 45 Mbps, since Latency is the only impairment that is emulated.

Conducting Tests in Professional Mode

VE Desktop integrates a number of file types that can be imported and used to conduct tests and time transactions. For the integration with HP Software, using Professional mode you can import:

- ◆ **Network Scenario Files (.ntx)** The .ntx file contains the emulation parameters ([page 3-19](#))
- ◆ **Test Plans (.vet)** Test Plans include pre-defined emulation parameters and a list of transactions that need to be conducted ([page 3-19](#)).

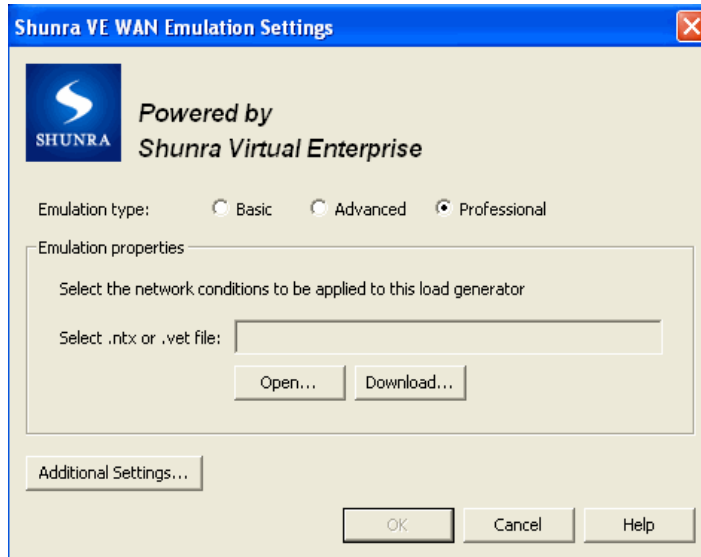
You cannot import a Network Scenario File that includes more than 10 Excluded IP addresses. For more information, see **Configuring IP Filters** on [page 3-17](#).

The file name in basic and advanced mode contains the preface of Basic or Advanced as relevant. The version of the file used for the test is the latest version, which may have been updated since the file was configured. If your configuration includes the VE Desktop server you can also download **Business Processes (.ves)** files, see [page 3-20](#).

The Samples directory also contains the TwoNetworks.vet file with network information but no transactions, that you can use to customize to your requirements.

To conduct a test in Professional Mode:

- 1 In the Shunra WAN Emulation Setting window, click the **Professional** option; professional options are displayed.

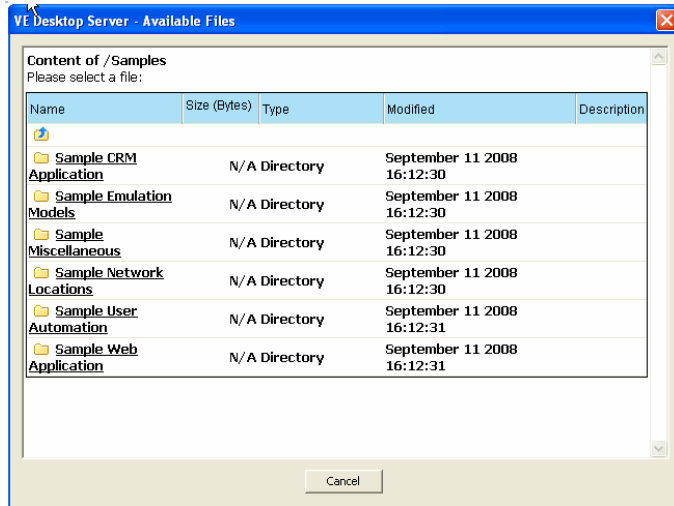


- 2 Click **[Open]** to import an .ntx or .vet file from a shared network drive and select the file from the Browser window. This file may have been created on the Network Editor or from the VE Network Catcher.

Or

Click **[Download]** to obtain files from the VE Desktop server. In the VE Desktop server window, select the required .vet and/or .ntx files.

Note: In the HP Performance Center, to enable importing of files, click Additional Settings. In the Settings tab, select "Professional mode includes opening files located on a shared drive" and click OK. In the Professional mode, the Open button appears.



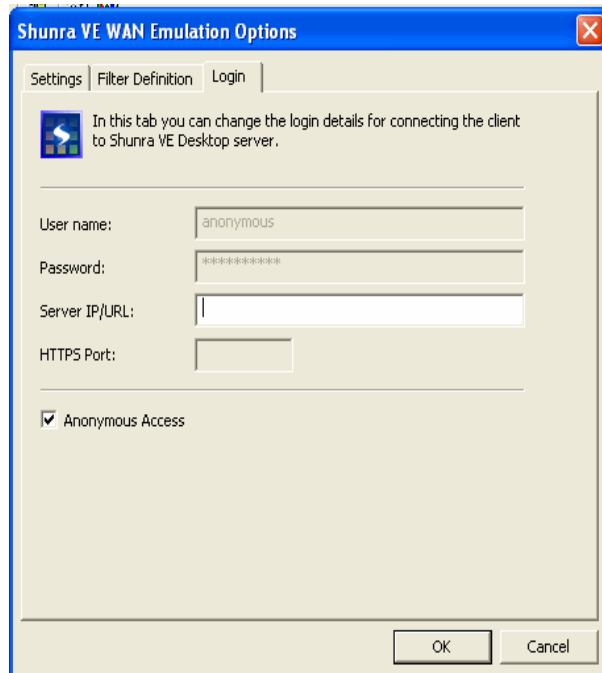
Configuring the VE Desktop server

If you have configured the VE Desktop server during installation, these settings are already configured.

To configure the VE Desktop server

- 1 In the Shunra WAN Emulation Setting window click "Additional Settings". The Shunra WAN Emulation Option window displays.

Note: These settings must also be defined in VE Desktop server.



- 2 Select the Login tab and enter the User Name, Password and IP address or URL of the server. For an explanation of the files types, see **File Types**.
- 3 To enable login using a generic set of pre-defined credentials, select Anonymous User.

As opposed to registered users (users with a personal user-name and password), Anonymous users cannot log directly into the VE Desktop server through web management and can only upload and download files from the client to predefined locations (by default Samples and Public folders). These settings are pre-configured and are activated by selecting the Anonymous User checkbox.

- 4 Click [OK].
- 5 In the Shunra WAN Emulation Setting window click [OK].

Configuring Additional Settings

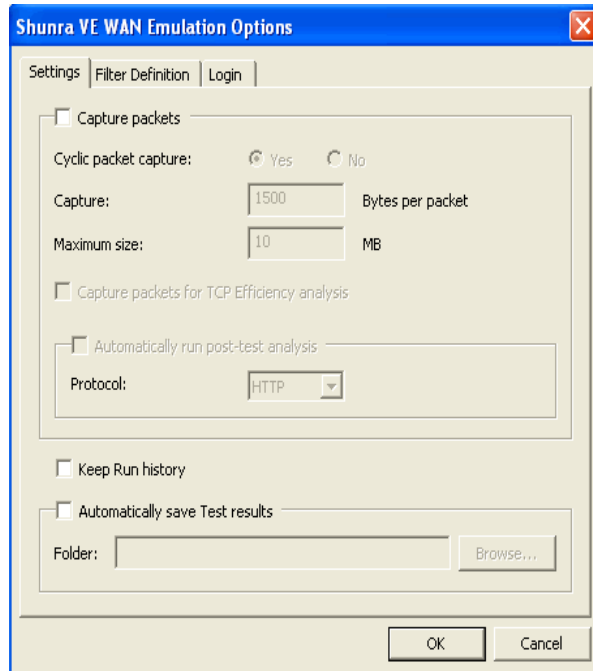
In Basic, Advanced or Professional mode, you can also define the following:

- ✦ **Capturing Packets** (on [page 3-14](#))
- ✦ **Including Additional Test Result Statistics** (on [page 3-16](#))
- ✦ **Configuring IP Filters** (on [page 3-17](#))

Capturing Packets

Use of packet capture is only relevant on HP Load Generators running a single user script. Packet capture is not appropriate on HP Load Generators that are running multiple VUsers and generating large volumes of traffic. You can configure an internal packet capture capability that can then be used to analyze traffic to determine the detailed causes of application performance results. The analysis includes an assessment of the "chattiness" of the application, as measured by application turns, and breaks out Client, Network, and Server components of response time.

- 1** To enable packet capture capability, in the Shunra WAN Emulation Setting window click "Additional Settings". The Shunra WAN Emulation Option window appears.



2 In the Settings tab, configure the following:

Capture packets:

- ❖ **Cyclic packet capture:** if you select Yes, once a Packet List has reached its maximum it will begin to overwrite the Packet List file from the beginning, otherwise it will drop all new packets once capture is full
- ❖ **Capture:** determines the maximum bytes per packet to be captured
- ❖ **Maximum size:** determines the maximum size of the overall capture file (maximum 100 MB)
- ❖ **Capture packets for TCP Efficiency analysis:** when selected, adds a second packet list to the test enabling VE Desktop to report the number of packets in flight

Note: Selecting this option adds a second packet list and subsequently divides the available number of packets for capture between the client-side packet list and server-side packet list.

Including Additional Test Result Statistics

For tests that include a packet list, you can automatically trigger a post-test analysis to occur on each HP Load Generator. It may take some time for the analysis to run, however this process should not interfere with HP LoadRunner's post-test cleanup processing.

Note: For the analysis, the VE Analyzer must be installed and licensed on the HP Load Generator. By default the VE Analyzer in the integration with HP Software provides 24 hours of analysis after the WAN emulation has completed.

For users of the VE Desktop for MS Visual Studio, a VE Desktop Professional license is required, contact Shunra sales.

- 1** To enable auto, in the Shunra WAN Emulation Setting window click "Additional Settings"; the Shunra WAN Emulation Option window appears.
- 2** Select "Automatically run post test analysis" and select the protocol you are using for data transfer (HTTP, HTTPS, UDP or TCP)
- 3** In addition, to mark specific transactions, use the VE Desktop API to "wrap" the `lr_start_transaction` and `lr_end_transaction` API calls, refer to **VE Desktop API**. by changing your scripts to refer to an API named `lr_start_transaction_extended` and `lr_stop_transaction_extended`. Then include the routine provided in the Appendix **Single User Script Modification**.

Saving Results

- 1** To save the results of the emulation and Run history, select Keep Run History.
- 2** To save the WAN emulation results to a file, select "Automatically save test results" and then choose a folder in which to save the file.

Note: When the VE Analyzer is installed on the same machine as the VE Desktop client, and "Keep Run history" is not selected, the previous Run is overwritten before a new Run begins. Upon exiting the VE Desktop client or a machine restart, the last Run is saved.

Configuring IP Filters

You can set the VE Desktop client to affect only network traffic to and from specified IP addresses (to "include" them) or to refrain from affecting traffic to certain addresses ("exclude" them).

For example the HP LoadRunner Controller's IP address is excluded by default, therefore, the traffic between the HP LoadRunner Controller and HP Load Generators does not affect the WAN emulation test results.

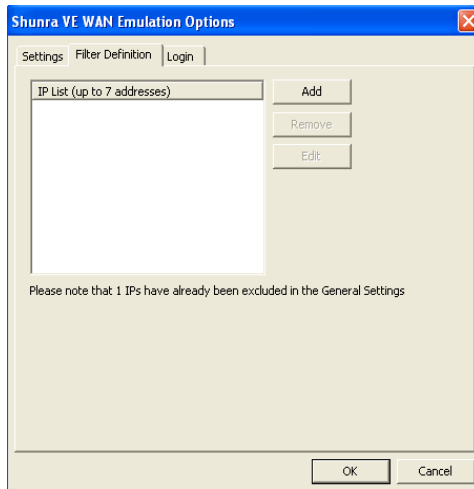
Network traffic that is not affected by the emulation will be forwarded as soon as possible, without experiencing any WAN effects. For example, you can test a web application while email continues to operate normally.

Note: If your configuration includes a VE Desktop server, the management traffic between the VE Desktop client and the VE Desktop server is automatically excluded from the emulation. If you configure a Proxy server, you need to add its IP address to the Exclude list so that it will not be included in the emulation.

 To include or exclude IP addresses:

- 1 In the Shunra WAN Emulation Options, click **Filter Definition**; the **Filter Definition** tab is displayed. Up to 10 IP addresses can be selected for each Policy either to include or exclude, by default the address of the HP LoadRunner Controller is excluded.

Note: IP addresses can also be excluded from the HP LoadRunner Controller, in Tools > Options > WAN Emulation.



- 2 Click **[Add]**; the Enter IP Address dialog box is displayed.
- 3 Enter the IP address to be **included** or **excluded**, and click **[OK]**.
- 4 Repeat steps 2 to 3 for all the IP addresses you wish to include or exclude.

- 5 Select **[OK]** to save the settings.

Note: An IP address can be removed from the list by selecting it and clicking **[Remove]**.

File Types

VE Desktop use the following files for conducting emulation:

- ✦ **Network Scenario Files (.ntx)**
- ✦ **Test Plans (.vet)**
- ✦ **Business Processes (.ves)**

Network Scenario Files (.ntx)

Network Scenario files contain emulation parameters including latency, bandwidth and packet loss. These files may originate in the VE Network Catcher and can be modified in the VE Network Editor, or created entirely in the VE Network Editor, refer to the Network Editor User Manual.

Once imported, the file automatically configures VE Desktop client component with the required emulation parameters which, when played, are applied to all traffic entering and leaving the local NIC. Network Scenario Files can be either a part of a Test Plan, or a stand alone file.

Test Plans (.vet)

Test Plans are files that contain instructions which are used to guide WAN emulations. They consist of:

- ✦ **Business Processes (.ves):** contains a list of transactions that reflect a logical process conducted by end-users that need to be tested

- ◆ **Timing Transactions:** contains emulation parameters which define the network conditions of the emulated WAN in which transactions will be tested

Test Plans are stored on the VE Desktop server and downloaded into the VE Desktop client component.

A Test Plan is a group of pre-defined test parameters that guides users in conducting tests. Test Plans consist of network conditions (defined in NetworX files - .ntx) including Latency, Bandwidth and Packet Loss; a list transactions (defined in Business Process files - .ves) that need to be conducted, and the ability to add a packet list which enables the analysis of transactions.

Test Plan (.vet) = Network Scenario (.ntx) + Business Process (.ves)

Once created on the VE Desktop server, Test Plans are downloaded to the VE Desktop client component and conducted. Finally, test results including transaction response times, reports, and packet lists can then be uploaded to the VE Desktop server central results repository.

Business Processes (.ves)

A Business Process is a list of transactions that represent a set of actions being emulated for testing. They are used to test the amount of time it takes to conduct transactions, refer to the VE Desktop server User Manual.

4

Monitoring Performance

Within the HP LoadRunner, you can view statistics in the HP LoadRunner Analysis application. In addition, the VE Desktop client component on the HP Load Generator includes certain reports. Additional analysis can be conducted in the VE Analyzer which is included in the VE Desktop license for up to 24 hours after the WAN emulation play has completed.

Note: The VE Reporter/VE Analyzer must be installed on the HP Load Generator to enable automated post-test analysis.

The PerfMon Counters

The PerfMon Counters must be added per HP Load Generator on which WAN emulation is performed.

 To setup the WAN emulation monitors:

- 1 On the HP Load Generator, in the Windows Resources, click [Add].
- 2 In the Add Counters window, right-click Add Measurements.

The counters listed below are divided according to the following categories:

Locations (NICs)

Object Name	Object Description
Shunra VE Server	Server Location measurements of the Shunra VE emulation
Shunra VE Client	Client Location measurements of the Shunra VE emulation
Counter Name	Counter Description
Total Throughput In (bytes)	Total throughput sent by this location.
Total Throughput Out (bytes)	Total throughput received by this location.
Current bps In	Number of bits sent by this location during the last second
Current bps Out	Number of bits received by this location during the last second
Bandwidth Util In %	Percentage of outgoing bandwidth used by this location during the last second
Bandwidth Util Out %	Percentage of incoming bandwidth used by this location during the last second

Cloud

Object Name	Object Description
-------------	--------------------

Shunra VE Server	Server Location measurements of the Shunra VE emulation
Shunra VE Client	Client Location measurements of the Shunra VE emulation
Counter Name	Counter Description
Total Throughput In (bytes)	Total throughput sent by this location.
Total Throughput Out (bytes)	Total throughput received by this location.
Current bps In	Number of bits sent by this location during the last second
Current bps Out	Number of bits received by this location during the last second
Bandwidth Util In %	Percentage of outgoing bandwidth used by this location during the last second
Bandwidth Util Out %	Percentage of incoming bandwidth used by this location during the last second

Cloud

Object Name	Object Description
Shunra VE Cloud	Wide area network measurements of the Shunra VE emulation

Counter Name	Counter Description
Minimum Latency (msec)	Minimum latency in the Network during the last second
Average Latency (msec)	Average latency in the Network during the last second
Maximum Latency (msec)	Maximum latency in the Network during the last second
Packet Loss Count	Packet loss count in the Network during the last second
Packet Loss %	Packet Loss percentage in the Network during the last second
Packet Loss Total	Packet Loss during the entire test

Viewing Statistics in the HP LoadRunner Analysis

After the Perfmon counters have been added, the following statistics can be monitored in the Windows Resources of the HP LoadRunner Controller and/or HP Performance Center, also according to Emulated Locations:

- ◆ Bandwidth utilization (in/out)
- ◆ Current BPS (in/out)
- ◆ Latency
- ◆ Total Throughput in Bytes (in/out)

Viewing Statistics in the VE Reporter

The Shunra VE Reporter provides extended reporting features, such as viewing both run-time and offline data. VE Reporter

requires a separate license and provides the following information:

- ◆ View runtime and offline data about the current test
- ◆ Generate reports and upload data to the VE Desktop server:
 - ❖ Generate VE Desktop reports in MS Word or HTML
 - ❖ Upload data to the VE Desktop server
 - ❖ Export raw data to VED format

After a test is complete, you can configure each load generator to automatically post the test results to a central file share or to the VE Desktop Server. VE Reporter can be used to open a test results file (.ved) created during a test. VE Reporter enables you to generate additional reports.

Reports can be generated in both Microsoft Word and HTML format, while raw data can be exported to VE Desktop format, which can then be used for a number of purposes, for example to create customized charts in applications such as Microsoft Excel. For further information, refer to the VE Reporter User Manual.

5

VE Desktop API

This section provides information about using the API to automate tests. For users of HP Software such as HP LoadRunner or Performance Center with versions from 9.5, the VE Desktop API can be used to incorporate WAN emulation in addition to the user interface integration. For users of versions previous to 9.5, the VE Desktop API is the only method available for incorporating WAN emulation. For specific information about marking transactions, refer to the VE Desktop for HP Software user manual.

Note: For users of the MS Visual Studio, a VE Desktop Professional license is required, contact sales@shunra.com.

The VE Desktop API is enabled with a set of pre-defined commands as described in the following sections:

- ◆ **Guidelines for Working with the API** ([page 5-1](#))
- ◆ **Working with the External DLL Interface** ([page 5-4](#))
- ◆ **Working with the COM Interface** ([page 5-6](#))
- ◆ **Working with the CLI** ([page 5-7](#))

Guidelines for Working with the API

This section provides guidelines and information to assist in automating VE Desktop tests, including:

- ◆ **General Guidelines and Limitations:** ([page 5-2](#))
- ◆ **Notes about VE Analyzer Reports:** ([page 5-2](#))

- ✦ **Transaction Markers vs. Time Stamp:** ([page 5-3](#))

General Guidelines and Limitations

The following is a list of guidelines to be followed when automating VE Desktop tests:

- ✦ Only one testing script can work with the API at any given moment, for the duration of the application test
- ✦ Transaction information listed in the transaction pane of the VE Desktop may not update during a test, particularly when using dynamic transactions
- ✦ VE Desktop does not support nested calls (transactions or virtual users). This means that only a single transaction can be conducted per VE Desktop at a time (the current transaction needs to stop before the next transaction can start)
- ✦ Transaction Response Times (TRT) reported in VE Desktop and displayed in VE reports reflect an overhead of hundred or so milliseconds. The exact amount depends on machine resources. This overhead does not impact analysis

Note: A filter should be defined in all participating VE Desktop clients to exclude traffic to the VE Desktop Server or HP Software controller host IP. This will prevent administration traffic from undergoing emulation.

Notes about VE Analyzer Reports

When analyzing tests results displayed in reports generated from VE Desktop data, it is important to keep in mind the following:

- ✦ **Transaction Data Report:** TRTs are set by the start and stop commands
- ✦ **Transaction Marker Report:** Transaction Markers are set by the start and stop commands
- ✦ **Application Efficiency & Infrastructure Breakdown Reports:** data displayed in reports is measured from the moment the

first packet is sent out from the client until the last packet is received by the client

Note: User down time that takes place between the first and last packets is calculated as part of the TRT (for example, if a user answers a phone call while a test is running and is distracted, that time would appear as part of the TRT in resulting reports).

For further information, see the VE Reporter user manual.

Transaction Markers vs. Time Stamp

There are two mechanisms in VE Desktop that can have an impact on results reported in VE Applications. One is the actual timer that is activated when you use Start and Stop transactions, and which is registered as the Transaction Response Time (TRT) reported in the VE Desktop Transaction Summary report. The second is the mechanism that sets a marker in the packet list, which demarcates the beginning of an action. This marker tells VE Analyzer that the next packet in the capture is the first packet of the transaction to be analyzed.

Technically, a transaction marker is set, the Transaction Timer is Started, the Transaction Timer is Ended, and then the closing transaction marker is set. This may result in a small discrepancy between the VE Desktop Transaction Summary report, and information displayed in load testing tools such as HP LoadRunner, and is a result of VE Desktop overhead. However, because analysis is based on the first and last packet, analysis is always accurate.

Working with the External DLL Interface

The VE Desktop has been developed with an external DLL interface that uses the ShunraVEController.dll to synchronize commands between external tools and VE Desktop client. Load test scripts using the VE Desktop API also reference the ShunraVEController.dll which is used to issue commands to the VE Desktop client running on each HP Load Generator. These commands are embedded in the automation script.

Note: Commands are case sensitive. When working with load tools, Start transaction and Stop transaction commands need to be wrapped in VE Desktop Start transaction and Stop transaction commands.

- ✦ **Process Overview:** ([page 5-4](#))
- ✦ **Automation Script Requirements:** ([page 5-5](#))
- ✦ **Supporting Files:** ([page 5-5](#))

Process Overview

The process of integrating VE Desktop emulations into automation tools using the External DLL Interface includes:

- ✦ Configuring a filter on the load tool controller host IP to prevent controller traffic from undergoing emulation
- ✦ Modifying code to integrate VE Desktop External DLL Interface commands
- ✦ Running a script or code that manages VED Client licenses, changes VED Client settings; starts/stops VEDesktop Client; conducts transactions, fetches runtime statistics, and saves or uploads results

Automation Script Requirements

When integrating External .dll Interface commands with automation scripts, you need to insert calls to the **ShunraVEController.dll**. This dll prevents processing a command until the previous VE Desktop client command has completed, ensuring that transactions are synchronized between the caller and VE Desktop. In HP LoadRunner, for example, you can load this .dll to the script by adding the following: `lr_load_dll( "ShunraVEController.dll" );`

Note: For further information regarding the .dll see the file `ShunraVEControllerExports.h` located in the Shunra Virtual Enterprise\VE Desktop client\Samples folder which is located in the VE Desktop installation folder.

Supporting Files

A number of files are installed with VE Desktop that support the operation of the External DLL Interface and illustrate its use, these include:

- ◆ **ShunraVEController.dll:** .dll file that provides External DLL Interface functions
- ◆ **Sample HP LoadRunner script:** a sample HP LoadRunner script that utilizes Automation API has been included with the application. The sample can be found in the **Shunra Virtual Enterprise\VE Desktop Client \Samples\HP LoadRunner Integration** folder which is located in the VE Desktop installation folder.
- ◆ **ShunraVEControllerExports.h:** list of exported functions from the DLL located in the Shunra Virtual Enterprise\VE Desktop client\Samples folder which is located in the VE Desktop installation folder

Working with the COM Interface

Integrating WAN emulation into automation tests using the COM interface includes:

- ✦ From your code create the "ShunraVEAPI.ShunraVEDController" object. Use this object to change VED client settings, start/stop tests and save/upload results.
- ✦ From your code create the "ShunraVEAPI.ShunraVEStatistics" object. Use this object to poll VED Client runtime statistics.
- ✦ From your code create the "VEDLicenseMgrLib.VedLmAutomation" object. Use this object to manage the VED Client license
- ✦ Configure a filter on the HP LoadRunner Controller host IP to prevent important traffic from undergoing emulation
- ✦ Modify your code to call the ShunraVEDController interface functions
- ✦ Run the code that changes VE Desktop client settings, starts VE Desktop client, conducts transactions, polls statistics; uploads results and manages the license

Note: When working with load tools, Start and Stop transaction commands need to be wrapped in VE Desktop Start and Stop transaction commands

VE Desktop COM Interface Definitions

The type libraries for the VED COM API are installed by default in the following folders:

- ✦ ShunraVEDController.tlb is found in:
C:\Program Files\Shunra Virtual Enterprise\VE Desktop Client\bin

- ◆ ShunraVEAPI.tlb is found in:
C:\Program Files\Shunra Virtual Enterprise\VE Reporter\Bin
- ◆ VEDLicenseMgr.tlb is found in
C:\Program Files\Common Files\Shunra Virtual Enterprise\VE Desktop

Working with the CLI

The directory in which the VEDCLI command is located has been added to the Path Environment Variable. This enables you to type commands without having to accompany them with the path in the CLI.

Note: Commands are case sensitive. When working with load tools, Start and Stop commands need to be wrapped in VE Desktop Start and Stop commands

To control the VE Desktop client from the CLI:

- 1 From the Start menu, choose **Run**.
- 2 Type CMD, then click **[OK]**; a Command window opens.
- 3 Enter the required commands.

Note: All CLI commands and parameters noted in square brackets [] are optional.

API Commands and Parameters List

Details about the following API commands and parameters are provided in **API Commands and Parameters Details**:

- ◆ **StartAdvanced**: starts a test in Advanced mode
- ◆ **StartProfessional**: starts a test in Professional mode
- ◆ **StopTest**: stops the test

- ✦ **StartTransaction**: starts the next transaction in the test plan, or adds a transaction to a test plan and starts a dynamic transaction
- ✦ **StartNextTransaction**: starts the next transaction in the test plan
- ✦ **NextNetwork** : loads the next NetworX file
- ✦ **RestartBusinessProcess**: restarts the current Business in the Test Plan
- ✦ **NextBusinessProcess**: loads the next Business Process in the Test Plan
- ✦ **StopTransaction**: stops current transaction
- ✦ **SaveRun**: saves the test run
- ✦ **UploadRun**: uploads test run to VE Desktop server
- ✦ **SetLogFile**: Activates log file listing all executed commands and errors
- ✦ **ActivatePacketList**: adds a packet list to a test
- ✦ **CancelPacketList**: removes a packet list from a test
- ✦ **LicenseCheckout**: checks a VE Desktop client license out for use
- ✦ **LicenseCheckin** : checks a VE Desktop client back into the VE Desktop Server
- ✦ **GetRunningTestDescription**: Gets description of the test that is currently running
- ✦ **GetErrorString**: Gets error description for a given error code
- ✦ **SetAutoAnalysisParams**: Sets parameters for automatically conducting post-test analysis
- ✦ **SetKeepRunHistory**: Sets whether to keep the runs history (override or not the previous run data in the database)
- ✦ **GetKeepRunHistory**: Obtains setting of whether the run history is kept or not kept
- ✦ **SetIPFilterParameters**: Sets IP Filter parameters

- ◆ **GetIPFilterParameters:** Gets IP Filter parameters
- ◆ **GetIPFilterItem:** Obtains IP from IP Filter list by index
- ◆ **GetIPFilterCount:** Obtains the number of IPs in the IP Filter list
- ◆ **AddIPFilterItem:** Adds IP address to the IP Filter list
- ◆ **RemoveIPFilterItem:** Removes IP address from the IP Filter list
- ◆ **ClearIPFilter:** Clears the IP Filter list
- ◆ **SetPacketListParameters:** Sets the packet capture parameters
- ◆ **GetPacketListParameters:** Gets the packet capture parameters
- ◆ **SetAutoSaveTestResults:** Sets parameters for automatically saving test results at the end of test
- ◆ **SetResultFolderPrefix:** Sets prefix to be added to the result folder name when saving or uploading test results
- ◆ **GetAutoSaveTestResults:** Gets parameters of automatically saving test results at the end of test
- ◆ **SetCalculateTopTalkers:** Sets the top talker calculation parameters
- ◆ **GetCalculateTopTalkers:** Gets the top talker calculation parameters
- ◆ **VEDStats_GetCurrentStatistics:** Gets the latest statistic polling error code and description
- ◆ **VEDStats_GetErrorString:** Gets statistic polling error description for given error code
- ◆ **VEDStats_GetLastError:** Gets the latest statistic polling error code and description
- ◆ **GetHostID:** Gets the VED Client's Host ID required for creating the license.
- ◆ **SetLicenseKey:** Sets the VED Client's license key.

API Commands and Parameters Details

This section reviews commands, syntax, and various parameters that are available for each command when embedding commands in automation scripts and using the External DLL Interface, using the command through the COM interface or through the command line interface (CLI). Calls to the .dll follow C/C++ coding standards.

StartAdvanced

Starts a test in Advanced mode.

Syntax

External DLL

```
int VED_StartAdvanced(int nLatency, int nPacketLoss,  
int nBandWidth=0);
```

CLI

```
VEDCLI -StartStd latency pl bw
```

Parameters

- ◆ latency: configures latency in msec
- ◆ pl: configures packet loss in percentage (%)
- ◆ bw: configures bandwidth in kbps where unrestricted value is 0 or "unrestricted" string value

COM

Interface: ShunraVEDController

```
HRESULT StartAdvanced([in] LONG nLatency, [in]  
DOUBLE fPacketLoss, [in] DOUBLE fBandWidth,  
[out,retval] LONG* pResult);
```

Parameters

- ◆ Latency: configures latency in msec
- ◆ PacketLoss: configures packet loss in percentage (%)
- ◆ BandWdith: configures bandwidth in kbps where unrestricted value is 0 or unrestricted

Examples

External DLL

```
VED_StartAdvanced(50,4,0);
```

CLI

```
VEDCLI -StartStd 50 4 0
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEController")  
  
nRet = ctl.StartAdvanced(50,4,0
```

StartProfessional

Starts a test located on the VE Desktop Server or the local host in Professional mode. By default, and unless a full path is entered in the command, the CLI searches the main VE Desktop Server folder.

Syntax

External DLL

```
int VED_StartProfessional(char* pszPath);
```

The following parameters are used to configure the StartProfessional command.

- ◆ pszPath: the location of the.vet file you want to run

COM

Interface: ShunraVEDController

```
HRESULT StartProfessional([in] BSTR strPath, [out,retval] LONG* pResult);
```

Parameters

- ◆ strPath: the location of the.vet file

CLI

```
VEDCLI -StartProf path
```

Parameters

- ◆ path: the location of the.vet file

Examples

External DLL

```
VED_StartProfessional("c:\\tests\\4TransLat200  
Lat50.vet");
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet =  
ctrl.StartProfessional("c:\tests\4TransLat200L  
at50.vet")
```

CLI

```
VEDCLI -StartProf c:\tests\4TransLat200Lat50.vet
```

StopTest

Stops the test. This command is only valid if VE Desktop client is running a test.

Syntax

External DLL

```
int VED_StopTest();
```

There are no additional parameters available for this command.

COM

Interface: ShunraVEDController

```
HRESULT StopTest([out,retval] LONG* pResult);
```

CLI

```
VEDCLI -Stop
```

Examples

External DLL

```
VED_StopTest();
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
nRet = ctl.StopTest()
```

CLI

```
VEDCLI -Stop
```

StartTransaction

In the External DLL and CLI interfaces: if called without the transaction name parameter, starts the next coming transaction in a Test Plan (Professional mode); otherwise adds and starts a dynamic transaction with the name provided.

In the COM interface: requires a mandatory transaction name parameter; adds and starts a dynamic transaction with the provided name. for more information about how to start the next transaction in a Test Plan via the COM API, see StartNextTransaction.

Syntax

External DLL

```
int VED_StartTransaction(char*
pszTransName=NULL);
```

The following parameters can be used to configure the StartTransaction command.

- ◆ pszTransName:
 - ❖ when null: starts the next transaction
 - ❖ when not null: adds new transaction with the given name and starts it

COM

Interface: ShunraVEDController

```
HRESULT StartTransaction([in] BSTR strTransName,
[out,retval] LONG* pResult);
```

Parameters

- ◆ strTransName: mandatory parameter; adds new transaction with the given name and starts it

CLI

```
VEDCLI -StartTrans [name [description]]
```

Parameters

- ◆ name:
 - ❖ when not given: starts the next transaction
 - ❖ when given: adds new transaction with the given name and starts it
- ◆ description: transaction description

Examples

```
VED_StartTransaction();  
VED_StartTransaction("Login");
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
  
nRet = ctl.StartTransaction("Login")
```

CLI

```
VEDCLI -StartTrans LoginTrans
```

StartNextTransaction

Relevant only for the COM interface. Starts the next coming transaction in a Test Plan (Professional mode).

External DLL

N/A

CLI

N/A

COM

Interface: ShunraVEDController

```
HRESULT StartNextTransaction([out,retval] LONG*  
pResult);
```

Examples

COM (using VB Script)

```
Dim ctl, nRet  
  
On Error Resume Next  
  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
  
nRet = ctl.StartNextTransaction()
```

NextNetwork

Loads the next NetworX file from the Test Plan into VE Desktop client.

Syntax

External DLL

```
External DLLint VED_NextNetwork();
```

There are no additional parameters available for this command.

COM

Interface: ShunraVEDController

```
HRESULT StartNextTransaction([out,retval] LONG*  
pResult);
```

CLI

```
VEDCLI -NextNTX
```

Examples

External DLL

```
VED_NextNetwork();
```

COM (using VB Script)

```
Dim ctl, nRet  
  
On Error Resume Next  
  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
  
nRet = ctl.NextNetwork()
```

CLI

```
VEDCLI -NextNTX
```

RestartBusinessProcess

Restarts the current Business Process from the first transaction.

Syntax

External DLL

```
int VED_RestartBusinessProcess();
```

There are no additional parameters available for this command.

COM

Interface: ShunraVEDController

```
HRESULT RestartBusinessProcess([out,retval] LONG*  
pResult);
```

CLI

```
VEDCLI -StopBProc repeat
```

Examples

External DLL

```
VED_RestartBusinessProcess();
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
nRet = ctl.RestartBusinessProcess()
```

CLI

```
VEDCLI -StopBProc repeat
```

NextBusinessProcess

Loads the next Business Process in the Test Plan.

Syntax

External DLL

```
int VED_NextBusinessProcess();
```

There are no additional parameters available for this command.

COM

Interface: ShunraVEDController

```
HRESULT NextBusinessProcess([out,retval] LONG*  
pResult);
```

CLI

```
VEDCLI -StopBProc continue
```

Examples

External DLL

```
VED_NextBusinessProcess();
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
nRet = ctl.NextBusinessProcess()
```

CLI

```
VEDCLI -StopBProc continue
```

StopTransaction

Stops current transaction in Test Plan.

Syntax

External DLL

```
int VED_StopTransaction(int nStatus);
```

The following parameters are used to configure the StopTransaction status.

- ✦ **nStatus**: sets the current transaction as success, failure, 1 or 0.

COM

Interface: ShunraVEDController

```
HRESULT StopTransaction([in] LONG nStatus,  
[out,retval] LONG* pResult);
```

Parameters

- ✦ **nStatus**: sets the current transaction as success (1) or failure (0).

CLI

```
VEDCLI -StopTrans status
```

Parameters

- ✦ **status**: 1 for success, 0 for failure (also "success" / "failure" string values).

Examples

External DLL

```
VED_StopTransaction(1);
```

COM (using VB Script)

```
Dim ctl, nRet
```

```
On Error Resume Next
Set ctrl =
CreateObject("ShunraVEAPI.ShunraVEDController")
nRet = ctrl.StopTransaction(1)
```

CLI

```
VEDCLI -StopTrans status
```

SaveRun

Saves test results to the indicated path.

Syntax

External DLL

```
int VED_SaveRun(char* pszLocalPath);
```

The following parameters are used to configure the SaveRun command.

- ✦ **pszLocalPath:** path to the location in which you want to save the test

COM

Interface: ShunraVEDController

```
HRESULT SaveRun([in] BSTR strLocalPath, [out,retval]  
LONG* pResult);Example
```

Parameters

- ✦ **strLocalPath:** path to the local folder in which you want to save the test

CLI

```
VEDCLI -Save path
```

Parameters

path: path to the local folder in which you want to save the test

Note: Directory name is not required. If given, a new directory will be created to house the test results.

Examples

External DLL

```
VED_SaveRun("c:\\Runs");
```

Giving this command will result in a folder named Runs being created in the C:/ drive and all test results being saved in it.

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
nRet = ctrl.SaveRun("c:\runs")
```

CLI

```
VEDCLI -Save c:\runs
```

UploadRun

Uploads test results to VE Desktop Server.

Syntax

External DLL

```
int VED_UploadRun(char* pszServerPath);
```

Note: When saving a test, you can only save to those directories on the VE Desktop Server to which you have permission.

The following parameters are used to configure the UploadRun command.

- ✦ `pszServerPath`: the subdirectory under the root on the remote server to which you want to upload the test

COM

Interface: ShunraVEDController

```
HRESULT UploadRun([in] BSTR strServerPath,  
[out,retval] LONG* pResult);
```

Parameters

- ✦ `strServerPath`: the subdirectory under the root on the remote server to which you want to upload the test

CLI

```
VEDCLI -Upload remote_folder
```

Parameters

- ✦ `remote_folder`: the subdirectory under the root on the remote server to which you want to upload the test

Examples

```
VED_UploadRun("Results");
```

Using this command uploads the results of the last test to the VE Desktop Server folder named Results.

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
nRet = ctrl.UploadRun("Results")
```

CLI

```
VEDCLI -Upload Results
```

SetLogFile

Activates log file which records events, including all CLI commands executed and errors encountered. Each log entry is recorded with a time and date stamp.

Syntax

External DLL

```
int VED_SetLogFile(char* pszPath);
```

The following parameters are used to configure the SetLogFile command.

- ✦ pszPath: sets the path to the location to save the log and the log filename

COM

Interface: ShunraVEDController

```
HRESULT SetLogFile([in] BSTR strPath, [out,retval]  
LONG* pResult);
```

Parameters

- ✦ strPath: the log file name (including full path)

CLI

```
VEDCLI -SetLogFile path
```

Parameters

- ✦ path: the log file name (including full path)

Examples

External DLL

```
VED_SetLogFile("C:\\TraceFiles\\API_Trace.log");
```

COM (using VB Script)

```
Dim ctl, nRet

On Error Resume Next

Set ctrl =
CreateObject("ShunraVEAPI.ShunraVEDController")

nRet =
ctrl.SetLogFile("C:\TraceFiles\API_Trace.log")
```

CLI

```
VED_SetLogFile("C:\\TraceFiles");
```

ActivatePacketList

Adds a Packet List to a test.

Syntax

External DLL

```
int VED_ActivatePacketList();
```

Note: Current settings regarding packet list parameters are determined by what is configured in the VE Desktop Standard options dialog box.

There are no additional parameters available for this command.

COM

Interface: ShunraVEDController

```
HRESULT ActivatePacketList([out,retval] LONG*  
pResult);
```

CLI

```
VEDCLI -StartPacketList
```

Examples

External DLL

```
VED_ActivatePacketList();
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet = ctrl.ActivatePacketList()
```

CLI

```
VED_ActivatePacketList();
```

CancelPacketList

Removes a Packet List from a test.

Syntax

External DLL

```
int VED_CancelPacketList();
```

There are no additional parameters available for this command.

COM

Interface: ShunraVEDController

```
HRESULT CancelPacketList([out,retval] LONG*  
pResult);
```

CLI

```
VEDCLI -SctopPacketList
```

Examples

External DLL

```
VED_CancelPacketList();
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
nRet = ctl.CancelPacketList()
```

CLI

```
VED_CancelPacketList();
```

LicenseCheckout

Checks a VE Desktop client license out from the server.

Syntax

External DLL

```
int VED_LicenseCheckout(int nDays);
```

The following parameters are used to configure the LicenseCheckout command.

- ◆ nDays: number of days for which to checkout the license (1-32000)

COM

Interface: ShunraVEDController

```
HRESULT LicenseCheckOut([in] LONG nDays,  
[out,retval] LONG* pResult);
```

Parameters

- ◆ nDays: number of days for which to checkout the license

CLI

```
VEDCLI -LicenseCheckout [days]
```

Parameters

- ◆ days: number of days for which to checkout the license

Examples

External DLL

```
VED_LicenseCheckout(5);
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next
```

```
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
  
nRet = ctrl.LicenseCheckOut(5)
```

CLI

```
VEDCLI -LicenseCheckout 5
```

LicenseCheckin

Checks a license back into the VE Desktop Server.

Syntax

External DLL

```
int VED_LicenseCheckin()
```

There are no additional parameters available for this command.

COM

Interface: ShunraVEDController

```
HRESULT LicenseCheckIn([out,retval] LONG* pResult);
```

CLI

```
VEDCLI -LicenseCheckin
```

Examples

External DLL

```
VED_LicenseCheckin();
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
nRet = ctl.LicenseCheckIn()
```

CLI

```
VEDCLI -LicenseCheckin
```

GetRunningTestDescription

Gets description of the test that is currently running.

Syntax

External DLL

```
int VED_GetRunningTestDescription(char*
    pszDescriptionBuf, int nDescriptionBufLen);
```

Parameters:

- ✦ pszDescriptionBuf: contains the test description upon return
- ✦ nDescriptionBufLen: maximal length of pszDescriptionBuf

CLI

```
VEDCLI -GetRunConf [output_file_path]
```

Parameters:

- ✦ output_file_path: optional output file path; if not set, the standard output is used

COM

Interface: ShunraVEDController

```
HRESULT GetRunningTestDescription([out]
    VARIANT* pDescription, [out,retval] LONG*
    pResult);
```

Parameters:

- ✦ pDescription: contains the test description upon return

Examples

External DLL

```
char sTestDescription[200];

VED_GetRunningTestDescription
(sTestDescription,1000);
```

CLI

```
VEDCLI -GetRunConf c:\TestDescription.log
```

COM (using VB Script)

```
Dim ctl, nRet, sDescription  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet =  
ctrl.GetRunningTestDescription(sDescription)
```

GetErrorString

Gets the error description for a given error code.

Syntax

External DLL

```
int VED_GetErrorString(DWORD dwErrorCode, char*  
pszErrorStrBuf, int nErrorStrBufLen);
```

Parameters:

- ✦ dwErrorCode: error code to return description for.
- ✦ pszErrorStrBuf: contains the error description upon return
- ✦ nErrorStrBufLen: maximal length of pszErrorStrBuf

CLI

```
VEDCLI -GetErrStr err_id [output_file_path]
```

Parameters:

- ✦ err_id: error code to return description for.
- ✦ output_file_path: optional output file path; if not set, the standard output is used

COM

Interface: ShunraVEDController

```
HRESULT GetErrorString([in] LONG nCode,  
[out,retval] BSTR* pText);
```

Parameters:

- ✦ nCode: error code for which the description is returned
- ✦ pText: error description, the return value of the function

Examples

External DLL

```
char sError[200];  
VED_GetErrorString(2, sError,200);
```

CLI

```
VEDCLI -GetErrStr 2 c:\ErrorDescription.log
```

COM (using VB Script)

```
Dim sErrorDescription  
On Error Resume Next  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
sErrorDescription = ctrl.GetErrorString(2)
```

IsEmulationRunning

Gets whether the VED emulation is currently running.

Syntax

External DLL

N/A

CLI

N/A

COM

```
HRESULT IsEmulationRunning([out, retval]  
    VARIANT_BOOL* pVal)
```

Examples

COM (using VB Script)

```
Dim ctrl, bEmulationRunning  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
bEmulationRunning = ctrl.IsEmulationRunning()
```

SetAutoAnalysisParams

Sets parameters for automatically conducting post-test analysis.

Syntax

External DLL

```
int VED_SetAutoAnalysisParams(short bEnable,  
char* pszProtocol, short bClientServer);
```

Parameters:

- ◆ bEnable: enable (1) / disable (0) post-test analysis.
- ◆ pszProtocol: "HTTP", "TCP", "UDP" or "HTTPS".
- ◆ bClientServer: conduct analysis from client to server (1) / from server to client (0)

CLI

```
VEDCLI -SetAutoAnalysis execute protocol  
direction
```

Parameters:

- ◆ execute: yes, no, 1 or 0
- ◆ protocol: HTTP, HTTPS, TCP or UDP
- ◆ direction: cs or sc (client-server or server-client)

COM

Interface: ShunraVEDController

```
HRESULT SetAutoAnalysisParams([in]  
VARIANT_BOOL bExecute, [in] BSTR strType, [in]  
VARIANT_BOOL bDirectionClientServer,  
[out,retval] LONG* pResult);
```

Parameters:

- ◆ bExecute: conduct (VARIANT_TRUE) / do not conduct (VARIANT_FALSE) analysis.

- ✦ strType: "HTTP", "TCP", "UDP" or "HTTPS".
- ✦ bDirectionClientServer: from client to server (VARIANT_TRUE) / from server to client (VARIANT_FALSE).

Examples

External DLL

```
VED_ SetAutoAnalysisParams (1, "HTTP", 1);
```

CLI

```
VEDCLI -SetAutoAnalysis yes HTTP cs
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet = ctrl.SetAutoAnalysisParams(true,"HTTP",  
true)
```

SetKeepRunHistory

Sets whether to keep the runs history (override or not the previous run data in the database).

Syntax

External DLL

```
int VED_SetKeepRunHistory (short bKeep);
```

Parameters:

✦ bKeep: keep (1) / do not keep (0).

CLI

```
VEDCLI -SetKeepRunHistory keep
```

Parameters:

✦ keep: yes, no, 1 or 0

COM

Interface: ShunraVEDController

```
HRESULT SetKeepRunHistory([in] VARIANT_BOOL  
bKeep, [out,retval] LONG* pResult);
```

Parameters:

✦ bKeep: keep (VARIANT_TRUE) / do not keep (VARIANT_FALSE).

Examples

External DLL

```
VED_SetKeepRunHistory (1);
```

CLI

```
VEDCLI - SetKeepRunHistory yes
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet = ctl.SetKeepRunHistory(true)
```

GetKeepRunHistory

Obtains setting of whether the run history is kept or not kept

Syntax

External DLL

```
int VED_GetKeepRunHistory(short* bKeep);
```

Parameters:

- ◆ `bKeep`: upon return contains the keep history value (1 or 0).

CLI

```
VEDCLI -GetKeepRunHistory [output_file_path]
```

Parameters:

- ◆ `output_file_path`: optional output file path; if not set, the standard output is used

COM

Interface: `ShunraVEDController`

```
HRESULT GetKeepRunHistory([out] VARIANT*  
pbKeep, [out,retval] LONG* pResult);
```

Parameters:

- ◆ `pbKeep`: upon return contains `VARIANT_TRUE` / `VARIANT_FALSE` in the `boolval` field.

Examples

External DLL

`short bKeep;`

```
VED_GetKeepRunHistory (&bKeep);
```

CLI

```
VEDCLI -GetKeepRunHistory c:\KeepRunHistory.log
```

COM (using VB Script)

```
Dim ctrl, nRet, bKeepRunHistory  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
  
nRet = ctrl.GetKeepRunHistory(bKeepRunHistory)
```

SetIPFilterParameters

Sets IP Filter parameters.

Syntax

External DLL

```
int VED_SetIPFilterParameters(short bExclude,  
    short bImportedOverride);
```

Parameters:

- ◆ bExclude: 1 for exclude policy, 0 for include policy.
- ◆ bImportedOverride: override (1) / do not override (0)
imported network filter

CLI

```
VEDCLI -SetIPFilterParams policy  
    override_imported
```

Parameters:

- ◆ policy: exclude or include
- ◆ override_imported: yes, no, 1 or 0

COM

Interface: ShunraVEDController

```
HRESULT SetIPFilterParameters([in]  
    VARIANT_BOOL bExclude, [in] VARIANT_BOOL  
    bImportedOverride, [out,retval] LONG* pResult);
```

Parameters:

- ◆ bExclude: VARIANT_TRUE for exclude policy,
VARIANT_FALSE for include policy.
- ◆ bImportedOverride: override (VARIANT_TRUE) / do not
override (VARIANT_FALSE) imported network filter.

Examples

External DLL

```
VED_SetIPFilterParameters(1,1);
```

CLI

```
VEDCLI -SetIPFilterParams exclude yes
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet = ctrl.SetIPFilterParameters(true,true)
```

GetIPFilterParameters

Gets IP Filter parameters.

Syntax

External DLL

```
int VED_GetIPFilterParameters(short* bExclude,  
short* bImportedOverride);
```

Parameters:

- ◆ `bExclude`: upon return contains the filter policy (1 for exclude, 0 for include).
- ◆ `bImportedOverride`: upon return contains the override imported network value (1 for override, 1 for do not override).

CLI

```
VEDCLI -GetIPFilterParams [output_file_path]
```

Parameters:

- ◆ `output_file_path`: optional output file path; if not set, the standard output is used

COM

Interface: `ShunraVEDController`

```
HRESULT GetIPFilterParameters([out] VARIANT*  
pbExclude, [out] VARIANT* pbImportedOverride,  
[out,retval] LONG* pResult);
```

Parameters:

- ◆ `pbExclude`: upon return contains the filter policy in boolval field (`VARIANT_TRUE` for exclude, `VARIANT_FALSE` for include).
- ◆ `pbImportedOverride`: upon return contains the override imported network value in boolval field (`VARIANT_TRUE` for override, `VARIANT_FALSE` for do not override).

Examples

External DLL

```
short bExclude, bImportedOverride;  
VED_GetIPFilterParameters(&bExclude,  
&bImportedOverride);
```

CLI

```
VEDCLI -GetIPFilterParams  
c:\IPFilterParams.log
```

COM (using VB Script)

```
Dim ctrl, nRet, bPolicy, bOverrideImported  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet =  
ctrl.GetIPFilterParameters(bPolicy,bOverrideIm  
ported)
```

GetIPFilterItem

Gets IP from IP Filter list by index.

Syntax

External DLL

```
int VED_GetIPFilterItem(int nIndex, char*  
pszIP, int nIPBufLen);
```

Parameters:

- ◆ nIndex: zero based index into the Filter list.
- ◆ pszIP: upon return contains the IP of the required item.
- ◆ nIPBufLen: maximal size of pszIP.

CLI

```
VEDCLI -GetIPFilterItem index  
[output_file_path]
```

Parameters:

- ◆ index: zero based index into the Filter list.
- ◆ output_file_path: optional output file path; if not set, the standard output is used.

COM

```
HRESULT GetIPFilterItem([in] LONG nIndex, [out]  
VARIANT* pstrIP, [out,retval] LONG* pResult);
```

Parameters:

- ◆ nIndex: zero based index into the Filter list.
- ◆ pszIP: upon return contains the IP of the required item in the bstrVal field.

Examples

External DLL

```
char szIP[20];  
VED_GetIPFilterItem(0,szIP,20);
```

CLI

```
VEDCLI -GetIPFilterItem 0 c:\IPFilterItem.log
```

COM (using VB Script)

```
Dim ctrl, nRet, sIPFilterItem  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet = ctrl.GetIPFilterItem(0,sIPFilterItem)
```

GetIPFilterCount

Gets IP from IP Filter list by index.

Syntax

External DLL

```
int VED_GetIPFilterItem(int nIndex, char*  
pszIP, int nIPBufLen);
```

Parameters:

- ◆ nIndex: zero based index into the Filter list.
- ◆ pszIP: upon return contains the IP of the required item.
- ◆ nIPBufLen: maximal size of pszIP.

CLI

```
VEDCLI -GetIPFilterItem index  
[output_file_path]
```

Parameters:

- ◆ index: zero based index into the Filter list.
- ◆ output_file_path: optional output file path; if not set, the standard output is used.

COM

Interface: ShunraVEDController

```
HRESULT GetIPFilterItem([in] LONG nIndex, [out]  
VARIANT* pstrIP, [out,retval] LONG* pResult);
```

Parameters:

- ◆ nIndex: zero based index into the Filter list.
- ◆ pszIP: upon return contains the IP of the required item in the bstrVal field.

Examples

External DLL

```
char szIP[20];  
VED_GetIPFilterItem(0,szIP,20);
```

CLI

```
VEDCLI -GetIPFilterItem 0 c:\IPFilterItem.log
```

COM (using VB Script)

```
Dim ctrl, nRet, sIPFilterItem  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet = ctrl.GetIPFilterItem(0,sIPFilterItem)
```

AddIPFilterItem

Adds IP to the IP Filter list.

Syntax

External DLL

```
int VED_AddIPFilterItem(char* pszIP);
```

Parameters:

- ◆ pszIP: IP to be added to the IP Filter list.

CLI

```
VEDCLI -AddToIPFilter ip
```

Parameters:

- ◆ ip: IP to be added to the IP Filter list.

COM

Interface: ShunraVEDController

```
HRESULT AddIPFilterItem([in] BSTR strIP,  
[out,retval] LONG* pResult);
```

Parameters:

- ◆ strIP: IP to be added to the IP Filter list.

Examples

External DLL

```
VED_AddIPFilterItem("192.168.1.82");
```

CLI

```
VEDCLI -AddToIPFilter 192.168.1.82
```

COM (using VB Script)

```
Dim ctl, nRet
```

On Error Resume Next

Set ctrl =

CreateObject("ShunraVEAPI.ShunraVEDController"
)

nRet = ctrl.AddIPFilterItem("192.168.1.82")

RemoveIPFilterItem

Removes IP address from the IP Filter list

Syntax

External DLL

```
int VED_RemoveIPFilterItem(char* pszIP);
```

Parameters:

- ◆ pszIP: IP to be removed from the IP Filter list.

CLI

```
VEDCLI -RemoveFromIPFilter ip
```

Parameters:

- ◆ ip: IP to be removed from the IP Filter list.

COM

Interface: ShunraVEDController

```
HRESULT RemoveIPFilterItem([in] BSTR strIP,  
[out,retval] LONG* pResult);
```

Parameters:

- ◆ strIP: IP to be removed from the IP Filter list.

Examples

External DLL

```
VED_RemoveIPFilterItem("192.168.1.82");
```

CLI

```
VEDCLI -RemoveFromIPFilter 192.168.1.82
```

COM (using VB Script)

```
Dim ctl, nRet
```

On Error Resume Next

Set ctrl =

CreateObject("ShunraVEAPI.ShunraVEDController"
)

nRet = ctrl.RemoveIPFilterItem("192.168.1.82")

ClearIPFilter

Clears the IP Filter list.

Syntax

External DLL

```
int VED_ClearIPFilter();
```

CLI

```
VEDCLI -ClearIPFilter
```

COM

Interface: ShunraVEDController

```
HRESULT ClearIPFilter([out,retval] LONG*  
pResult);
```

Examples

External DLL

```
VED_ ClearIPFilter();
```

CLI

```
VEDCLI -ClearIPFilter
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
nRet = ctrl.ClearIPFilter()
```

SetPacketListParameters

Sets the packet capture parameters.

Syntax

External DLL

```
int VED_SetPacketListParameters(short  
bCapturePackets, short bCaptureCyclic, int  
nCapturePacketSize,  
  
int nCaptureBufferSize,  
  
short bCaptureTcpEfficiency);
```

Parameters:

- ✦ bCapturePackets: enable (1) / disable (0) packet capture (this parameter actually substitutes for the VED_ActivatePacketList/VED_CancelPacketList functions)
- ✦ bCaptureCyclic: capture packets in cycle (1) / not in cycle (0)
- ✦ nCapturePacketSize: maximal captured packet size (bytes, valid values between 14 and 1500)
- ✦ nCaptureBufferSize: maximal size of packets buffer (MB, valid values between 1 and 100)
- ✦ bCaptureTcpEfficiency: capture (1) / do not capture (0) packets for TCP efficiency

CLI

```
VEDCLI -SetPacketListParams capture cyclic  
packet_size buffer_size capture_tcp_efficiency
```

Parameters:

- ✦ capture: yes, no, 1 or 0; actually substitutes for -StartPacketList/-StopPacketList
- ✦ cyclic: yes, no, 1 or 0
- ✦ packet_size: integer between 14 and 1500

- ◆ `buffer_size`: integer between 1 and 100
- ◆ `capture_tcp_efficiency`: yes, no, 1 or 0

COM

Interface: `ShunraVEDController`

```
HRESULT SetPacketListParameters([in]  
    VARIANT_BOOL bCapturePackets, [in] VARIANT_BOOL  
    bCaptureCyclic, [in] LONG nCapturePacketSize,  
    [in] LONG nCaptureBufferSize, [in] VARIANT_BOOL  
    bCaptureTcpEfficiency, [out,retval] LONG*  
    pResult);
```

Parameters:

- ◆ `bCapturePackets`: enable (`VARIANT_TRUE`) / disable (`VARIANT_FALSE`) packet capture (this parameter actually substitutes for the `ActivatePacketList/CancelPacketList` functions)
- ◆ `bCaptureCyclic`: capture packets in cycle (`VARIANT_TRUE`) / not in cycle (`VARIANT_FALSE`)
- ◆ `nCapturePacketSize`: maximal captured packet size (bytes, valid values between 14 and 1500)
- ◆ `nCaptureBufferSize`: maximal size of packets buffer (MB, valid values between 1 and 100)
- ◆ `bCaptureTcpEfficiency`: capture (`VARIANT_TRUE`) / do not capture (`VARIANT_FALSE`) packets for TCP efficiency

Examples

External DLL

```
VED_SetPacketListParameters(1,1,1200,50,1);
```

CLI

```
VEDCLI -SetPacketListParams yes yes 1200 50 yes
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet =  
ctrl.SetPacketListParameters(true,true,1200,50  
,true)
```

GetPacketListParameters

Gets the packet capture parameters.

Syntax

External DLL

```
int VED_GetPacketListParameters(short*  
    bCapturePackets,  
    short* bCaptureCyclic,  
    int* nCapturePacketSize,  
    int* nCaptureBufferSize,  
    short* bCaptureTcpEfficiency);
```

Parameters:

- ◆ **bCapturePackets**: upon return contains packet capture enabled value (1 for enabled, 0 for disabled)
- ◆ **bCaptureCyclic**: upon return contains cyclic capture value (1 for cyclic, 0 for non-cyclic)
- ◆ **nCapturePacketSize**: upon return contains maximal captured packet size
- ◆ **nCaptureBufferSize**: upon return contains maximal size of packets buffer
- ◆ **bCaptureTcpEfficiency**: upon return contains capture TCP efficiency value (1 for capture, 0 for do not capture)

CLI

```
VEDCLI -GetPacketListParams [output_file_path]
```

Parameters:

- ◆ **output_file_path**: optional output file path; if not set, the standard output is used.

COM

```
HRESULT GetPacketListParameters([out] VARIANT*
pbCapturePackets,
[out] VARIANT* pbCaptureCyclic,
[out] VARIANT* pnCapturePacketSize,
[out] VARIANT* pnCaptureBufferSize,
[out] VARIANT* pbCaptureTcpEfficiency,
[out,retval] LONG* pResult);
```

Parameters:

- ✦ pbCapturePackets: upon return contains packet capture enabled value (VARIANT_TRUE for enabled, VARIANT_FALSE for disabled) in the boolVal field
- ✦ pbCaptureCyclic: upon return contains cyclic capture value (VARIANT_TRUE for cyclic, VARIANT_FALSE for non-cyclic) in the boolVal field
- ✦ pnCapturePacketSize: upon return contains maximal captured packet size in the lVal field
- ✦ pnCaptureBufferSize: upon return contains maximal size of packets buffer in the lVal field
- ✦ pbCaptureTcpEfficiency: upon return contains capture TCP efficiency value (VARIANT_TRUE for capture, VARIANT_FALSE for do not capture) in the boolVal field

Examples**External DLL**

```
short bCapturePackets, bCaptureCyclic, bCaptureCy-
clic;

int nCapturePacketSize, nCaptureBufferSize;

VED_GetPacketListParameters(&bCapturePackets,
&bCaptureCyclic,
```

```
&nCapturePacketSize, &nCaptureBufferSize,  
&bCaptureCyclic);
```

CLI

```
VEDCLI -GetPacketListParams  
c:\PacketListParams.log
```

COM (using VB Script)

```
Dim ctl, nRet, bCapturePackets, bCaptureCyclic,  
nPacketSize, nPacketBufferSize,  
bCaptureForTCPEfficiency  
  
On Error Resume Next  
  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet =  
ctl.GetPacketListParameters(bCapturePackets,  
bCaptureCyclic, nPacketSize,  
nPacketBufferSize, bCaptureForTCPEfficiency)
```

SetAutoSaveTestResults

Sets parameters for automatically saving test results at the end of test.

Syntax

External DLL

```
int VED_SetAutoSaveTestResults(short bSave,  
    char* pszPath);
```

Parameters:

- ✦ bSave: save (1) / do not save (0) test results.
- ✦ pszPath: path to a valid local folder to save the results in.

CLI

```
VEDCLI -SetAutoSaveTest save path
```

Parameters:

- ✦ save: yes, no, 1 or 0
- ✦ path: path to a valid local folder.

COM

Interface: ShunraVEDController

```
HRESULT SetAutoSaveTestResults([in]  
    VARIANT_BOOL bSave, [in] BSTR strPath,  
    [out,retval] LONG* pResult);
```

Parameters:

- ✦ bSave: save (VARIANT_TRUE) / do not save (VARIANT_FALSE) test results.
- ✦ pszPath: path to a valid local folder.

Examples

External DLL

```
VED_SetAutoSaveTestResults(1,  
"C:\\TestResults");
```

CLI

```
VEDCLI -SetAutoSaveTest yes c:\\TestResults
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
  
Set ctl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet =  
ctl.SetAutoSaveTestResults(true, "c:\\TestResults")
```

SetResultFolderPrefix

Sets prefix to be added to the result folder names when saving/uploading test results.

Syntax

External DLL

```
int VED_SetResultFolderPrefix(char*  
    pszPrefix);
```

Parameters:

- ✦ `pszPrefix`: prefix to be added to the result folder names (up to 16 characters).

CLI

```
VEDCLI -SetResultFolderPrefix prefix
```

Parameters:

- ✦ `prefix`: prefix to be added to the result folder names

COM

Interface: ShunraVEDController

```
HRESULT SetResultFolderPrefix([in] BSTR  
    bsPrefix, [out,retval] LONG* pResult);
```

Parameters:

- ✦ `bsPrefix`: prefix to be added to the result folder names.

Examples

External DLL

```
VED_SetResultFolderPrefix("MyPrefix");
```

CLI

```
VEDCLI -SetResultFolderPrefix MyPrefix
```

COM (using VB Script)

```
Dim ctl, nRet
On Error Resume Next

Set ctrl =
CreateObject("ShunraVEAPI.ShunraVEDController"
)

nRet = ctrl.SetResultFolderPrefix("MyPrefix")
```

GetAutoSaveTestResults

Gets parameters of automatically saving test results at the end of test.

Syntax

External DLL

```
int VED_GetAutoSaveTestResults(short* bSave,  
char* pszPath);
```

Parameters:

- ✦ **bSave**: upon return contains the auto save test results value (1 for save, 0 for do not save).
- ✦ **pszPath**: upon return contains path to a valid local folder.

CLI

```
VEDCLI -GetAutoSaveTest [output_file_path]
```

Parameters:

- ✦ **output_file_path**: optional output file path; if not set, the standard output is used

COM

```
HRESULT GetAutoSaveTestResults([out] VARIANT*  
pbSave, [out] VARIANT* pstrPath, [out,retval]  
LONG* pResult);
```

Parameters:

- ✦ **pbSave**: upon return contains the auto save test results value (VARIANT_TRUE for save, VARIANT_FALSE for do not save) in the **booVal** field.
- ✦ **pstrPath**: upon return contains path to a valid local folder in the **bstrVal** field.

Examples

External DLL

```
char psPath[100];  
short bSave;  
VED_GetAutoSaveTestResults(&bSave, pszPath);
```

CLI

```
VEDCLI -GetAutoSaveTest  
c:\AutoSaveTestParams.log
```

COM (using VB Script)

```
Dim ctl, nRet, bAutoSaveTestResults,  
sAutoSavePath  
  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet =  
ctrl.GetAutoSaveTestResults(bAutoSaveTestResults,  
sAutoSavePath)
```

SetCalculateTopTalkers

Sets the top talker calculation parameters.

Syntax

External DLL

```
int VED_SetCalculateTopTalkers(short  
    bCalculateTopTalkers, int nMaxTopTalkerPairs);
```

Parameters:

- ✦ bCalculateTopTalkers: 1 for calculate, 0 for do not calculate.
- ✦ nMaxTopTalkerPairs: maximal number of top talker pairs to show (valid values between 1 and 10).

CLI

```
VEDCLI -SetCalcTopTalkers calc max_pairs
```

Parameters:

- ✦ calc: yes, no, 1 or 0
- ✦ max_pairs: maximal number of top talker pairs

COM

Interface: ShunraVEDController

```
HRESULT SetCalculateTopTalkers([in]  
    VARIANT_BOOL bCalculateTopTalkers,  
    [in] LONG nMaxPairs, [out,retval] LONG*  
    pResult);
```

Parameters:

- ✦ bCalculateTopTalkers: VARIANT_TRUE for calculate, VARIANT_FALSE for do not calculate.
- ✦ nMaxPairs: maximal number of top talker pairs

Examples

External DLL

```
VED_SetCalculateTopTalkers(1, 8);
```

CLI

```
VEDCLI -SetCalcTopTalkers yes 8
```

COM (using VB Script)

```
Dim ctl, nRet  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController"  
)  
  
nRet = ctrl.SetCalculateTopTalkers(true,8)
```

GetCalculateTopTalkers

Gets the top talker calculation parameters.

Syntax

External DLL

```
int VED_GetCalculateTopTalkers(short*  
    bCalculateTopTalkers,  
    int* nMaxTopTalkerPairs);
```

Parameters:

- ✦ **bCalculateTopTalkers**: upon return contains the calculate top talkers value (1 for calculate, 0 for do not calculate).
- ✦ **nMaxTopTalkerPairs**: upon return contains the maximal number of top talker pairs.

CLI

```
VEDCLI -GetCalcTopTalkers [output_file_path]
```

Parameters:

- ✦ **output_file_path**: optional output file path; if not set, the standard output is used

COM

Interface: ShunraVEDController

```
HRESULT GetCalculateTopTalkers([out] VARIANT*  
    pbCalculateTopTalkers,  
    [out] VARIANT* pnMaxPairs,  
    [out,retval] LONG* pResult);
```

Parameters:

- ✦ **pbCalculateTopTalkers**: upon return contains the calculate top talkers value (VARIANT_TRUE for calculate, VARIANT_FALSE for do not calculate) in the boolVal field.

- ◆ pnMaxPairs: upon return contains the maximal number of top talker pairs

Examples

External DLL

```
short bCalculateTopTalkers;  
int nMaxTopTalkerPairs;  
VED_GetCalculateTopTalkers(&bCalculateTopTalkers,  
    &nMaxTopTalkerPairs);
```

CLI

```
VEDCLI -GetCalcTopTalkers  
c:\CalcTopTalkerParams.log
```

COM (using VB Script)

```
Dim ctrl, nRet, bCalculate, nMaxPairs  
On Error Resume Next  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEDController")  
  
nRet = ctrl.GetCalculateTopTalkers(bCalculate,  
nMaxPairs)
```

VEDStats_GetCurrentStatistics

Gets the last second runtime statistics xml.

Example of returned statistics xml:

```
<statistics version="1.0">
  <items>
    <item description="Current State" type="STATE"
      id="ID_CURRENT_STATE">
      <params>
        <param description="Running"
          name="running">1</param>
        <param description="Current Configuration"
          name="current_config">0</param>
      </params>
    </item>
    <item description="Server" type="LOCATION"
      id="ID_SERVER_GW_NIC_2">
      <params>
        <param description="Total Throughput In
          [bytes]" name="total_in">6190</param>
        <param description="Total Throughput Out
          [bytes]" name="total_out">6372</param>
        <param description="Current In [bps]"
          name="bps_in">72</param>
        <param description="Current Out [bps]"
          name="bps_out">349</param>
        <param description="Bandwidth Util In [%]"
          name="bw_util_in">0.0</param>
        <param description="Bandwidth Util Out [%]"
          name="bw_util_out">0.5</param>
      </params>
    </item>
  </items>
</statistics>
```

```
</params>

</item>

<item description="Client" type="LOCATION"
id="ID_CLIENT_GW_NIC_1">
<params>

<param description="Total Throughput In
[bytes]" name="total_in">6190</param>

<param description="Total Throughput Out
[bytes]" name="total_out">6372</param>

<param description="Current In [bps]"
name="bps_in">72</param>

<param description="Current Out [bps]"
name="bps_out">349</param>

<param description="Bandwidth Util In [%]"
name="bw_util_in">0.0</param>

<param description="Bandwidth Util Out [%]"
name="bw_util_out">0.5</param>

</params>

</item>

</items>

</statistics>
```

Syntax

External DLL

```
int VEStats_GetCurrentStatistics(char*
pszApplicationId, char* pszStatisticsBuf,
int nStatisticsBufLen);
```

Parameters:

- ◆ pszApplicationId: ID of Shunra application for which statistics is required (only "VEDesktop" is currently applicable)

- ✦ pszStatisticsBuf: upon return contains the statistics xml.
- ✦ nStatisticsBufLen: length of pszStatisticsBuf.

CLI

NA

COM

Interface: ShunraVEStatistics

```
long GetCurrentStatistics([in] BSTR  
    strApplicationID,  
                           [out] BSTR* strXML);
```

Parameters:

- ✦ strApplicationID: ID of Shunra application for which statistics is required.
- ✦ strXML: upon return contains the statistics xml

Examples**External DLL**

```
char szStatisticXml[5000];  
  
VEStats_GetCurrentStatistics("VEDesktop",  
    szStatisticXml, 5000);
```

COM (using VB Script)

```
Dim ctrl, nRet, sStatistics  
  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEStatistics")  
  
nRet = ctrl.GetCurrentStatistics("VEDesktop",  
    sStatistics)
```

VEDStats_GetErrorString

Gets statistic polling error description for given error code.

Syntax

External DLL

```
int VEDStats_GetErrorString(DWORD dwErrorCode,  
char* pszErrorStrBuf,  
int nErrorStrBufLen)
```

Parameters:

- ◆ dwErrorCode: error code to return error description for (statistic polling error codes are negative).
- ◆ pszErrorStrBuf: upon return contains the error description.
- ◆ nErrorStrBufLen: length of pszErrorStrBuf.

CLI

N/A

COM

```
Interface: ShunraVEStatisticsBSTR  
GetErrorString([in] long nErrorCode);
```

Parameters:

- ◆ nErrorCode: error code to return error description for.

Examples

External DLL

```
char szErrorStrBuf [200];  
VEDStats_GetErrorString(-2, szErrorStrBuf,  
200);
```

COM (using VB Script)

```
Dim ctrl, sErrorString
On Error Resume Next

Set ctrl =
CreateObject("ShunraVEAPI.ShunraVEStatistics")

sErrorString = ctrl.GetErrorString(-2)
```

VEDStats_GetLastError

Gets the latest statistic polling error code and description.

Syntax

External DLL

```
int VEDStats_GetLastError(int *nErrorCode, char*  
    pszErrorStrBuf,  
    int nErrorStrBufLen);
```

Parameters:

- ◆ **nErrorCode**: upon return contains the error code of the latest error.
- ◆ **pszErrorStrBuf**: upon return contains the error description of the latest error.
- ◆ **nErrorStrBufLen**: length of pszErrorStrBuf.

CLI

N/A

COM

Interface: ShunraVEStatistics

```
void GetLastError([out] long* nErrorCode, [out]  
    BSTR* strErrorMessage);
```

Parameters:

- ◆ **nErrorCode**: upon return contains the error code of the latest error.
- ◆ **strErrorMessage**: upon return contains the error description of the latest error.

Examples

External DLL

```
char szErrorStrBuf [200];  
  
int nErrCode;  
  
VEStats_GetErrorString(&nErrCode,  
szErrorStrBuf, 200);
```

COM (using VB Script)

```
Dim ctrl, nErrorCode, sErrorString, ret  
  
On Error Resume Next  
  
Set ctrl =  
CreateObject("ShunraVEAPI.ShunraVEStatistics")  
  
ret = ctrl.GetLastError(nErrorCode,  
sErrorString)
```

GetHostID

Gets the VED Client's Host ID required for creating the license.

Syntax

External DLL

```
int VEDLM_GetHostID(char* pszHostID, int  
nHostIdBufLen, char* pszMessage, int  
nMessageBufLen)
```

Parameters:

- ◆ pszHostID: upon return contains the Host ID string.
- ◆ nHostIdBufLen: length of pszHostID
- ◆ pszMessage: in case of call failure, upon return contains the error description.
- ◆ nMessageBufLen: length of pszMessage.

CLI

N/A

COM

Interface: VedLmAutomation

```
HRESULT GetHostID([out] VARIANT* pvarHostID,  
[out] VARIANT* pvarError, [out,retval] VARIANT*  
pvarRetVal);
```

Parameters:

- ◆ pvarHostID: : upon return contains the Host ID string in the bstrVal field.
- ◆ pvarError: in case of call failure, upon return contains the error description in the bstrVal field.

Examples

External DLL

```
char szHostID[40], szErrorStrBuf [200];  
  
VEDLM_GetHostID(szHostID, 40, szErrorStrBuf,  
200)
```

COM (using VB Script)

```
Dim ctrl, nRet, sHostID, sError  
  
On Error Resume Next  
  
Set ctrl =  
CreateObject("VEDLicenseMgr.VedLmAutomation")  
  
nRet = ctrl.GetHostID(sHostID,sError)
```

SetLicenseKey

Sets the VED Client's license key.

Syntax

External DLL

```
int VEDLM_SetLicenseKey(char* pszLicKey, char*  
nMessage,  
  
int nMessageBufLen)
```

Parameters:

- ◆ pszLicKey: the license key.
- ◆ nHostIdBufLen: length of pszHostID
- ◆ pszMessage: in case of call failure, upon return contains the error description.
- ◆ nMessageBufLen: length of pszMessage.

CLI

N/A

COM

Interface: VedLmAutomation

```
HRESULT SetLicenseKey([in] VARIANT varLicKey,  
[out] VARIANT* pvarError, [out,retval]  
VARIANT* pvarRetVal);
```

Parameters:

- ◆ varLicKey: in bstrVal field pass the license key.
- ◆ pvarError: in case of call failure, upon return contains the error description in the bstrVal field.

Examples

External DLL

```
char szErrorStrBuf [200];  
  
VEDLM_SetLicenseKey("E032 BC1E 1D7C 5AB7 EABB  
8A4D A7", szErrorStrBuf,200)
```

COM (using VB Script)

```
Dim ctrl, nRet, sError  
  
On Error Resume Next  
  
Set ctrl =  
CreateObject("VEDLicenseMgr.VedLmAutomation")  
  
nRet = ctrl.SetLicenseKey("E032 BC1E 1D7C 5AB7  
EABB 8A4D A7",sError)
```


A

Single User Script Modification

In addition, to mark specific transactions, use the VE Desktop API to "wrap" the `lr_start_transaction` and `lr_end_transaction` API calls, by changing your scripts to refer to an API named `lr_start_transaction_extended` and `lr_stop_transaction_extended`.

Include the routine shown below that is provided by Shunra. This routine will optionally call the Shunra VE Desktop APIs required for transaction analysis, as long as the script is being run for only one VUser. Shunra's transaction analysis is not currently supported when transactions from more than one VUser are run at the same time, on a single load generator. Note that the transactions must not overlap. If you are using transactions that do overlap, you can manually select non-overlapping transactions as described in the Action section below.

```
vuser_init()
{
    // Begin Shunra specific vuser_init content
    int iWinErr = 0;

    // Load Shunra VE Controller DLL into memory.
    // If the DLL is in the same location as VEDesktop.exe
    // then the script will be able to load it.
    iWinErr = lr_load_dll( "ShunraVEController.dll" );

    if ( iWinErr != 0 )
    {
```

```
                                // Report Error to LR
                                lr_error_message( "Can't load
ShunraVEController.dll" );
}

// End Shunra specific vuser_init content

return 0;
}
```

```
Action()
{

// Begin Shunra specific Action header content

void ProcessIfError(int nError)
{
    int nGetErrorStrError = 0;

    if(nError != 0)
    {
        char sError[500];

        nGetErrorStrError =
VED_GetErrorString(nError,sError,500);

        if(nGetErrorStrError == 0)
            lr_error_message(sError);
    }
}

void lr_start_transaction_extended(char *name)
```

```
{  
    int nError;  
  
    nError = VED_StartTransaction(name);  
    lr_start_transaction(name);  
    ProcessIfError(nError);  
  
}
```

```
void lr_end_transaction_extended(char *name, int status)  
{  
    int nError;  
  
    lr_end_transaction(name, status);  
    nError = VED_StopTransaction(1);  
    ProcessIfError(nError);  
}
```

```
// End Shunra specific Action header content
```

```
/* To mark all transactions for Shunra Analysis, replace  
all instances of "lr_start_transaction" with  
"lr_start_transaction_extended" and "lr_end_transaction"  
with "lr_end_transaction_extended".
```

If you do not want to analyze every transaction, you can manually change the function calls around only the transactions you wish to analyze.

Note: Shunra's analysis can not analyze multiple transactions running in parallel. The transactions must run serially (one stops before the next begins).

```
*/
```

```
int nError = 0;
```

```
lr_start_transaction_extended ("Login");
```

```
web_url("192.168.1.44",
"URL=http://192.168.1.44/",
"Resource=0",
"RecContentType=text/html",
"Referer=",
"Snapshot=t1.inf",
"Mode=HTML",
LAST);

lr_end_transaction_extended
("Login", LR_AUTO);

lr_start_transaction_extended
("Select items");

web_image("mainmenu.gif",
"Src=images/mainmenu.gif",
"MapName=map",
"AreaOrdinal=1",
"Snapshot=t2.inf",
LAST);

lr_end_transaction_extended ("Select
items", LR_AUTO);

lr_start_transaction_extended
("Payment");

web_image("signon.gif",
"Src=images/signon.gif",
"Snapshot=t3.inf",
LAST);

lr_end_transaction_extended ("Payment", LR_AUTO);

lr_start_transaction_extended ("Sign Off");
```

```
web_link("Sign Off",
        "Text=Sign Off",
        "Snapshot=t4.inf",
        LAST);

lr_end_transaction_extended ("Sign Off", LR_AUTO);

return 0;
}

vuser_end()
{

    Return 0;

}
```


B

Presets.xml file

This file contains data from pre-defined locations. It is located by default in the VE Desktop client in:

C:\Program Files\Shunra Virtual Enterprise\VE Desktop
Network Editor.

In the integration with HP Software it is found by default in:

C:\Program Files\Common Files\Shunra Virtual Enterprise\VE Desktop

WAN Emulation Specifications

```
- <PRESET_VALUES>
- <PRESET_VALUE TYPE="ClientConfiguration">
<PV NAME="VEDServer" VALUE="" Mandatory="yes"
Description="The Server IP/URL has been changed." />
<PV NAME="HTTPSPort" VALUE="" Mandatory="yes"
Description="The HTTPS port has been changed." />
</PRESET_VALUE>
- <PRESET_VALUE TYPE="Bandwidth">
<PV VALUE="15440" />
<PV VALUE="2560" />
<PV VALUE="640" />
<PV VALUE="336" />
```

Appendix B

```
<PV VALUE="144" />

</PRESET_VALUE>

_ <PRESET_VALUE TYPE="Latency">

<PV ORIGIN="Amsterdam" DESTINATION="Atlanta"
VALUE="62.85" />

<PV ORIGIN="Amsterdam" DESTINATION="Chicago"
VALUE="67.85" />

<PV ORIGIN="Amsterdam" DESTINATION="Cleveland"
VALUE="64.35" />

<PV ORIGIN="Amsterdam" DESTINATION="Dallas"
VALUE="67.45" />

<PV ORIGIN="Amsterdam" DESTINATION="Denver"
VALUE="91.45" />

<PV ORIGIN="Amsterdam" DESTINATION="Detroit"
VALUE="65.75" />

<PV ORIGIN="Amsterdam" DESTINATION="Elmhurst"
VALUE="68.7" />

<PV ORIGIN="Amsterdam" DESTINATION="Herndon"
VALUE="64.65" />

<PV ORIGIN="Amsterdam" DESTINATION="Houston"
VALUE="69.75" />

<PV ORIGIN="Amsterdam" DESTINATION="Irvine"
VALUE="82" />

<PV ORIGIN="Amsterdam" DESTINATION="Kansas City"
VALUE="72.2" />

<PV ORIGIN="Amsterdam" DESTINATION="Los Angeles"
VALUE="83.3" />

<PV ORIGIN="Amsterdam" DESTINATION="Miami"
VALUE="82.35" />

<PV ORIGIN="Amsterdam" DESTINATION="New York"
VALUE="67.5" />

<PV ORIGIN="Amsterdam" DESTINATION="Richardson"
VALUE="64.05" />
```

```
<PV ORIGIN="Amsterdam" DESTINATION="San Francisco"
VALUE="91.2" />

<PV ORIGIN="Amsterdam" DESTINATION="Santa Clara"
VALUE="86.3" />

<PV ORIGIN="Amsterdam" DESTINATION="Seattle"
VALUE="90.65" />

<PV ORIGIN="Atlanta" DESTINATION="Amsterdam"
VALUE="62.85" />

<PV ORIGIN="Atlanta" DESTINATION="Boston"
VALUE="13.5" />

<PV ORIGIN="Atlanta" DESTINATION="Chicago"
VALUE="13.505" />

<PV ORIGIN="Atlanta" DESTINATION="Cleveland"
VALUE="10.15" />

<PV ORIGIN="Atlanta" DESTINATION="Copenhagen"
VALUE="60" />

<PV ORIGIN="Atlanta" DESTINATION="Dallas"
VALUE="9.85" />

<PV ORIGIN="Atlanta" DESTINATION="Denver" VALUE="18"
/>

<PV ORIGIN="Atlanta" DESTINATION="Detroit"
VALUE="11.825" />

<PV ORIGIN="Atlanta" DESTINATION="Elmhurst"
VALUE="13.585" />

<PV ORIGIN="Atlanta" DESTINATION="Frankfurt"
VALUE="65.5" />

<PV ORIGIN="Atlanta" DESTINATION="Helsinki"
VALUE="62.6" />

<PV ORIGIN="Atlanta" DESTINATION="Herndon"
VALUE="5.905" />

<PV ORIGIN="Atlanta" DESTINATION="Hong Kong"
VALUE="103.75" />

<PV ORIGIN="Atlanta" DESTINATION="Houston"
VALUE="6.995" />
```

Appendix B

```
<PV ORIGIN="Atlanta" DESTINATION="Irvine"  
VALUE="21.945" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Kansas City"  
VALUE="14.89" />  
  
<PV ORIGIN="Atlanta" DESTINATION="London"  
VALUE="49.795" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Los Angeles"  
VALUE="22.715" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Madrid"  
VALUE="63.35" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Mexico City"  
VALUE="37.39" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Miami" VALUE="6.5"  
/>  
  
<PV ORIGIN="Atlanta" DESTINATION="Milan"  
VALUE="61.05" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Milwaukee"  
VALUE="17" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Minneapolis"  
VALUE="16" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Munich"  
VALUE="60.9" />  
  
<PV ORIGIN="Atlanta" DESTINATION="New York"  
VALUE="8.815" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Oslo" VALUE="74" /  
>  
  
<PV ORIGIN="Atlanta" DESTINATION="Paris"  
VALUE="60.25" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Philadelphia"  
VALUE="10" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Phoenix"  
VALUE="21" />  
  
<PV ORIGIN="Atlanta" DESTINATION="Richardson"  
VALUE="9.445" />
```

```
<PV ORIGIN="Atlanta" DESTINATION="San Diego"
VALUE="26.5" />

<PV ORIGIN="Atlanta" DESTINATION="San Francisco"
VALUE="28" />

<PV ORIGIN="Atlanta" DESTINATION="San Jose"
VALUE="27.5" />

<PV ORIGIN="Atlanta" DESTINATION="Santa Clara"
VALUE="27.57" />

<PV ORIGIN="Atlanta" DESTINATION="Seattle"
VALUE="31" />

<PV ORIGIN="Atlanta" DESTINATION="Seoul"
VALUE="104.2" />

<PV ORIGIN="Atlanta" DESTINATION="Shanghai"
VALUE="134.9" />

<PV ORIGIN="Atlanta" DESTINATION="Singapore"
VALUE="135.95" />

<PV ORIGIN="Atlanta" DESTINATION="St. Louis"
VALUE="19" />

<PV ORIGIN="Atlanta" DESTINATION="Stockholm"
VALUE="56.95" />

<PV ORIGIN="Atlanta" DESTINATION="Sydney"
VALUE="97.15" />

<PV ORIGIN="Atlanta" DESTINATION="Taipei"
VALUE="91.3" />

<PV ORIGIN="Atlanta" DESTINATION="Tampa" VALUE="7" /
>

<PV ORIGIN="Atlanta" DESTINATION="Tokyo"
VALUE="95.35" />

<PV ORIGIN="Atlanta" DESTINATION="Toronto"
VALUE="16" />

<PV ORIGIN="Atlanta" DESTINATION="Vancouver"
VALUE="43.2" />

<PV ORIGIN="Atlanta" DESTINATION="Washington D.C."
VALUE="9.5" />
```

```
<PV ORIGIN="Boston" DESTINATION="Atlanta"
VALUE="13.5" />

<PV ORIGIN="Boston" DESTINATION="Chicago" VALUE="11"
/>

<PV ORIGIN="Boston" DESTINATION="Cleveland"
VALUE="16" />

<PV ORIGIN="Boston" DESTINATION="Dallas"
VALUE="23.5" />

<PV ORIGIN="Boston" DESTINATION="Denver"
VALUE="22.5" />

<PV ORIGIN="Boston" DESTINATION="Detroit" VALUE="15"
/>

<PV ORIGIN="Boston" DESTINATION="Houston"
VALUE="24.5" />

<PV ORIGIN="Boston" DESTINATION="Kansas City"
VALUE="18.5" />

<PV ORIGIN="Boston" DESTINATION="Los Angeles"
VALUE="39.5" />

<PV ORIGIN="Boston" DESTINATION="Miami" VALUE="19" /
>

<PV ORIGIN="Boston" DESTINATION="Milwaukee"
VALUE="16.5" />

<PV ORIGIN="Boston" DESTINATION="Minneapolis"
VALUE="15" />

<PV ORIGIN="Boston" DESTINATION="New York" VALUE="3"
/>

<PV ORIGIN="Boston" DESTINATION="Philadelphia"
VALUE="4" />

<PV ORIGIN="Boston" DESTINATION="Phoenix" VALUE="34"
/>

<PV ORIGIN="Boston" DESTINATION="San Diego"
VALUE="40" />

<PV ORIGIN="Boston" DESTINATION="San Francisco"
VALUE="37" />
```

```
<PV ORIGIN="Boston" DESTINATION="San Jose"
VALUE="43.5" />

<PV ORIGIN="Boston" DESTINATION="Seattle" VALUE="35"
/>

<PV ORIGIN="Boston" DESTINATION="St. Louis"
VALUE="17" />

<PV ORIGIN="Boston" DESTINATION="Tampa" VALUE="24" /
>

<PV ORIGIN="Boston" DESTINATION="Toronto" VALUE="9"
/>

<PV ORIGIN="Boston" DESTINATION="Washington D.C."
VALUE="8.5" />

<PV ORIGIN="Chicago" DESTINATION="Amsterdam"
VALUE="67.85" />

<PV ORIGIN="Chicago" DESTINATION="Atlanta"
VALUE="13.345" />

<PV ORIGIN="Chicago" DESTINATION="Boston" VALUE="11"
/>

<PV ORIGIN="Chicago" DESTINATION="Cleveland"
VALUE="4.27" />

<PV ORIGIN="Chicago" DESTINATION="Copenhagen"
VALUE="62.95" />

<PV ORIGIN="Chicago" DESTINATION="Dallas"
VALUE="11.675" />

<PV ORIGIN="Chicago" DESTINATION="Denver"
VALUE="11.61" />

<PV ORIGIN="Chicago" DESTINATION="Denver" VALUE="12"
/>

<PV ORIGIN="Chicago" DESTINATION="Detroit"
VALUE="3.445" />

<PV ORIGIN="Chicago" DESTINATION="Elmhurst"
VALUE="0.69" />

<PV ORIGIN="Chicago" DESTINATION="Frankfurt"
VALUE="68.6" />
```

Appendix B

```
<PV ORIGIN="Chicago" DESTINATION="Helsinki"  
VALUE="63.85" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Herndon"  
VALUE="7.845" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Hong Kong"  
VALUE="104.85" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Houston"  
VALUE="14.095" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Irvine"  
VALUE="31.105" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Kansas City"  
VALUE="6.1" />
```

```
<PV ORIGIN="Chicago" DESTINATION="London"  
VALUE="52.35" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Los Angeles"  
VALUE="28" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Madrid"  
VALUE="63.8" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Mexico City"  
VALUE="32.925" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Miami"  
VALUE="20.635" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Milan"  
VALUE="57.35" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Milwaukee"  
VALUE="4" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Minneapolis"  
VALUE="4.5" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Munich"  
VALUE="63.95" />
```

```
<PV ORIGIN="Chicago" DESTINATION="New York"  
VALUE="12.115" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Oslo"  
VALUE="77.25" />
```

```
<PV ORIGIN="Chicago" DESTINATION="Paris"
VALUE="63.9" />

<PV ORIGIN="Chicago" DESTINATION="Philadelphia"
VALUE="14" />

<PV ORIGIN="Chicago" DESTINATION="Phoenix"
VALUE="24.5" />

<PV ORIGIN="Chicago" DESTINATION="Richardson"
VALUE="11.41" />

<PV ORIGIN="Chicago" DESTINATION="San Diego"
VALUE="32" />

<PV ORIGIN="Chicago" DESTINATION="San Francisco"
VALUE="25.785" />

<PV ORIGIN="Chicago" DESTINATION="San Jose"
VALUE="31" />

<PV ORIGIN="Chicago" DESTINATION="Santa Clara"
VALUE="25.12" />

<PV ORIGIN="Chicago" DESTINATION="Seattle"
VALUE="25.93" />

<PV ORIGIN="Chicago" DESTINATION="Seoul" VALUE="99"
/>

<PV ORIGIN="Chicago" DESTINATION="Shanghai"
VALUE="143" />

<PV ORIGIN="Chicago" DESTINATION="Singapore"
VALUE="131.95" />

<PV ORIGIN="Chicago" DESTINATION="St. Louis"
VALUE="4" />

<PV ORIGIN="Chicago" DESTINATION="Stockholm"
VALUE="59.8" />

<PV ORIGIN="Chicago" DESTINATION="Sydney"
VALUE="118.75" />

<PV ORIGIN="Chicago" DESTINATION="Taipei"
VALUE="94.2" />

<PV ORIGIN="Chicago" DESTINATION="Tampa" VALUE="15"
/>
```

```
<PV ORIGIN="Chicago" DESTINATION="Tokyo"  
VALUE="88.95" />  
  
<PV ORIGIN="Chicago" DESTINATION="Toronto"  
VALUE="5.5" />  
  
<PV ORIGIN="Chicago" DESTINATION="Vancouver"  
VALUE="31.125" />  
  
<PV ORIGIN="Chicago" DESTINATION="Washington D.C."  
VALUE="11" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Amsterdam"  
VALUE="64.35" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Atlanta"  
VALUE="10.28" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Boston"  
VALUE="16" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Chicago"  
VALUE="4.46" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Copenhagen"  
VALUE="59.4" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Dallas"  
VALUE="15.285" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Denver"  
VALUE="15.935" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Detroit"  
VALUE="13" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Detroit"  
VALUE="2.585" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Elmhurst"  
VALUE="4.74" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Frankfurt"  
VALUE="64.65" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Helsinki"  
VALUE="61.75" />  
  
<PV ORIGIN="Cleveland" DESTINATION="Herndon"  
VALUE="4.625" />
```

```
<PV ORIGIN="Cleveland" DESTINATION="Hong Kong"
VALUE="108.05" />

<PV ORIGIN="Cleveland" DESTINATION="Houston"
VALUE="16.68" />

<PV ORIGIN="Cleveland" DESTINATION="Irvine"
VALUE="31.71" />

<PV ORIGIN="Cleveland" DESTINATION="Kansas City"
VALUE="9.85" />

<PV ORIGIN="Cleveland" DESTINATION="London"
VALUE="49.125" />

<PV ORIGIN="Cleveland" DESTINATION="Los Angeles"
VALUE="33.485" />

<PV ORIGIN="Cleveland" DESTINATION="Madrid"
VALUE="60.25" />

<PV ORIGIN="Cleveland" DESTINATION="Mexico City"
VALUE="39.965" />

<PV ORIGIN="Cleveland" DESTINATION="Miami"
VALUE="16.82" />

<PV ORIGIN="Cleveland" DESTINATION="Milan"
VALUE="55.5" />

<PV ORIGIN="Cleveland" DESTINATION="Milwaukee"
VALUE="16" />

<PV ORIGIN="Cleveland" DESTINATION="Minneapolis"
VALUE="18" />

<PV ORIGIN="Cleveland" DESTINATION="Munich"
VALUE="59.85" />

<PV ORIGIN="Cleveland" DESTINATION="New York"
VALUE="8.5" />

<PV ORIGIN="Cleveland" DESTINATION="Oslo"
VALUE="73.2" />

<PV ORIGIN="Cleveland" DESTINATION="Paris"
VALUE="61.7" />

<PV ORIGIN="Cleveland" DESTINATION="Philadelphia"
VALUE="7" />
```

```
<PV ORIGIN="Cleveland" DESTINATION="Phoenix"
VALUE="31" />

<PV ORIGIN="Cleveland" DESTINATION="Richardson"
VALUE="15.525" />

<PV ORIGIN="Cleveland" DESTINATION="San Diego"
VALUE="39" />

<PV ORIGIN="Cleveland" DESTINATION="San Francisco"
VALUE="29.755" />

<PV ORIGIN="Cleveland" DESTINATION="San Jose"
VALUE="34" />

<PV ORIGIN="Cleveland" DESTINATION="Santa Clara"
VALUE="29.5" />

<PV ORIGIN="Cleveland" DESTINATION="Seattle"
VALUE="29.7" />

<PV ORIGIN="Cleveland" DESTINATION="Seoul"
VALUE="107.1" />

<PV ORIGIN="Cleveland" DESTINATION="Shanghai"
VALUE="147.3" />

<PV ORIGIN="Cleveland" DESTINATION="Singapore"
VALUE="142.8" />

<PV ORIGIN="Cleveland" DESTINATION="St. Louis"
VALUE="12" />

<PV ORIGIN="Cleveland" DESTINATION="Stockholm"
VALUE="56.4" />

<PV ORIGIN="Cleveland" DESTINATION="Sydney"
VALUE="115.75" />

<PV ORIGIN="Cleveland" DESTINATION="Taipei"
VALUE="96.5" />

<PV ORIGIN="Cleveland" DESTINATION="Tampa"
VALUE="21" />

<PV ORIGIN="Cleveland" DESTINATION="Tokyo"
VALUE="96.9" />

<PV ORIGIN="Cleveland" DESTINATION="Toronto"
VALUE="11.62" />
```

```
<PV ORIGIN="Cleveland" DESTINATION="Vancouver"
VALUE="43.48" />

<PV ORIGIN="Cleveland" DESTINATION="Washington D.C."
VALUE="6" />

<PV ORIGIN="Copenhagen" DESTINATION="Atlanta"
VALUE="60" />

<PV ORIGIN="Copenhagen" DESTINATION="Chicago"
VALUE="62.95" />

<PV ORIGIN="Copenhagen" DESTINATION="Cleveland"
VALUE="59.4" />

<PV ORIGIN="Copenhagen" DESTINATION="Dallas"
VALUE="71.35" />

<PV ORIGIN="Copenhagen" DESTINATION="Denver"
VALUE="74.65" />

<PV ORIGIN="Copenhagen" DESTINATION="Detroit"
VALUE="61.1" />

<PV ORIGIN="Copenhagen" DESTINATION="Elmhurst"
VALUE="63.7" />

<PV ORIGIN="Copenhagen" DESTINATION="Herndon"
VALUE="54.5" />

<PV ORIGIN="Copenhagen" DESTINATION="Houston"
VALUE="66.4" />

<PV ORIGIN="Copenhagen" DESTINATION="Irvine"
VALUE="81.15" />

<PV ORIGIN="Copenhagen" DESTINATION="Kansas City"
VALUE="68.65" />

<PV ORIGIN="Copenhagen" DESTINATION="Los Angeles"
VALUE="82.2" />

<PV ORIGIN="Copenhagen" DESTINATION="Miami"
VALUE="66.65" />

<PV ORIGIN="Copenhagen" DESTINATION="New York"
VALUE="49.875" />

<PV ORIGIN="Copenhagen" DESTINATION="Richardson"
VALUE="71.2" />
```

```
<PV ORIGIN="Copenhagen" DESTINATION="San Francisco"  
VALUE="89.8" />
```

```
<PV ORIGIN="Copenhagen" DESTINATION="Santa Clara"  
VALUE="88.05" />
```

```
<PV ORIGIN="Copenhagen" DESTINATION="Seattle"  
VALUE="88.25" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Amsterdam"  
VALUE="67.45" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Atlanta"  
VALUE="9.915" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Boston"  
VALUE="23.5" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Chicago"  
VALUE="11.725" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Cleveland"  
VALUE="15.325" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Copenhagen"  
VALUE="71.35" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Denver" VALUE="7.5"  
/>
```

```
<PV ORIGIN="Dallas" DESTINATION="Detroit"  
VALUE="14.555" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Elmhurst"  
VALUE="11.8" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Frankfurt"  
VALUE="74.55" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Helsinki"  
VALUE="73.85" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Herndon"  
VALUE="17.325" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Hong Kong"  
VALUE="101.45" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Houston"  
VALUE="3.64" />
```

```
<PV ORIGIN="Dallas" DESTINATION="Irvine"  
VALUE="18.305" />  
  
<PV ORIGIN="Dallas" DESTINATION="Kansas City"  
VALUE="6.035" />  
  
<PV ORIGIN="Dallas" DESTINATION="London"  
VALUE="58.7" />  
  
<PV ORIGIN="Dallas" DESTINATION="Los Angeles"  
VALUE="16.5" />  
  
<PV ORIGIN="Dallas" DESTINATION="Madrid"  
VALUE="71.9" />  
  
<PV ORIGIN="Dallas" DESTINATION="Mexico City"  
VALUE="21.6" />  
  
<PV ORIGIN="Dallas" DESTINATION="Miami" VALUE="14" /  
>  
  
<PV ORIGIN="Dallas" DESTINATION="Milan"  
VALUE="65.65" />  
  
<PV ORIGIN="Dallas" DESTINATION="Milwaukee"  
VALUE="16" />  
  
<PV ORIGIN="Dallas" DESTINATION="Minneapolis"  
VALUE="17" />  
  
<PV ORIGIN="Dallas" DESTINATION="Munich"  
VALUE="72.5" />  
  
<PV ORIGIN="Dallas" DESTINATION="New York"  
VALUE="20.24" />  
  
<PV ORIGIN="Dallas" DESTINATION="Oslo" VALUE="85.8"  
/>  
  
<PV ORIGIN="Dallas" DESTINATION="Paris" VALUE="64.9"  
/>  
  
<PV ORIGIN="Dallas" DESTINATION="Philadelphia"  
VALUE="19.5" />  
  
<PV ORIGIN="Dallas" DESTINATION="Phoenix"  
VALUE="11.5" />  
  
<PV ORIGIN="Dallas" DESTINATION="Richardson"  
VALUE="0.81" />
```

```
<PV ORIGIN="Dallas" DESTINATION="San Diego"
VALUE="16" />

<PV ORIGIN="Dallas" DESTINATION="San Francisco"
VALUE="20.5" />

<PV ORIGIN="Dallas" DESTINATION="San Jose"
VALUE="23.5" />

<PV ORIGIN="Dallas" DESTINATION="Santa Clara"
VALUE="23.495" />

<PV ORIGIN="Dallas" DESTINATION="Seattle"
VALUE="20.5" />

<PV ORIGIN="Dallas" DESTINATION="Seoul"
VALUE="89.55" />

<PV ORIGIN="Dallas" DESTINATION="Shanghai"
VALUE="121.3" />

<PV ORIGIN="Dallas" DESTINATION="Singapore"
VALUE="123.55" />

<PV ORIGIN="Dallas" DESTINATION="St. Louis"
VALUE="19" />

<PV ORIGIN="Dallas" DESTINATION="Stockholm"
VALUE="71.65" />

<PV ORIGIN="Dallas" DESTINATION="Sydney"
VALUE="92.25" />

<PV ORIGIN="Dallas" DESTINATION="Taipei"
VALUE="86.9" />

<PV ORIGIN="Dallas" DESTINATION="Tampa" VALUE="18" /
>

<PV ORIGIN="Dallas" DESTINATION="Tokyo" VALUE="89" /
>

<PV ORIGIN="Dallas" DESTINATION="Toronto"
VALUE="18.725" />

<PV ORIGIN="Dallas" DESTINATION="Vancouver"
VALUE="35.985" />

<PV ORIGIN="Dallas" DESTINATION="Washington D.C."
VALUE="19" />
```

```
<PV ORIGIN="Denver" DESTINATION="Amsterdam"
VALUE="91.45" />

<PV ORIGIN="Denver" DESTINATION="Atlanta" VALUE="18"
/>

<PV ORIGIN="Denver" DESTINATION="Boston"
VALUE="22.5" />

<PV ORIGIN="Denver" DESTINATION="Chicago"
VALUE="11.655" />

<PV ORIGIN="Denver" DESTINATION="Cleveland"
VALUE="15.81" />

<PV ORIGIN="Denver" DESTINATION="Copenhagen"
VALUE="74.65" />

<PV ORIGIN="Denver" DESTINATION="Dallas" VALUE="7.5"
/>

<PV ORIGIN="Denver" DESTINATION="Dallas"
VALUE="11.77" />

<PV ORIGIN="Denver" DESTINATION="Detroit"
VALUE="14.98" />

<PV ORIGIN="Denver" DESTINATION="Elmhurst"
VALUE="12.44" />

<PV ORIGIN="Denver" DESTINATION="Frankfurt"
VALUE="85.3" />

<PV ORIGIN="Denver" DESTINATION="Helsinki"
VALUE="89.8" />

<PV ORIGIN="Denver" DESTINATION="Herndon"
VALUE="19.7" />

<PV ORIGIN="Denver" DESTINATION="Hong Kong"
VALUE="94.2" />

<PV ORIGIN="Denver" DESTINATION="Houston"
VALUE="13.885" />

<PV ORIGIN="Denver" DESTINATION="Irvine"
VALUE="20.56" />

<PV ORIGIN="Denver" DESTINATION="Kansas City"
VALUE="6.98" />
```

```
<PV ORIGIN="Denver" DESTINATION="London"
VALUE="71.85" />

<PV ORIGIN="Denver" DESTINATION="Los Angeles"
VALUE="16.5" />

<PV ORIGIN="Denver" DESTINATION="Madrid"
VALUE="82.15" />

<PV ORIGIN="Denver" DESTINATION="Mexico City"
VALUE="45.875" />

<PV ORIGIN="Denver" DESTINATION="Miami" VALUE="21.5"
/>

<PV ORIGIN="Denver" DESTINATION="Milan"
VALUE="83.05" />

<PV ORIGIN="Denver" DESTINATION="Milwaukee"
VALUE="19.5" />

<PV ORIGIN="Denver" DESTINATION="Minneapolis"
VALUE="14" />

<PV ORIGIN="Denver" DESTINATION="Munich"
VALUE="74.55" />

<PV ORIGIN="Denver" DESTINATION="New York"
VALUE="24.72" />

<PV ORIGIN="Denver" DESTINATION="Oslo" VALUE="89.45"
/>

<PV ORIGIN="Denver" DESTINATION="Paris"
VALUE="85.25" />

<PV ORIGIN="Denver" DESTINATION="Philadelphia"
VALUE="24" />

<PV ORIGIN="Denver" DESTINATION="Phoenix" VALUE="19"
/>

<PV ORIGIN="Denver" DESTINATION="Richardson"
VALUE="11.665" />

<PV ORIGIN="Denver" DESTINATION="San Diego"
VALUE="23" />

<PV ORIGIN="Denver" DESTINATION="San Francisco"
VALUE="13.96" />
```

```
<PV ORIGIN="Denver" DESTINATION="San Jose"
VALUE="28" />

<PV ORIGIN="Denver" DESTINATION="Santa Clara"
VALUE="13.83" />

<PV ORIGIN="Denver" DESTINATION="Seattle"
VALUE="13.5" />

<PV ORIGIN="Denver" DESTINATION="Seoul" VALUE="92.3"
/>

<PV ORIGIN="Denver" DESTINATION="Shanghai"
VALUE="124.6" />

<PV ORIGIN="Denver" DESTINATION="Singapore"
VALUE="118.85" />

<PV ORIGIN="Denver" DESTINATION="St. Louis"
VALUE="33" />

<PV ORIGIN="Denver" DESTINATION="Stockholm"
VALUE="82.4" />

<PV ORIGIN="Denver" DESTINATION="Sydney"
VALUE="103.85" />

<PV ORIGIN="Denver" DESTINATION="Taipei"
VALUE="83.5" />

<PV ORIGIN="Denver" DESTINATION="Tampa" VALUE="26" /
>

<PV ORIGIN="Denver" DESTINATION="Tokyo"
VALUE="72.25" />

<PV ORIGIN="Denver" DESTINATION="Toronto"
VALUE="17.5" />

<PV ORIGIN="Denver" DESTINATION="Vancouver"
VALUE="26.425" />

<PV ORIGIN="Denver" DESTINATION="Washington D.C."
VALUE="24.5" />

<PV ORIGIN="Detroit" DESTINATION="Amsterdam"
VALUE="65.75" />

<PV ORIGIN="Detroit" DESTINATION="Atlanta"
VALUE="11.93" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Boston" VALUE="15"  
>
```

```
<PV ORIGIN="Detroit" DESTINATION="Chicago"  
VALUE="3.61" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Cleveland"  
VALUE="2.69" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Copenhagen"  
VALUE="61.1" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Dallas"  
VALUE="14.75" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Denver"  
VALUE="15.195" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Elmhurst"  
VALUE="3.885" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Frankfurt"  
VALUE="66.15" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Helsinki"  
VALUE="64.3" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Herndon"  
VALUE="6.47" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Hong Kong"  
VALUE="106.75" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Houston"  
VALUE="18.45" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Irvine"  
VALUE="33.55" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Kansas City"  
VALUE="9.065" />
```

```
<PV ORIGIN="Detroit" DESTINATION="London"  
VALUE="51.1" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Los Angeles"  
VALUE="34.98" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Madrid"  
VALUE="62.9" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Mexico City"
VALUE="40.345" />

<PV ORIGIN="Detroit" DESTINATION="Miami"
VALUE="18.66" />

<PV ORIGIN="Detroit" DESTINATION="Milan"
VALUE="55.7" />

<PV ORIGIN="Detroit" DESTINATION="Milwaukee"
VALUE="6" />

<PV ORIGIN="Detroit" DESTINATION="Minneapolis"
VALUE="33" />

<PV ORIGIN="Detroit" DESTINATION="Munich"
VALUE="61.7" />

<PV ORIGIN="Detroit" DESTINATION="New York"
VALUE="9.475" />

<PV ORIGIN="Detroit" DESTINATION="Oslo"
VALUE="74.85" />

<PV ORIGIN="Detroit" DESTINATION="Paris"
VALUE="61.35" />

<PV ORIGIN="Detroit" DESTINATION="Philadelphia"
VALUE="15" />

<PV ORIGIN="Detroit" DESTINATION="Phoenix"
VALUE="30" />

<PV ORIGIN="Detroit" DESTINATION="Richardson"
VALUE="14.45" />

<PV ORIGIN="Detroit" DESTINATION="San Diego"
VALUE="36" />

<PV ORIGIN="Detroit" DESTINATION="San Francisco"
VALUE="29.185" />

<PV ORIGIN="Detroit" DESTINATION="San Jose"
VALUE="35" />

<PV ORIGIN="Detroit" DESTINATION="Santa Clara"
VALUE="28.64" />

<PV ORIGIN="Detroit" DESTINATION="Seattle"
VALUE="28.99" />
```

```
<PV ORIGIN="Detroit" DESTINATION="Seoul"  
VALUE="108.5" />  
  
<PV ORIGIN="Detroit" DESTINATION="Shanghai"  
VALUE="137.3" />  
  
<PV ORIGIN="Detroit" DESTINATION="Singapore"  
VALUE="142.2" />  
  
<PV ORIGIN="Detroit" DESTINATION="St. Louis"  
VALUE="6.5" />  
  
<PV ORIGIN="Detroit" DESTINATION="Stockholm"  
VALUE="57.15" />  
  
<PV ORIGIN="Detroit" DESTINATION="Sydney"  
VALUE="114" />  
  
<PV ORIGIN="Detroit" DESTINATION="Taipei"  
VALUE="96.5" />  
  
<PV ORIGIN="Detroit" DESTINATION="Tampa" VALUE="30"  
/>  
  
<PV ORIGIN="Detroit" DESTINATION="Tokyo" VALUE="97"  
/>  
  
<PV ORIGIN="Detroit" DESTINATION="Toronto"  
VALUE="10.66" />  
  
<PV ORIGIN="Detroit" DESTINATION="Vancouver"  
VALUE="43.92" />  
  
<PV ORIGIN="Detroit" DESTINATION="Washington D.C."  
VALUE="15" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Amsterdam"  
VALUE="68.7" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Atlanta"  
VALUE="13.51" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Chicago"  
VALUE="0.725" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Cleveland"  
VALUE="4.61" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Copenhagen"  
VALUE="63.7" />
```

```
<PV ORIGIN="Elmhurst" DESTINATION="Dallas"
VALUE="11.81" />

<PV ORIGIN="Elmhurst" DESTINATION="Denver"
VALUE="12.48" />

<PV ORIGIN="Elmhurst" DESTINATION="Detroit"
VALUE="3.745" />

<PV ORIGIN="Elmhurst" DESTINATION="Elmhurst"
VALUE="NA" />

<PV ORIGIN="Elmhurst" DESTINATION="Frankfurt"
VALUE="69.5" />

<PV ORIGIN="Elmhurst" DESTINATION="Helsinki"
VALUE="66.5" />

<PV ORIGIN="Elmhurst" DESTINATION="Herndon"
VALUE="8.17" />

<PV ORIGIN="Elmhurst" DESTINATION="Hong Kong"
VALUE="104.85" />

<PV ORIGIN="Elmhurst" DESTINATION="Houston"
VALUE="13.985" />

<PV ORIGIN="Elmhurst" DESTINATION="Irvine"
VALUE="32.49" />

<PV ORIGIN="Elmhurst" DESTINATION="Kansas City"
VALUE="6.5" />

<PV ORIGIN="Elmhurst" DESTINATION="London"
VALUE="52.3" />

<PV ORIGIN="Elmhurst" DESTINATION="Los Angeles"
VALUE="31.9" />

<PV ORIGIN="Elmhurst" DESTINATION="Madrid"
VALUE="64.4" />

<PV ORIGIN="Elmhurst" DESTINATION="Mexico City"
VALUE="32.74" />

<PV ORIGIN="Elmhurst" DESTINATION="Miami"
VALUE="22.035" />

<PV ORIGIN="Elmhurst" DESTINATION="Milan"
VALUE="57.2" />
```

```
<PV ORIGIN="Elmhurst" DESTINATION="Munich"  
VALUE="63.95" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="New York"  
VALUE="11.015" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Oslo"  
VALUE="77.75" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Paris" VALUE="63"  
/>  
  
<PV ORIGIN="Elmhurst" DESTINATION="Richardson"  
VALUE="11.805" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="San Francisco"  
VALUE="26.02" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Santa Clara"  
VALUE="25.39" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Seattle"  
VALUE="26.005" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Seoul"  
VALUE="97.55" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Shanghai"  
VALUE="144.8" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Singapore"  
VALUE="131.9" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Stockholm"  
VALUE="58.8" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Sydney"  
VALUE="118.1" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Taipei"  
VALUE="94.05" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Tokyo"  
VALUE="89.15" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Toronto"  
VALUE="7.33" />  
  
<PV ORIGIN="Elmhurst" DESTINATION="Vancouver"  
VALUE="31.46" />
```

```
<PV ORIGIN="Frankfurt" DESTINATION="Atlanta"
VALUE="65.5" />

<PV ORIGIN="Frankfurt" DESTINATION="Chicago"
VALUE="68.6" />

<PV ORIGIN="Frankfurt" DESTINATION="Cleveland"
VALUE="64.65" />

<PV ORIGIN="Frankfurt" DESTINATION="Dallas"
VALUE="74.55" />

<PV ORIGIN="Frankfurt" DESTINATION="Denver"
VALUE="85.3" />

<PV ORIGIN="Frankfurt" DESTINATION="Detroit"
VALUE="66.15" />

<PV ORIGIN="Frankfurt" DESTINATION="Elmhurst"
VALUE="69.5" />

<PV ORIGIN="Frankfurt" DESTINATION="Herndon"
VALUE="62.1" />

<PV ORIGIN="Frankfurt" DESTINATION="Houston"
VALUE="71.05" />

<PV ORIGIN="Frankfurt" DESTINATION="Irvine"
VALUE="86.55" />

<PV ORIGIN="Frankfurt" DESTINATION="Kansas City"
VALUE="75.25" />

<PV ORIGIN="Frankfurt" DESTINATION="Los Angeles"
VALUE="87.3" />

<PV ORIGIN="Frankfurt" DESTINATION="Miami"
VALUE="76.45" />

<PV ORIGIN="Frankfurt" DESTINATION="New York"
VALUE="65.7" />

<PV ORIGIN="Frankfurt" DESTINATION="Richardson"
VALUE="73.55" />

<PV ORIGIN="Frankfurt" DESTINATION="San Francisco"
VALUE="93.7" />

<PV ORIGIN="Frankfurt" DESTINATION="Santa Clara"
VALUE="91.85" />
```

Appendix B

```
<PV ORIGIN="Frankfurt" DESTINATION="Seattle"
VALUE="95.3" />

<PV ORIGIN="Helsinki" DESTINATION="Atlanta"
VALUE="62.6" />

<PV ORIGIN="Helsinki" DESTINATION="Chicago"
VALUE="63.85" />

<PV ORIGIN="Helsinki" DESTINATION="Cleveland"
VALUE="61.75" />

<PV ORIGIN="Helsinki" DESTINATION="Dallas"
VALUE="73.85" />

<PV ORIGIN="Helsinki" DESTINATION="Denver"
VALUE="89.8" />

<PV ORIGIN="Helsinki" DESTINATION="Detroit"
VALUE="64.3" />

<PV ORIGIN="Helsinki" DESTINATION="Elmhurst"
VALUE="66.5" />

<PV ORIGIN="Helsinki" DESTINATION="Herndon"
VALUE="58" />

<PV ORIGIN="Helsinki" DESTINATION="Houston"
VALUE="68.2" />

<PV ORIGIN="Helsinki" DESTINATION="Irvine"
VALUE="90.9" />

<PV ORIGIN="Helsinki" DESTINATION="Kansas City"
VALUE="69.4" />

<PV ORIGIN="Helsinki" DESTINATION="Los Angeles"
VALUE="87.7" />

<PV ORIGIN="Helsinki" DESTINATION="Miami"
VALUE="68.1" />

<PV ORIGIN="Helsinki" DESTINATION="New York"
VALUE="52.25" />

<PV ORIGIN="Helsinki" DESTINATION="Richardson"
VALUE="74.4" />

<PV ORIGIN="Helsinki" DESTINATION="San Francisco"
VALUE="89.8" />
```

```
<PV ORIGIN="Helsinki" DESTINATION="Santa Clara"  
VALUE="91.6" />
```

```
<PV ORIGIN="Helsinki" DESTINATION="Seattle"  
VALUE="92.4" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Amsterdam"  
VALUE="64.65" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Atlanta"  
VALUE="6.165" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Chicago"  
VALUE="8.105" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Cleveland"  
VALUE="4.66" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Copenhagen"  
VALUE="54.5" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Dallas"  
VALUE="17.645" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Denver"  
VALUE="20.27" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Detroit"  
VALUE="6.48" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Elmhurst"  
VALUE="8.47" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Frankfurt"  
VALUE="62.1" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Helsinki"  
VALUE="58" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Herndon"  
VALUE="NA" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Hong Kong"  
VALUE="109.9" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Houston"  
VALUE="12.815" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Irvine"  
VALUE="28.205" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Kansas City"
VALUE="13.43" />

<PV ORIGIN="Herndon" DESTINATION="London"
VALUE="45.085" />

<PV ORIGIN="Herndon" DESTINATION="Los Angeles"
VALUE="28.29" />

<PV ORIGIN="Herndon" DESTINATION="Madrid"
VALUE="58.2" />

<PV ORIGIN="Herndon" DESTINATION="Mexico City"
VALUE="38.645" />

<PV ORIGIN="Herndon" DESTINATION="Miami"
VALUE="12.63" />

<PV ORIGIN="Herndon" DESTINATION="Milan"
VALUE="55.5" />

<PV ORIGIN="Herndon" DESTINATION="Munich"
VALUE="55.65" />

<PV ORIGIN="Herndon" DESTINATION="New York"
VALUE="3.345" />

<PV ORIGIN="Herndon" DESTINATION="Oslo"
VALUE="75.25" />

<PV ORIGIN="Herndon" DESTINATION="Paris"
VALUE="58.1" />

<PV ORIGIN="Herndon" DESTINATION="Richardson"
VALUE="17.42" />

<PV ORIGIN="Herndon" DESTINATION="San Francisco"
VALUE="34.35" />

<PV ORIGIN="Herndon" DESTINATION="Santa Clara"
VALUE="33.555" />

<PV ORIGIN="Herndon" DESTINATION="Seattle"
VALUE="33.4" />

<PV ORIGIN="Herndon" DESTINATION="Seoul"
VALUE="104.55" />

<PV ORIGIN="Herndon" DESTINATION="Shanghai"
VALUE="140.9" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Singapore"  
VALUE="143.1" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Stockholm"  
VALUE="51.65" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Sydney"  
VALUE="100.75" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Taipei"  
VALUE="96.4" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Tokyo"  
VALUE="95.75" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Toronto"  
VALUE="13.685" />
```

```
<PV ORIGIN="Herndon" DESTINATION="Vancouver"  
VALUE="43.5" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Atlanta"  
VALUE="103.75" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Chicago"  
VALUE="104.85" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Cleveland"  
VALUE="108.05" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Dallas"  
VALUE="101.45" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Denver"  
VALUE="94.2" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Detroit"  
VALUE="106.75" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Elmhurst"  
VALUE="104.85" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Herndon"  
VALUE="109.9" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Houston"  
VALUE="98.4" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Irvine"  
VALUE="82.95" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Kansas City"  
VALUE="99.6" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Los Angeles"  
VALUE="83.75" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Miami"  
VALUE="111" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="New York"  
VALUE="114.55" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Richardson"  
VALUE="101.9" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="San Francisco"  
VALUE="83.95" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Santa Clara"  
VALUE="80.65" />
```

```
<PV ORIGIN="Hong Kong" DESTINATION="Seattle"  
VALUE="89.05" />
```

```
<PV ORIGIN="Houston" DESTINATION="Amsterdam"  
VALUE="69.75" />
```

```
<PV ORIGIN="Houston" DESTINATION="Atlanta"  
VALUE="7.105" />
```

```
<PV ORIGIN="Houston" DESTINATION="Boston"  
VALUE="24.5" />
```

```
<PV ORIGIN="Houston" DESTINATION="Chicago"  
VALUE="14.195" />
```

```
<PV ORIGIN="Houston" DESTINATION="Cleveland"  
VALUE="16.635" />
```

```
<PV ORIGIN="Houston" DESTINATION="Copenhagen"  
VALUE="66.4" />
```

```
<PV ORIGIN="Houston" DESTINATION="Dallas"  
VALUE="3.67" />
```

```
<PV ORIGIN="Houston" DESTINATION="Denver"  
VALUE="13.94" />
```

```
<PV ORIGIN="Houston" DESTINATION="Detroit"  
VALUE="18.285" />
```

```
<PV ORIGIN="Houston" DESTINATION="Elmhurst"  
VALUE="14.03" />
```

```
<PV ORIGIN="Houston" DESTINATION="Frankfurt"  
VALUE="71.05" />
```

```
<PV ORIGIN="Houston" DESTINATION="Helsinki"  
VALUE="68.2" />
```

```
<PV ORIGIN="Houston" DESTINATION="Herndon"  
VALUE="12.475" />
```

```
<PV ORIGIN="Houston" DESTINATION="Hong Kong"  
VALUE="98.4" />
```

```
<PV ORIGIN="Houston" DESTINATION="Irvine"  
VALUE="15.785" />
```

```
<PV ORIGIN="Houston" DESTINATION="Kansas City"  
VALUE="8.54" />
```

```
<PV ORIGIN="Houston" DESTINATION="London"  
VALUE="57.3" />
```

```
<PV ORIGIN="Houston" DESTINATION="Los Angeles"  
VALUE="16.45" />
```

```
<PV ORIGIN="Houston" DESTINATION="Madrid"  
VALUE="72.95" />
```

```
<PV ORIGIN="Houston" DESTINATION="Mexico City"  
VALUE="24.545" />
```

```
<PV ORIGIN="Houston" DESTINATION="Miami"  
VALUE="13.39" />
```

```
<PV ORIGIN="Houston" DESTINATION="Milan"  
VALUE="68.7" />
```

```
<PV ORIGIN="Houston" DESTINATION="Milwaukee"  
VALUE="16" />
```

```
<PV ORIGIN="Houston" DESTINATION="Minneapolis"  
VALUE="19" />
```

```
<PV ORIGIN="Houston" DESTINATION="Munich"  
VALUE="67.85" />
```

```
<PV ORIGIN="Houston" DESTINATION="New York"  
VALUE="15.46" />
```

```
<PV ORIGIN="Houston" DESTINATION="Oslo" VALUE="80.4"  
>
```

```
<PV ORIGIN="Houston" DESTINATION="Paris"  
VALUE="66.75" >
```

```
<PV ORIGIN="Houston" DESTINATION="Philadelphia"  
VALUE="28" >
```

```
<PV ORIGIN="Houston" DESTINATION="Phoenix"  
VALUE="14" >
```

```
<PV ORIGIN="Houston" DESTINATION="Richardson"  
VALUE="3.41" >
```

```
<PV ORIGIN="Houston" DESTINATION="San Diego"  
VALUE="19.5" >
```

```
<PV ORIGIN="Houston" DESTINATION="San Francisco"  
VALUE="23" >
```

```
<PV ORIGIN="Houston" DESTINATION="San Jose"  
VALUE="25" >
```

```
<PV ORIGIN="Houston" DESTINATION="Santa Clara"  
VALUE="21.26" >
```

```
<PV ORIGIN="Houston" DESTINATION="Seattle"  
VALUE="28.705" >
```

```
<PV ORIGIN="Houston" DESTINATION="Seoul"  
VALUE="90.65" >
```

```
<PV ORIGIN="Houston" DESTINATION="Shanghai"  
VALUE="127.55" >
```

```
<PV ORIGIN="Houston" DESTINATION="Singapore"  
VALUE="124.75" >
```

```
<PV ORIGIN="Houston" DESTINATION="St. Louis"  
VALUE="19" >
```

```
<PV ORIGIN="Houston" DESTINATION="Stockholm"  
VALUE="65.7" >
```

```
<PV ORIGIN="Houston" DESTINATION="Sydney"  
VALUE="90.75" >
```

```
<PV ORIGIN="Houston" DESTINATION="Taipei"  
VALUE="84.45" >
```

```
<PV ORIGIN="Houston" DESTINATION="Tampa" VALUE="26"  
/>
```

```
<PV ORIGIN="Houston" DESTINATION="Tokyo"  
VALUE="89.6" />
```

```
<PV ORIGIN="Houston" DESTINATION="Toronto"  
VALUE="21.155" />
```

```
<PV ORIGIN="Houston" DESTINATION="Vancouver"  
VALUE="42.73" />
```

```
<PV ORIGIN="Houston" DESTINATION="Washington D.C."  
VALUE="22" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Amsterdam"  
VALUE="82" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Atlanta"  
VALUE="21.925" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Chicago"  
VALUE="31.24" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Cleveland"  
VALUE="31.575" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Copenhagen"  
VALUE="81.15" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Dallas"  
VALUE="18.45" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Denver"  
VALUE="20.555" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Detroit"  
VALUE="33.53" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Elmhurst"  
VALUE="32.665" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Frankfurt"  
VALUE="86.55" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Helsinki"  
VALUE="90.9" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Herndon"  
VALUE="27.545" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Hong Kong"
VALUE="82.95" />

<PV ORIGIN="Irvine" DESTINATION="Houston"
VALUE="15.815" />

<PV ORIGIN="Irvine" DESTINATION="Irvine" VALUE="NA"
/>

<PV ORIGIN="Irvine" DESTINATION="Kansas City"
VALUE="27.17" />

<PV ORIGIN="Irvine" DESTINATION="London"
VALUE="71.25" />

<PV ORIGIN="Irvine" DESTINATION="Los Angeles"
VALUE="0.905" />

<PV ORIGIN="Irvine" DESTINATION="Madrid"
VALUE="89.75" />

<PV ORIGIN="Irvine" DESTINATION="Mexico City"
VALUE="36.605" />

<PV ORIGIN="Irvine" DESTINATION="Miami"
VALUE="28.24" />

<PV ORIGIN="Irvine" DESTINATION="Milan"
VALUE="80.95" />

<PV ORIGIN="Irvine" DESTINATION="Munich"
VALUE="82.95" />

<PV ORIGIN="Irvine" DESTINATION="New York"
VALUE="30.22" />

<PV ORIGIN="Irvine" DESTINATION="Oslo" VALUE="95.55"
/>

<PV ORIGIN="Irvine" DESTINATION="Paris" VALUE="82.5"
/>

<PV ORIGIN="Irvine" DESTINATION="Richardson"
VALUE="17.995" />

<PV ORIGIN="Irvine" DESTINATION="San Francisco"
VALUE="7.68" />

<PV ORIGIN="Irvine" DESTINATION="Santa Clara"
VALUE="6.635" />
```

```
<PV ORIGIN="Irvine" DESTINATION="Seattle"
VALUE="16.13" />

<PV ORIGIN="Irvine" DESTINATION="Seoul" VALUE="91.7"
/>

<PV ORIGIN="Irvine" DESTINATION="Shanghai"
VALUE="105.05" />

<PV ORIGIN="Irvine" DESTINATION="Singapore"
VALUE="108.2" />

<PV ORIGIN="Irvine" DESTINATION="Stockholm"
VALUE="82.25" />

<PV ORIGIN="Irvine" DESTINATION="Sydney"
VALUE="75.3" />

<PV ORIGIN="Irvine" DESTINATION="Taipei"
VALUE="70.95" />

<PV ORIGIN="Irvine" DESTINATION="Tokyo"
VALUE="71.85" />

<PV ORIGIN="Irvine" DESTINATION="Toronto"
VALUE="40.425" />

<PV ORIGIN="Irvine" DESTINATION="Vancouver"
VALUE="19.26" />

<PV ORIGIN="Kansas City" DESTINATION="Amsterdam"
VALUE="72.2" />

<PV ORIGIN="Kansas City" DESTINATION="Atlanta"
VALUE="14.695" />

<PV ORIGIN="Kansas City" DESTINATION="Boston"
VALUE="18.5" />

<PV ORIGIN="Kansas City" DESTINATION="Chicago"
VALUE="6.055" />

<PV ORIGIN="Kansas City" DESTINATION="Cleveland"
VALUE="9.575" />

<PV ORIGIN="Kansas City" DESTINATION="Copenhagen"
VALUE="68.65" />

<PV ORIGIN="Kansas City" DESTINATION="Dallas"
VALUE="5.915" />
```

```
<PV ORIGIN="Kansas City" DESTINATION="Denver"
VALUE="6.885" />

<PV ORIGIN="Kansas City" DESTINATION="Detroit"
VALUE="8.725" />

<PV ORIGIN="Kansas City" DESTINATION="Elmhurst"
VALUE="6.405" />

<PV ORIGIN="Kansas City" DESTINATION="Frankfurt"
VALUE="75.25" />

<PV ORIGIN="Kansas City" DESTINATION="Helsinki"
VALUE="69.4" />

<PV ORIGIN="Kansas City" DESTINATION="Herndon"
VALUE="13.195" />

<PV ORIGIN="Kansas City" DESTINATION="Hong Kong"
VALUE="99.6" />

<PV ORIGIN="Kansas City" DESTINATION="Houston"
VALUE="8.325" />

<PV ORIGIN="Kansas City" DESTINATION="Irvine"
VALUE="26.985" />

<PV ORIGIN="Kansas City" DESTINATION="London"
VALUE="59.75" />

<PV ORIGIN="Kansas City" DESTINATION="Los Angeles"
VALUE="25.915" />

<PV ORIGIN="Kansas City" DESTINATION="Madrid"
VALUE="71.65" />

<PV ORIGIN="Kansas City" DESTINATION="Mexico City"
VALUE="27.11" />

<PV ORIGIN="Kansas City" DESTINATION="Miami"
VALUE="20.995" />

<PV ORIGIN="Kansas City" DESTINATION="Milan"
VALUE="65.85" />

<PV ORIGIN="Kansas City" DESTINATION="Milwaukee"
VALUE="15" />

<PV ORIGIN="Kansas City" DESTINATION="Minneapolis"
VALUE="14" />
```

```
<PV ORIGIN="Kansas City" DESTINATION="Munich"  
VALUE="69.05" />  
  
<PV ORIGIN="Kansas City" DESTINATION="New York"  
VALUE="15.69" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Oslo"  
VALUE="82.2" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Paris"  
VALUE="69.15" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Philadelphia"  
VALUE="18.5" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Phoenix"  
VALUE="22.5" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Richardson"  
VALUE="6.005" />  
  
<PV ORIGIN="Kansas City" DESTINATION="San Diego"  
VALUE="27" />  
  
<PV ORIGIN="Kansas City" DESTINATION="San Francisco"  
VALUE="20.255" />  
  
<PV ORIGIN="Kansas City" DESTINATION="San Jose"  
VALUE="31" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Santa Clara"  
VALUE="19.805" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Seattle"  
VALUE="20.375" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Seoul"  
VALUE="94.95" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Shanghai"  
VALUE="134.2" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Singapore"  
VALUE="123.9" />  
  
<PV ORIGIN="Kansas City" DESTINATION="St. Louis"  
VALUE="14" />  
  
<PV ORIGIN="Kansas City" DESTINATION="Stockholm"  
VALUE="67.5" />
```

Appendix B

```
<PV ORIGIN="Kansas City" DESTINATION="Sydney"
VALUE="113.7" />

<PV ORIGIN="Kansas City" DESTINATION="Taipei"
VALUE="88.2" />

<PV ORIGIN="Kansas City" DESTINATION="Tampa"
VALUE="20" />

<PV ORIGIN="Kansas City" DESTINATION="Tokyo"
VALUE="89.9" />

<PV ORIGIN="Kansas City" DESTINATION="Toronto"
VALUE="13.56" />

<PV ORIGIN="Kansas City" DESTINATION="Vancouver"
VALUE="31.535" />

<PV ORIGIN="Kansas City" DESTINATION="Washington
D.C." VALUE="20.5" />

<PV ORIGIN="London" DESTINATION="Atlanta"
VALUE="49.795" />

<PV ORIGIN="London" DESTINATION="Chicago"
VALUE="52.35" />

<PV ORIGIN="London" DESTINATION="Cleveland"
VALUE="49.125" />

<PV ORIGIN="London" DESTINATION="Dallas"
VALUE="58.7" />

<PV ORIGIN="London" DESTINATION="Denver"
VALUE="71.85" />

<PV ORIGIN="London" DESTINATION="Detroit"
VALUE="51.1" />

<PV ORIGIN="London" DESTINATION="Elmhurst"
VALUE="52.3" />

<PV ORIGIN="London" DESTINATION="Herndon"
VALUE="45.085" />

<PV ORIGIN="London" DESTINATION="Houston"
VALUE="57.3" />

<PV ORIGIN="London" DESTINATION="Irvine"
VALUE="71.25" />
```

```
<PV ORIGIN="London" DESTINATION="Kansas City"
VALUE="59.75" />

<PV ORIGIN="London" DESTINATION="Los Angeles"
VALUE="71.8" />

<PV ORIGIN="London" DESTINATION="Miami"
VALUE="62.95" />

<PV ORIGIN="London" DESTINATION="New York"
VALUE="45.955" />

<PV ORIGIN="London" DESTINATION="Richardson"
VALUE="59.3" />

<PV ORIGIN="London" DESTINATION="San Francisco"
VALUE="77.65" />

<PV ORIGIN="London" DESTINATION="Santa Clara"
VALUE="77.15" />

<PV ORIGIN="London" DESTINATION="Seattle"
VALUE="81.4" />

<PV ORIGIN="Los Angeles" DESTINATION="Amsterdam"
VALUE="83.3" />

<PV ORIGIN="Los Angeles" DESTINATION="Atlanta"
VALUE="22.46" />

<PV ORIGIN="Los Angeles" DESTINATION="Boston"
VALUE="39.5" />

<PV ORIGIN="Los Angeles" DESTINATION="Chicago"
VALUE="28.5" />

<PV ORIGIN="Los Angeles" DESTINATION="Cleveland"
VALUE="33.2" />

<PV ORIGIN="Los Angeles" DESTINATION="Copenhagen"
VALUE="82.2" />

<PV ORIGIN="Los Angeles" DESTINATION="Dallas"
VALUE="16.5" />

<PV ORIGIN="Los Angeles" DESTINATION="Denver"
VALUE="16.5" />

<PV ORIGIN="Los Angeles" DESTINATION="Detroit"
VALUE="34.66" />
```

Appendix B

```
<PV ORIGIN="Los Angeles" DESTINATION="Elmhurst"  
VALUE="31.78" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Frankfurt"  
VALUE="87.3" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Helsinki"  
VALUE="87.7" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Herndon"  
VALUE="27.93" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Hong Kong"  
VALUE="83.75" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Houston"  
VALUE="16.2" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Irvine"  
VALUE="0.735" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Kansas City"  
VALUE="25.9" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="London"  
VALUE="71.8" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Madrid"  
VALUE="90.05" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Mexico City"  
VALUE="37.315" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Miami"  
VALUE="28.905" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Milan"  
VALUE="80.6" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Milwaukee"  
VALUE="35.5" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Minneapolis"  
VALUE="27.5" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="Munich"  
VALUE="83.45" />  
  
<PV ORIGIN="Los Angeles" DESTINATION="New York"  
VALUE="31.055" />
```

```
<PV ORIGIN="Los Angeles" DESTINATION="Oslo"
VALUE="96.8" />

<PV ORIGIN="Los Angeles" DESTINATION="Paris"
VALUE="82.55" />

<PV ORIGIN="Los Angeles" DESTINATION="Philadelphia"
VALUE="36.5" />

<PV ORIGIN="Los Angeles" DESTINATION="Phoenix"
VALUE="5" />

<PV ORIGIN="Los Angeles" DESTINATION="Richardson"
VALUE="18.525" />

<PV ORIGIN="Los Angeles" DESTINATION="San Diego"
VALUE="4" />

<PV ORIGIN="Los Angeles" DESTINATION="San Francisco"
VALUE="4.5" />

<PV ORIGIN="Los Angeles" DESTINATION="San Jose"
VALUE="8.5" />

<PV ORIGIN="Los Angeles" DESTINATION="Santa Clara"
VALUE="7.145" />

<PV ORIGIN="Los Angeles" DESTINATION="Seattle"
VALUE="12.5" />

<PV ORIGIN="Los Angeles" DESTINATION="Seoul"
VALUE="93.25" />

<PV ORIGIN="Los Angeles" DESTINATION="Shanghai"
VALUE="109.15" />

<PV ORIGIN="Los Angeles" DESTINATION="Singapore"
VALUE="107.2" />

<PV ORIGIN="Los Angeles" DESTINATION="St. Louis"
VALUE="33.5" />

<PV ORIGIN="Los Angeles" DESTINATION="Stockholm"
VALUE="86.75" />

<PV ORIGIN="Los Angeles" DESTINATION="Sydney"
VALUE="78.05" />

<PV ORIGIN="Los Angeles" DESTINATION="Taipei"
VALUE="70.3" />
```

Appendix B

```
<PV ORIGIN="Los Angeles" DESTINATION="Tampa"  
VALUE="33" />
```

```
<PV ORIGIN="Los Angeles" DESTINATION="Tokyo"  
VALUE="72.65" />
```

```
<PV ORIGIN="Los Angeles" DESTINATION="Toronto"  
VALUE="32.5" />
```

```
<PV ORIGIN="Los Angeles" DESTINATION="Vancouver"  
VALUE="18.15" />
```

```
<PV ORIGIN="Los Angeles" DESTINATION="Washington  
D.C." VALUE="33" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Atlanta"  
VALUE="63.35" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Chicago"  
VALUE="63.8" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Cleveland"  
VALUE="60.25" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Dallas"  
VALUE="71.9" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Denver"  
VALUE="82.15" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Detroit"  
VALUE="62.9" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Elmhurst"  
VALUE="64.4" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Herndon"  
VALUE="58.2" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Houston"  
VALUE="72.95" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Irvine"  
VALUE="89.75" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Kansas City"  
VALUE="71.65" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Los Angeles"  
VALUE="90.05" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Miami" VALUE="80.5"  
/>
```

```
<PV ORIGIN="Madrid" DESTINATION="New York"  
VALUE="62.7" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Richardson"  
VALUE="71.45" />
```

```
<PV ORIGIN="Madrid" DESTINATION="San Francisco"  
VALUE="93.45" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Santa Clara"  
VALUE="91.25" />
```

```
<PV ORIGIN="Madrid" DESTINATION="Seattle"  
VALUE="91.95" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Atlanta"  
VALUE="37.39" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Chicago"  
VALUE="32.925" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Cleveland"  
VALUE="39.965" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Dallas"  
VALUE="21.6" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Denver"  
VALUE="45.875" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Detroit"  
VALUE="40.345" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Elmhurst"  
VALUE="32.74" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Herndon"  
VALUE="38.645" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Houston"  
VALUE="24.545" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Irvine"  
VALUE="36.605" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Kansas City"  
VALUE="27.11" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Los Angeles"  
VALUE="37.315" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Miami"  
VALUE="37.225" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="New York"  
VALUE="41.5" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Richardson"  
VALUE="21.62" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="San Francisco"  
VALUE="47.375" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Santa Clara"  
VALUE="46.925" />
```

```
<PV ORIGIN="Mexico City" DESTINATION="Seattle"  
VALUE="50.8" />
```

```
<PV ORIGIN="Miami" DESTINATION="Amsterdam"  
VALUE="82.35" />
```

```
<PV ORIGIN="Miami" DESTINATION="Atlanta" VALUE="6.5"  
/>
```

```
<PV ORIGIN="Miami" DESTINATION="Boston" VALUE="19.5"  
/>
```

```
<PV ORIGIN="Miami" DESTINATION="Chicago"  
VALUE="20.585" />
```

```
<PV ORIGIN="Miami" DESTINATION="Cleveland"  
VALUE="16.69" />
```

```
<PV ORIGIN="Miami" DESTINATION="Copenhagen"  
VALUE="66.65" />
```

```
<PV ORIGIN="Miami" DESTINATION="Dallas" VALUE="14" /  
>
```

```
<PV ORIGIN="Miami" DESTINATION="Denver" VALUE="21.5"  
/>
```

```
<PV ORIGIN="Miami" DESTINATION="Detroit"  
VALUE="18.28" />
```

```
<PV ORIGIN="Miami" DESTINATION="Elmhurst"  
VALUE="22.03" />
```

```
<PV ORIGIN="Miami" DESTINATION="Frankfurt"  
VALUE="76.45" />  
  
<PV ORIGIN="Miami" DESTINATION="Helsinki"  
VALUE="68.1" />  
  
<PV ORIGIN="Miami" DESTINATION="Herndon"  
VALUE="12.335" />  
  
<PV ORIGIN="Miami" DESTINATION="Hong Kong"  
VALUE="111" />  
  
<PV ORIGIN="Miami" DESTINATION="Houston"  
VALUE="13.23" />  
  
<PV ORIGIN="Miami" DESTINATION="Irvine"  
VALUE="28.05" />  
  
<PV ORIGIN="Miami" DESTINATION="Kansas City"  
VALUE="20.93" />  
  
<PV ORIGIN="Miami" DESTINATION="London"  
VALUE="62.95" />  
  
<PV ORIGIN="Miami" DESTINATION="Los Angeles"  
VALUE="28.875" />  
  
<PV ORIGIN="Miami" DESTINATION="Madrid" VALUE="80.5"  
/>  
  
<PV ORIGIN="Miami" DESTINATION="Mexico City"  
VALUE="37.225" />  
  
<PV ORIGIN="Miami" DESTINATION="Milan" VALUE="77.9"  
/>  
  
<PV ORIGIN="Miami" DESTINATION="Milwaukee"  
VALUE="32" />  
  
<PV ORIGIN="Miami" DESTINATION="Minneapolis"  
VALUE="23" />  
  
<PV ORIGIN="Miami" DESTINATION="Munich"  
VALUE="67.65" />  
  
<PV ORIGIN="Miami" DESTINATION="New York"  
VALUE="15.21" />  
  
<PV ORIGIN="Miami" DESTINATION="Oslo" VALUE="80.25"  
/>
```

```
<PV ORIGIN="Miami" DESTINATION="Paris" VALUE="77.9"
/>

<PV ORIGIN="Miami" DESTINATION="Philadelphia"
VALUE="16" />

<PV ORIGIN="Miami" DESTINATION="Phoenix"
VALUE="25.5" />

<PV ORIGIN="Miami" DESTINATION="Richardson"
VALUE="16.055" />

<PV ORIGIN="Miami" DESTINATION="San Diego"
VALUE="40" />

<PV ORIGIN="Miami" DESTINATION="San Francisco"
VALUE="35.5" />

<PV ORIGIN="Miami" DESTINATION="San Jose" VALUE="44"
/>

<PV ORIGIN="Miami" DESTINATION="Santa Clara"
VALUE="34.24" />

<PV ORIGIN="Miami" DESTINATION="Seattle"
VALUE="34.5" />

<PV ORIGIN="Miami" DESTINATION="Seoul"
VALUE="107.65" />

<PV ORIGIN="Miami" DESTINATION="Shanghai"
VALUE="132.9" />

<PV ORIGIN="Miami" DESTINATION="Singapore"
VALUE="136.4" />

<PV ORIGIN="Miami" DESTINATION="St. Louis"
VALUE="21" />

<PV ORIGIN="Miami" DESTINATION="Stockholm"
VALUE="64.85" />

<PV ORIGIN="Miami" DESTINATION="Sydney"
VALUE="102.55" />

<PV ORIGIN="Miami" DESTINATION="Taipei" VALUE="96.9"
/>

<PV ORIGIN="Miami" DESTINATION="Tampa" VALUE="15" />

<PV ORIGIN="Miami" DESTINATION="Tokyo" VALUE="98.3"
/>
```

```
<PV ORIGIN="Miami" DESTINATION="Toronto" VALUE="23"
/>

<PV ORIGIN="Miami" DESTINATION="Vancouver"
VALUE="50.2" />

<PV ORIGIN="Miami" DESTINATION="Washington D.C."
VALUE="16" />

<PV ORIGIN="Milan" DESTINATION="Atlanta"
VALUE="61.05" />

<PV ORIGIN="Milan" DESTINATION="Chicago"
VALUE="57.35" />

<PV ORIGIN="Milan" DESTINATION="Cleveland"
VALUE="55.5" />

<PV ORIGIN="Milan" DESTINATION="Dallas"
VALUE="65.65" />

<PV ORIGIN="Milan" DESTINATION="Denver"
VALUE="83.05" />

<PV ORIGIN="Milan" DESTINATION="Detroit"
VALUE="55.7" />

<PV ORIGIN="Milan" DESTINATION="Elmhurst"
VALUE="57.2" />

<PV ORIGIN="Milan" DESTINATION="Herndon"
VALUE="55.5" />

<PV ORIGIN="Milan" DESTINATION="Houston"
VALUE="68.7" />

<PV ORIGIN="Milan" DESTINATION="Irvine"
VALUE="80.95" />

<PV ORIGIN="Milan" DESTINATION="Kansas City"
VALUE="65.85" />

<PV ORIGIN="Milan" DESTINATION="Los Angeles"
VALUE="80.6" />

<PV ORIGIN="Milan" DESTINATION="Miami" VALUE="77.9"
/>

<PV ORIGIN="Milan" DESTINATION="New York"
VALUE="56.3" />
```

Appendix B

```
<PV ORIGIN="Milan" DESTINATION="Richardson"
VALUE="65.45" />

<PV ORIGIN="Milan" DESTINATION="San Francisco"
VALUE="85.45" />

<PV ORIGIN="Milan" DESTINATION="Santa Clara"
VALUE="83.45" />

<PV ORIGIN="Milan" DESTINATION="Seattle"
VALUE="87.9" />

<PV ORIGIN="Milwaukee" DESTINATION="Atlanta"
VALUE="17" />

<PV ORIGIN="Milwaukee" DESTINATION="Boston"
VALUE="16.5" />

<PV ORIGIN="Milwaukee" DESTINATION="Chicago"
VALUE="4" />

<PV ORIGIN="Milwaukee" DESTINATION="Cleveland"
VALUE="16" />

<PV ORIGIN="Milwaukee" DESTINATION="Dallas"
VALUE="16" />

<PV ORIGIN="Milwaukee" DESTINATION="Denver"
VALUE="19.5" />

<PV ORIGIN="Milwaukee" DESTINATION="Detroit"
VALUE="6" />

<PV ORIGIN="Milwaukee" DESTINATION="Houston"
VALUE="16" />

<PV ORIGIN="Milwaukee" DESTINATION="Kansas City"
VALUE="15" />

<PV ORIGIN="Milwaukee" DESTINATION="Los Angeles"
VALUE="35.5" />

<PV ORIGIN="Milwaukee" DESTINATION="Miami"
VALUE="32" />

<PV ORIGIN="Milwaukee" DESTINATION="Milwaukee"
VALUE="0" />

<PV ORIGIN="Milwaukee" DESTINATION="Minneapolis"
VALUE="21" />
```

```
<PV ORIGIN="Milwaukee" DESTINATION="NYC"
VALUE="15.5" />

<PV ORIGIN="Milwaukee" DESTINATION="Philadelphia"
VALUE="19" />

<PV ORIGIN="Milwaukee" DESTINATION="Phoenix"
VALUE="23.5" />

<PV ORIGIN="Milwaukee" DESTINATION="San Diego"
VALUE="36" />

<PV ORIGIN="Milwaukee" DESTINATION="San Francisco"
VALUE="27.5" />

<PV ORIGIN="Milwaukee" DESTINATION="San Jose"
VALUE="32" />

<PV ORIGIN="Milwaukee" DESTINATION="Seattle"
VALUE="30.5" />

<PV ORIGIN="Milwaukee" DESTINATION="St. Louis"
VALUE="5.5" />

<PV ORIGIN="Milwaukee" DESTINATION="Tampa"
VALUE="23" />

<PV ORIGIN="Milwaukee" DESTINATION="Washington D.C."
VALUE="16.5" />

<PV ORIGIN="Minneapolis" DESTINATION="Atlanta"
VALUE="16" />

<PV ORIGIN="Minneapolis" DESTINATION="Boston"
VALUE="15" />

<PV ORIGIN="Minneapolis" DESTINATION="Chicago"
VALUE="4.5" />

<PV ORIGIN="Minneapolis" DESTINATION="Cleveland"
VALUE="18" />

<PV ORIGIN="Minneapolis" DESTINATION="Dallas"
VALUE="17" />

<PV ORIGIN="Minneapolis" DESTINATION="Denver"
VALUE="14" />

<PV ORIGIN="Minneapolis" DESTINATION="Detroit"
VALUE="33" />
```

```
<PV ORIGIN="Minneapolis" DESTINATION="Houston"
VALUE="19" />

<PV ORIGIN="Minneapolis" DESTINATION="Kansas City"
VALUE="14" />

<PV ORIGIN="Minneapolis" DESTINATION="Los Angeles"
VALUE="27.5" />

<PV ORIGIN="Minneapolis" DESTINATION="Miami"
VALUE="23" />

<PV ORIGIN="Minneapolis" DESTINATION="Milwaukee"
VALUE="21" />

<PV ORIGIN="Minneapolis" DESTINATION="New York"
VALUE="17" />

<PV ORIGIN="Minneapolis" DESTINATION="Philadelphia"
VALUE="18" />

<PV ORIGIN="Minneapolis" DESTINATION="Phoenix"
VALUE="27" />

<PV ORIGIN="Minneapolis" DESTINATION="San Diego"
VALUE="33" />

<PV ORIGIN="Minneapolis" DESTINATION="San Francisco"
VALUE="28.5" />

<PV ORIGIN="Minneapolis" DESTINATION="San Jose"
VALUE="34" />

<PV ORIGIN="Minneapolis" DESTINATION="Seattle"
VALUE="29.5" />

<PV ORIGIN="Minneapolis" DESTINATION="St. Louis"
VALUE="31" />

<PV ORIGIN="Minneapolis" DESTINATION="Tampa"
VALUE="20" />

<PV ORIGIN="Minneapolis" DESTINATION="Toronto"
VALUE="9.5" />

<PV ORIGIN="Minneapolis" DESTINATION="Washington
D.C." VALUE="16" />

<PV ORIGIN="Munich" DESTINATION="Atlanta"
VALUE="60.9" />
```

```
<PV ORIGIN="Munich" DESTINATION="Chicago"
VALUE="63.95" />

<PV ORIGIN="Munich" DESTINATION="Cleveland"
VALUE="59.85" />

<PV ORIGIN="Munich" DESTINATION="Dallas"
VALUE="72.5" />

<PV ORIGIN="Munich" DESTINATION="Denver"
VALUE="74.55" />

<PV ORIGIN="Munich" DESTINATION="Detroit"
VALUE="61.7" />

<PV ORIGIN="Munich" DESTINATION="Elmhurst"
VALUE="63.95" />

<PV ORIGIN="Munich" DESTINATION="Herndon"
VALUE="55.65" />

<PV ORIGIN="Munich" DESTINATION="Houston"
VALUE="67.85" />

<PV ORIGIN="Munich" DESTINATION="Irvine"
VALUE="82.95" />

<PV ORIGIN="Munich" DESTINATION="Kansas City"
VALUE="69.05" />

<PV ORIGIN="Munich" DESTINATION="Los Angeles"
VALUE="83.45" />

<PV ORIGIN="Munich" DESTINATION="Miami"
VALUE="67.65" />

<PV ORIGIN="Munich" DESTINATION="New York"
VALUE="54" />

<PV ORIGIN="Munich" DESTINATION="Richardson"
VALUE="72.35" />

<PV ORIGIN="Munich" DESTINATION="San Francisco"
VALUE="89.25" />

<PV ORIGIN="Munich" DESTINATION="Santa Clara"
VALUE="87.85" />

<PV ORIGIN="Munich" DESTINATION="Seattle"
VALUE="88.8" />
```

Appendix B

```
<PV ORIGIN="New York" DESTINATION="Amsterdam"
VALUE="67.5" />

<PV ORIGIN="New York" DESTINATION="Atlanta"
VALUE="8.895" />

<PV ORIGIN="New York" DESTINATION="Boston" VALUE="3"
/>

<PV ORIGIN="New York" DESTINATION="Chicago"
VALUE="12.31" />

<PV ORIGIN="New York" DESTINATION="Cleveland"
VALUE="8.5" />

<PV ORIGIN="New York" DESTINATION="Copenhagen"
VALUE="49.875" />

<PV ORIGIN="New York" DESTINATION="Dallas"
VALUE="20.455" />

<PV ORIGIN="New York" DESTINATION="Denver"
VALUE="25.295" />

<PV ORIGIN="New York" DESTINATION="Detroit"
VALUE="9.425" />

<PV ORIGIN="New York" DESTINATION="Elmhurst"
VALUE="11.165" />

<PV ORIGIN="New York" DESTINATION="Frankfurt"
VALUE="65.7" />

<PV ORIGIN="New York" DESTINATION="Helsinki"
VALUE="52.25" />

<PV ORIGIN="New York" DESTINATION="Herndon"
VALUE="3.4" />

<PV ORIGIN="New York" DESTINATION="Hong Kong"
VALUE="114.55" />

<PV ORIGIN="New York" DESTINATION="Houston"
VALUE="15.6" />

<PV ORIGIN="New York" DESTINATION="Irvine"
VALUE="30.92" />

<PV ORIGIN="New York" DESTINATION="Kansas City"
VALUE="15.97" />
```

```
<PV ORIGIN="New York" DESTINATION="London"
VALUE="45.955" />

<PV ORIGIN="New York" DESTINATION="Los Angeles"
VALUE="31.455" />

<PV ORIGIN="New York" DESTINATION="Madrid"
VALUE="62.7" />

<PV ORIGIN="New York" DESTINATION="Mexico City"
VALUE="41.5" />

<PV ORIGIN="New York" DESTINATION="Miami"
VALUE="15.44" />

<PV ORIGIN="New York" DESTINATION="Milan"
VALUE="56.3" />

<PV ORIGIN="New York" DESTINATION="Milwaukee"
VALUE="15.5" />

<PV ORIGIN="New York" DESTINATION="Minneapolis"
VALUE="17" />

<PV ORIGIN="New York" DESTINATION="Munich"
VALUE="54" />

<PV ORIGIN="New York" DESTINATION="Oslo"
VALUE="68.05" />

<PV ORIGIN="New York" DESTINATION="Paris"
VALUE="61.7" />

<PV ORIGIN="New York" DESTINATION="Philadelphia"
VALUE="1.5" />

<PV ORIGIN="New York" DESTINATION="Phoenix"
VALUE="29.5" />

<PV ORIGIN="New York" DESTINATION="Richardson"
VALUE="20.18" />

<PV ORIGIN="New York" DESTINATION="San Diego"
VALUE="35" />

<PV ORIGIN="New York" DESTINATION="San Francisco"
VALUE="32.5" />

<PV ORIGIN="New York" DESTINATION="San Jose"
VALUE="41" />
```

```
<PV ORIGIN="New York" DESTINATION="Santa Clara"  
VALUE="36.73" />
```

```
<PV ORIGIN="New York" DESTINATION="Seattle"  
VALUE="35" />
```

```
<PV ORIGIN="New York" DESTINATION="Seoul"  
VALUE="107.25" />
```

```
<PV ORIGIN="New York" DESTINATION="Shanghai"  
VALUE="136.1" />
```

```
<PV ORIGIN="New York" DESTINATION="Singapore"  
VALUE="139.65" />
```

```
<PV ORIGIN="New York" DESTINATION="St. Louis"  
VALUE="17" />
```

```
<PV ORIGIN="New York" DESTINATION="Stockholm"  
VALUE="47.635" />
```

```
<PV ORIGIN="New York" DESTINATION="Sydney"  
VALUE="106.6" />
```

```
<PV ORIGIN="New York" DESTINATION="Taipei"  
VALUE="99.1" />
```

```
<PV ORIGIN="New York" DESTINATION="Tampa"  
VALUE="19.5" />
```

```
<PV ORIGIN="New York" DESTINATION="Tokyo"  
VALUE="98.55" />
```

```
<PV ORIGIN="New York" DESTINATION="Toronto"  
VALUE="6" />
```

```
<PV ORIGIN="New York" DESTINATION="Vancouver"  
VALUE="46.045" />
```

```
<PV ORIGIN="New York" DESTINATION="Washington D.C."  
VALUE="3" />
```

```
<PV ORIGIN="Oslo" DESTINATION="Atlanta" VALUE="74" /  
>
```

```
<PV ORIGIN="Oslo" DESTINATION="Chicago"  
VALUE="77.25" />
```

```
<PV ORIGIN="Oslo" DESTINATION="Cleveland"  
VALUE="73.2" />
```

```
<PV ORIGIN="Oslo" DESTINATION="Dallas" VALUE="85.8"
/>

<PV ORIGIN="Oslo" DESTINATION="Denver" VALUE="89.45"
/>

<PV ORIGIN="Oslo" DESTINATION="Detroit"
VALUE="74.85" />

<PV ORIGIN="Oslo" DESTINATION="Elmhurst"
VALUE="77.75" />

<PV ORIGIN="Oslo" DESTINATION="Herndon"
VALUE="75.25" />

<PV ORIGIN="Oslo" DESTINATION="Houston" VALUE="80.4"
/>

<PV ORIGIN="Oslo" DESTINATION="Irvine" VALUE="95.55"
/>

<PV ORIGIN="Oslo" DESTINATION="Kansas City"
VALUE="82.2" />

<PV ORIGIN="Oslo" DESTINATION="Los Angeles"
VALUE="96.8" />

<PV ORIGIN="Oslo" DESTINATION="Miami" VALUE="80.25"
/>

<PV ORIGIN="Oslo" DESTINATION="New York"
VALUE="68.05" />

<PV ORIGIN="Oslo" DESTINATION="Richardson"
VALUE="85.7" />

<PV ORIGIN="Oslo" DESTINATION="San Francisco"
VALUE="103.45" />

<PV ORIGIN="Oslo" DESTINATION="Santa Clara"
VALUE="101.2" />

<PV ORIGIN="Oslo" DESTINATION="Seattle"
VALUE="102.2" />

<PV ORIGIN="Paris" DESTINATION="Atlanta"
VALUE="60.25" />

<PV ORIGIN="Paris" DESTINATION="Chicago"
VALUE="63.9" />
```

```
<PV ORIGIN="Paris" DESTINATION="Cleveland"
VALUE="61.7" />

<PV ORIGIN="Paris" DESTINATION="Dallas" VALUE="64.9"
/>

<PV ORIGIN="Paris" DESTINATION="Denver"
VALUE="85.25" />

<PV ORIGIN="Paris" DESTINATION="Detroit"
VALUE="61.35" />

<PV ORIGIN="Paris" DESTINATION="Elmhurst" VALUE="63"
/>

<PV ORIGIN="Paris" DESTINATION="Herndon"
VALUE="58.1" />

<PV ORIGIN="Paris" DESTINATION="Houston"
VALUE="66.75" />

<PV ORIGIN="Paris" DESTINATION="Irvine" VALUE="82.5"
/>

<PV ORIGIN="Paris" DESTINATION="Kansas City"
VALUE="69.15" />

<PV ORIGIN="Paris" DESTINATION="Los Angeles"
VALUE="82.55" />

<PV ORIGIN="Paris" DESTINATION="Miami" VALUE="77.9"
/>

<PV ORIGIN="Paris" DESTINATION="New York"
VALUE="61.7" />

<PV ORIGIN="Paris" DESTINATION="Richardson"
VALUE="64.95" />

<PV ORIGIN="Paris" DESTINATION="San Francisco"
VALUE="87.65" />

<PV ORIGIN="Paris" DESTINATION="Santa Clara"
VALUE="86.45" />

<PV ORIGIN="Paris" DESTINATION="Seattle"
VALUE="90.25" />

<PV ORIGIN="Philadelphia" DESTINATION="Atlanta"
VALUE="10" />
```

```
<PV ORIGIN="Philadelphia" DESTINATION="Boston"
VALUE="4" />

<PV ORIGIN="Philadelphia" DESTINATION="Chicago"
VALUE="14" />

<PV ORIGIN="Philadelphia" DESTINATION="Cleveland"
VALUE="7" />

<PV ORIGIN="Philadelphia" DESTINATION="Dallas"
VALUE="19.5" />

<PV ORIGIN="Philadelphia" DESTINATION="Denver"
VALUE="29.5" />

<PV ORIGIN="Philadelphia" DESTINATION="Detroit"
VALUE="15" />

<PV ORIGIN="Philadelphia" DESTINATION="Houston"
VALUE="28" />

<PV ORIGIN="Philadelphia" DESTINATION="Kansas City"
VALUE="18.5" />

<PV ORIGIN="Philadelphia" DESTINATION="Los Angeles"
VALUE="36.5" />

<PV ORIGIN="Philadelphia" DESTINATION="Miami"
VALUE="16" />

<PV ORIGIN="Philadelphia" DESTINATION="Milwaukee"
VALUE="19" />

<PV ORIGIN="Philadelphia" DESTINATION="Minneapolis"
VALUE="17.5" />

<PV ORIGIN="Philadelphia" DESTINATION="New York"
VALUE="1.5" />

<PV ORIGIN="Philadelphia" DESTINATION="Philadelphia"
VALUE="0" />

<PV ORIGIN="Philadelphia" DESTINATION="Philadelphia"
VALUE="PHY" />

<PV ORIGIN="Philadelphia" DESTINATION="Phoenix"
VALUE="31" />

<PV ORIGIN="Philadelphia" DESTINATION="San Diego"
VALUE="35.5" />
```

```
<PV ORIGIN="Philadelphia" DESTINATION="San
Francisco" VALUE="31.5" />

<PV ORIGIN="Philadelphia" DESTINATION="San Jose"
VALUE="44" />

<PV ORIGIN="Philadelphia" DESTINATION="Seattle"
VALUE="38.5" />

<PV ORIGIN="Philadelphia" DESTINATION="St. Louis"
VALUE="15" />

<PV ORIGIN="Philadelphia" DESTINATION="Tampa"
VALUE="20" />

<PV ORIGIN="Philadelphia" DESTINATION="Toronto"
VALUE="7" />

<PV ORIGIN="Philadelphia" DESTINATION="Washington
D.C." VALUE="2" />

<PV ORIGIN="Phoenix" DESTINATION="Atlanta"
VALUE="21" />

<PV ORIGIN="Phoenix" DESTINATION="Boston" VALUE="34"
/>

<PV ORIGIN="Phoenix" DESTINATION="Chicago"
VALUE="24.5" />

<PV ORIGIN="Phoenix" DESTINATION="Cleveland"
VALUE="31" />

<PV ORIGIN="Phoenix" DESTINATION="Dallas"
VALUE="11.5" />

<PV ORIGIN="Phoenix" DESTINATION="Denver" VALUE="19"
/>

<PV ORIGIN="Phoenix" DESTINATION="Detroit"
VALUE="30" />

<PV ORIGIN="Phoenix" DESTINATION="Houston"
VALUE="14" />

<PV ORIGIN="Phoenix" DESTINATION="Kansas City"
VALUE="22.5" />

<PV ORIGIN="Phoenix" DESTINATION="Los Angeles"
VALUE="5" />
```

```
<PV ORIGIN="Phoenix" DESTINATION="Miami"
VALUE="25.5" />

<PV ORIGIN="Phoenix" DESTINATION="Milwaukee"
VALUE="23.5" />

<PV ORIGIN="Phoenix" DESTINATION="Minneapolis"
VALUE="27" />

<PV ORIGIN="Phoenix" DESTINATION="New York"
VALUE="29.5" />

<PV ORIGIN="Phoenix" DESTINATION="Philadelphia"
VALUE="31" />

<PV ORIGIN="Phoenix" DESTINATION="Phoenix" VALUE="0"
/>

<PV ORIGIN="Phoenix" DESTINATION="Phoenix" VALUE="0"
/>

<PV ORIGIN="Phoenix" DESTINATION="San Diego"
VALUE="7" />

<PV ORIGIN="Phoenix" DESTINATION="San Francisco"
VALUE="9.5" />

<PV ORIGIN="Phoenix" DESTINATION="San Jose"
VALUE="22" />

<PV ORIGIN="Phoenix" DESTINATION="Seattle"
VALUE="17.5" />

<PV ORIGIN="Phoenix" DESTINATION="St. Louis"
VALUE="27" />

<PV ORIGIN="Phoenix" DESTINATION="Tampa" VALUE="28"
/>

<PV ORIGIN="Phoenix" DESTINATION="Toronto"
VALUE="29.5" />

<PV ORIGIN="Phoenix" DESTINATION="Washington D.C."
VALUE="28" />

<PV ORIGIN="Richardson" DESTINATION="Amsterdam"
VALUE="64.05" />

<PV ORIGIN="Richardson" DESTINATION="Atlanta"
VALUE="9.345" />
```

Appendix B

```
<PV ORIGIN="Richardson" DESTINATION="Chicago"
VALUE="11.43" />

<PV ORIGIN="Richardson" DESTINATION="Cleveland"
VALUE="15.35" />

<PV ORIGIN="Richardson" DESTINATION="Copenhagen"
VALUE="71.2" />

<PV ORIGIN="Richardson" DESTINATION="Dallas"
VALUE="0.68" />

<PV ORIGIN="Richardson" DESTINATION="Denver"
VALUE="11.605" />

<PV ORIGIN="Richardson" DESTINATION="Detroit"
VALUE="14.12" />

<PV ORIGIN="Richardson" DESTINATION="Elmhurst"
VALUE="11.77" />

<PV ORIGIN="Richardson" DESTINATION="Frankfurt"
VALUE="73.55" />

<PV ORIGIN="Richardson" DESTINATION="Helsinki"
VALUE="74.4" />

<PV ORIGIN="Richardson" DESTINATION="Herndon"
VALUE="17.13" />

<PV ORIGIN="Richardson" DESTINATION="Hong Kong"
VALUE="101.9" />

<PV ORIGIN="Richardson" DESTINATION="Houston"
VALUE="3.27" />

<PV ORIGIN="Richardson" DESTINATION="Irvine"
VALUE="17.805" />

<PV ORIGIN="Richardson" DESTINATION="Kansas City"
VALUE="6.035" />

<PV ORIGIN="Richardson" DESTINATION="London"
VALUE="59.3" />

<PV ORIGIN="Richardson" DESTINATION="Los Angeles"
VALUE="18.49" />

<PV ORIGIN="Richardson" DESTINATION="Madrid"
VALUE="71.45" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Mexico City"  
VALUE="21.62" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Miami"  
VALUE="15.92" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Milan"  
VALUE="65.45" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Munich"  
VALUE="72.35" />
```

```
<PV ORIGIN="Richardson" DESTINATION="New York"  
VALUE="19.865" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Oslo"  
VALUE="85.7" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Paris"  
VALUE="64.95" />
```

```
<PV ORIGIN="Richardson" DESTINATION="San Francisco"  
VALUE="25.81" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Santa Clara"  
VALUE="23.4" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Seattle"  
VALUE="25.94" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Seoul"  
VALUE="91.85" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Shanghai"  
VALUE="121.25" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Singapore"  
VALUE="123.45" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Stockholm"  
VALUE="70.95" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Sydney"  
VALUE="94.5" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Taipei"  
VALUE="83.1" />
```

```
<PV ORIGIN="Richardson" DESTINATION="Tokyo"  
VALUE="89.15" />
```

Appendix B

```
<PV ORIGIN="Richardson" DESTINATION="Toronto"
VALUE="19.185" />

<PV ORIGIN="Richardson" DESTINATION="Vancouver"
VALUE="36.415" />

<PV ORIGIN="San Diego" DESTINATION="Atlanta"
VALUE="26.5" />

<PV ORIGIN="San Diego" DESTINATION="Boston"
VALUE="40" />

<PV ORIGIN="San Diego" DESTINATION="Chicago"
VALUE="32" />

<PV ORIGIN="San Diego" DESTINATION="Cleveland"
VALUE="39" />

<PV ORIGIN="San Diego" DESTINATION="Dallas"
VALUE="16" />

<PV ORIGIN="San Diego" DESTINATION="Denver"
VALUE="23" />

<PV ORIGIN="San Diego" DESTINATION="Detroit"
VALUE="36" />

<PV ORIGIN="San Diego" DESTINATION="Houston"
VALUE="19.5" />

<PV ORIGIN="San Diego" DESTINATION="Kansas City"
VALUE="27" />

<PV ORIGIN="San Diego" DESTINATION="Los Angeles"
VALUE="4" />

<PV ORIGIN="San Diego" DESTINATION="Miami"
VALUE="40" />

<PV ORIGIN="San Diego" DESTINATION="Milwaukee"
VALUE="36" />

<PV ORIGIN="San Diego" DESTINATION="Minneapolis"
VALUE="33" />

<PV ORIGIN="San Diego" DESTINATION="NYC" VALUE="35"
/>

<PV ORIGIN="San Diego" DESTINATION="Philadelphia"
VALUE="35.5" />
```

```
<PV ORIGIN="San Diego" DESTINATION="Phoenix"
VALUE="7" />

<PV ORIGIN="San Diego" DESTINATION="San Diego"
VALUE="0" />

<PV ORIGIN="San Diego" DESTINATION="San Francisco"
VALUE="8" />

<PV ORIGIN="San Diego" DESTINATION="San Jose"
VALUE="10" />

<PV ORIGIN="San Diego" DESTINATION="Seattle"
VALUE="19" />

<PV ORIGIN="San Diego" DESTINATION="St. Louis"
VALUE="34" />

<PV ORIGIN="San Diego" DESTINATION="Tampa"
VALUE="35.5" />

<PV ORIGIN="San Diego" DESTINATION="Washington D.C."
VALUE="31.5" />

<PV ORIGIN="San Francisco" DESTINATION="Amsterdam"
VALUE="91.2" />

<PV ORIGIN="San Francisco" DESTINATION="Atlanta"
VALUE="28" />

<PV ORIGIN="San Francisco" DESTINATION="Boston"
VALUE="37" />

<PV ORIGIN="San Francisco" DESTINATION="Chicago"
VALUE="26.49" />

<PV ORIGIN="San Francisco" DESTINATION="Cleveland"
VALUE="33.625" />

<PV ORIGIN="San Francisco" DESTINATION="Copenhagen"
VALUE="89.8" />

<PV ORIGIN="San Francisco" DESTINATION="Dallas"
VALUE="20.5" />

<PV ORIGIN="San Francisco" DESTINATION="Denver"
VALUE="13.65" />

<PV ORIGIN="San Francisco" DESTINATION="Detroit"
VALUE="29.145" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Elmhurst"  
VALUE="26.495" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Frankfurt"  
VALUE="93.7" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Helsinki"  
VALUE="89.8" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Herndon"  
VALUE="33.855" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Hong Kong"  
VALUE="83.95" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Houston"  
VALUE="23" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Irvine"  
VALUE="7.44" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Kansas City"  
VALUE="22.055" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="London"  
VALUE="77.65" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Los Angeles"  
VALUE="4.5" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Madrid"  
VALUE="93.45" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Mexico City"  
VALUE="47.375" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Miami"  
VALUE="36" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Milan"  
VALUE="85.45" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Milwaukee"  
VALUE="27.5" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Minneapolis"  
VALUE="28.5" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="Munich"  
VALUE="89.25" />
```

```
<PV ORIGIN="San Francisco" DESTINATION="New York"
VALUE="32.5" />

<PV ORIGIN="San Francisco" DESTINATION="Oslo"
VALUE="103.45" />

<PV ORIGIN="San Francisco" DESTINATION="Paris"
VALUE="87.65" />

<PV ORIGIN="San Francisco"
DESTINATION="Philadelphia" VALUE="31.5" />

<PV ORIGIN="San Francisco" DESTINATION="Phoenix"
VALUE="9.5" />

<PV ORIGIN="San Francisco" DESTINATION="Richardson"
VALUE="28.83" />

<PV ORIGIN="San Francisco" DESTINATION="San Diego"
VALUE="8" />

<PV ORIGIN="San Francisco" DESTINATION="San Jose"
VALUE="3.5" />

<PV ORIGIN="San Francisco" DESTINATION="Santa Clara"
VALUE="1.025" />

<PV ORIGIN="San Francisco" DESTINATION="Seattle"
VALUE="8.5" />

<PV ORIGIN="San Francisco" DESTINATION="Seoul"
VALUE="94.3" />

<PV ORIGIN="San Francisco" DESTINATION="Shanghai"
VALUE="112.4" />

<PV ORIGIN="San Francisco" DESTINATION="Singapore"
VALUE="107.9" />

<PV ORIGIN="San Francisco" DESTINATION="St. Louis"
VALUE="27" />

<PV ORIGIN="San Francisco" DESTINATION="Stockholm"
VALUE="87.7" />

<PV ORIGIN="San Francisco" DESTINATION="Sydney"
VALUE="94.3" />

<PV ORIGIN="San Francisco" DESTINATION="Taipei"
VALUE="71.7" />
```

Appendix B

```
<PV ORIGIN="San Francisco" DESTINATION="Tampa"
VALUE="35" />

<PV ORIGIN="San Francisco" DESTINATION="Tokyo"
VALUE="65.5" />

<PV ORIGIN="San Francisco" DESTINATION="Toronto"
VALUE="33.075" />

<PV ORIGIN="San Francisco" DESTINATION="Vancouver"
VALUE="14.29" />

<PV ORIGIN="San Francisco" DESTINATION="Washington
D.C." VALUE="33" />

<PV ORIGIN="San Jose" DESTINATION="Atlanta"
VALUE="45" />

<PV ORIGIN="San Jose" DESTINATION="Boston"
VALUE="43.5" />

<PV ORIGIN="San Jose" DESTINATION="Chicago"
VALUE="31" />

<PV ORIGIN="San Jose" DESTINATION="Cleveland"
VALUE="34" />

<PV ORIGIN="San Jose" DESTINATION="Dallas"
VALUE="23.5" />

<PV ORIGIN="San Jose" DESTINATION="Denver"
VALUE="28" />

<PV ORIGIN="San Jose" DESTINATION="Detroit"
VALUE="35" />

<PV ORIGIN="San Jose" DESTINATION="Houston"
VALUE="25" />

<PV ORIGIN="San Jose" DESTINATION="Kansas City"
VALUE="31" />

<PV ORIGIN="San Jose" DESTINATION="Los Angeles"
VALUE="8.5" />

<PV ORIGIN="San Jose" DESTINATION="Miami" VALUE="44"
/>

<PV ORIGIN="San Jose" DESTINATION="Milwaukee"
VALUE="32" />
```

```
<PV ORIGIN="San Jose" DESTINATION="Minneapolis"
VALUE="34" />

<PV ORIGIN="San Jose" DESTINATION="NYC" VALUE="41" /
>

<PV ORIGIN="San Jose" DESTINATION="Philadelphia"
VALUE="44" />

<PV ORIGIN="San Jose" DESTINATION="Phoenix"
VALUE="22" />

<PV ORIGIN="San Jose" DESTINATION="San Diego"
VALUE="10" />

<PV ORIGIN="San Jose" DESTINATION="San Francisco"
VALUE="3.5" />

<PV ORIGIN="San Jose" DESTINATION="San Jose"
VALUE="0" />

<PV ORIGIN="San Jose" DESTINATION="Seattle"
VALUE="14" />

<PV ORIGIN="San Jose" DESTINATION="St. Louis"
VALUE="33" />

<PV ORIGIN="San Jose" DESTINATION="Tampa" VALUE="47"
/>

<PV ORIGIN="San Jose" DESTINATION="Washington D.C."
VALUE="41" />

<PV ORIGIN="Santa Clara" DESTINATION="Amsterdam"
VALUE="86.3" />

<PV ORIGIN="Santa Clara" DESTINATION="Atlanta"
VALUE="27.56" />

<PV ORIGIN="Santa Clara" DESTINATION="Chicago"
VALUE="25.305" />

<PV ORIGIN="Santa Clara" DESTINATION="Cleveland"
VALUE="29.49" />

<PV ORIGIN="Santa Clara" DESTINATION="Copenhagen"
VALUE="88.05" />

<PV ORIGIN="Santa Clara" DESTINATION="Dallas"
VALUE="23.465" />
```

```
<PV ORIGIN="Santa Clara" DESTINATION="Denver"
VALUE="14.025" />

<PV ORIGIN="Santa Clara" DESTINATION="Detroit"
VALUE="28.455" />

<PV ORIGIN="Santa Clara" DESTINATION="Elmhurst"
VALUE="25.6" />

<PV ORIGIN="Santa Clara" DESTINATION="Frankfurt"
VALUE="91.85" />

<PV ORIGIN="Santa Clara" DESTINATION="Helsinki"
VALUE="91.6" />

<PV ORIGIN="Santa Clara" DESTINATION="Herndon"
VALUE="33.375" />

<PV ORIGIN="Santa Clara" DESTINATION="Hong Kong"
VALUE="80.65" />

<PV ORIGIN="Santa Clara" DESTINATION="Houston"
VALUE="21.23" />

<PV ORIGIN="Santa Clara" DESTINATION="Irvine"
VALUE="6.585" />

<PV ORIGIN="Santa Clara" DESTINATION="Kansas City"
VALUE="20.13" />

<PV ORIGIN="Santa Clara" DESTINATION="London"
VALUE="77.15" />

<PV ORIGIN="Santa Clara" DESTINATION="Los Angeles"
VALUE="7.425" />

<PV ORIGIN="Santa Clara" DESTINATION="Madrid"
VALUE="91.25" />

<PV ORIGIN="Santa Clara" DESTINATION="Mexico City"
VALUE="46.925" />

<PV ORIGIN="Santa Clara" DESTINATION="Miami"
VALUE="34.445" />

<PV ORIGIN="Santa Clara" DESTINATION="Milan"
VALUE="83.45" />

<PV ORIGIN="Santa Clara" DESTINATION="Munich"
VALUE="87.85" />
```

```
<PV ORIGIN="Santa Clara" DESTINATION="New York"
VALUE="36.495" />

<PV ORIGIN="Santa Clara" DESTINATION="Oslo"
VALUE="101.2" />

<PV ORIGIN="Santa Clara" DESTINATION="Paris"
VALUE="86.45" />

<PV ORIGIN="Santa Clara" DESTINATION="Richardson"
VALUE="23.86" />

<PV ORIGIN="Santa Clara" DESTINATION="San Francisco"
VALUE="1.005" />

<PV ORIGIN="Santa Clara" DESTINATION="Santa Clara"
VALUE="NA" />

<PV ORIGIN="Santa Clara" DESTINATION="Seattle"
VALUE="10.28" />

<PV ORIGIN="Santa Clara" DESTINATION="Seoul"
VALUE="93.95" />

<PV ORIGIN="Santa Clara" DESTINATION="Shanghai"
VALUE="112.4" />

<PV ORIGIN="Santa Clara" DESTINATION="Singapore"
VALUE="142.85" />

<PV ORIGIN="Santa Clara" DESTINATION="Stockholm"
VALUE="85.35" />

<PV ORIGIN="Santa Clara" DESTINATION="Sydney"
VALUE="88.15" />

<PV ORIGIN="Santa Clara" DESTINATION="Taipei"
VALUE="71.1" />

<PV ORIGIN="Santa Clara" DESTINATION="Tokyo"
VALUE="68.9" />

<PV ORIGIN="Santa Clara" DESTINATION="Toronto"
VALUE="33.29" />

<PV ORIGIN="Santa Clara" DESTINATION="Vancouver"
VALUE="13.31" />

<PV ORIGIN="Seattle" DESTINATION="Amsterdam"
VALUE="90.65" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Atlanta"  
VALUE="30.5" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Boston" VALUE="35"  
/>
```

```
<PV ORIGIN="Seattle" DESTINATION="Chicago"  
VALUE="25" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Cleveland"  
VALUE="29.555" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Copenhagen"  
VALUE="88.25" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Dallas"  
VALUE="20.5" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Denver"  
VALUE="13.5" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Detroit"  
VALUE="28.65" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Elmhurst"  
VALUE="26.07" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Frankfurt"  
VALUE="95.3" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Helsinki"  
VALUE="92.4" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Herndon"  
VALUE="33.03" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Hong Kong"  
VALUE="89.05" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Houston"  
VALUE="28.505" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Irvine"  
VALUE="15.95" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Kansas City"  
VALUE="20.45" />
```

```
<PV ORIGIN="Seattle" DESTINATION="London"  
VALUE="81.4" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Los Angeles"
VALUE="12.5" />

<PV ORIGIN="Seattle" DESTINATION="Madrid"
VALUE="91.95" />

<PV ORIGIN="Seattle" DESTINATION="Mexico City"
VALUE="50.8" />

<PV ORIGIN="Seattle" DESTINATION="Miami"
VALUE="34.5" />

<PV ORIGIN="Seattle" DESTINATION="Milan"
VALUE="87.9" />

<PV ORIGIN="Seattle" DESTINATION="Milwaukee"
VALUE="30.5" />

<PV ORIGIN="Seattle" DESTINATION="Minneapolis"
VALUE="29" />

<PV ORIGIN="Seattle" DESTINATION="Munich"
VALUE="88.8" />

<PV ORIGIN="Seattle" DESTINATION="New York"
VALUE="35" />

<PV ORIGIN="Seattle" DESTINATION="Oslo"
VALUE="102.2" />

<PV ORIGIN="Seattle" DESTINATION="Paris"
VALUE="90.25" />

<PV ORIGIN="Seattle" DESTINATION="Philadelphia"
VALUE="38.5" />

<PV ORIGIN="Seattle" DESTINATION="Phoenix"
VALUE="17" />

<PV ORIGIN="Seattle" DESTINATION="Richardson"
VALUE="26.13" />

<PV ORIGIN="Seattle" DESTINATION="San Diego"
VALUE="19" />

<PV ORIGIN="Seattle" DESTINATION="San Francisco"
VALUE="8.5" />

<PV ORIGIN="Seattle" DESTINATION="San Jose"
VALUE="14" />
```

```
<PV ORIGIN="Seattle" DESTINATION="Santa Clara"
VALUE="9.98" />

<PV ORIGIN="Seattle" DESTINATION="Seoul"
VALUE="99.4" />

<PV ORIGIN="Seattle" DESTINATION="Shanghai"
VALUE="125.5" />

<PV ORIGIN="Seattle" DESTINATION="Singapore"
VALUE="114" />

<PV ORIGIN="Seattle" DESTINATION="St. Louis"
VALUE="37" />

<PV ORIGIN="Seattle" DESTINATION="Stockholm"
VALUE="88.45" />

<PV ORIGIN="Seattle" DESTINATION="Sydney"
VALUE="96.75" />

<PV ORIGIN="Seattle" DESTINATION="Taipei"
VALUE="79.25" />

<PV ORIGIN="Seattle" DESTINATION="Tampa" VALUE="46"
/>

<PV ORIGIN="Seattle" DESTINATION="Tokyo"
VALUE="72.45" />

<PV ORIGIN="Seattle" DESTINATION="Toronto"
VALUE="30.5" />

<PV ORIGIN="Seattle" DESTINATION="Vancouver"
VALUE="22.23" />

<PV ORIGIN="Seattle" DESTINATION="Washington D.C."
VALUE="34" />

<PV ORIGIN="Seoul" DESTINATION="Atlanta"
VALUE="104.2" />

<PV ORIGIN="Seoul" DESTINATION="Chicago" VALUE="99"
/>

<PV ORIGIN="Seoul" DESTINATION="Cleveland"
VALUE="107.1" />

<PV ORIGIN="Seoul" DESTINATION="Dallas"
VALUE="89.55" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Denver" VALUE="92.3"  
/>
```

```
<PV ORIGIN="Seoul" DESTINATION="Detroit"  
VALUE="108.5" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Elmhurst"  
VALUE="97.55" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Herndon"  
VALUE="104.55" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Houston"  
VALUE="90.65" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Irvine" VALUE="91.7"  
/>
```

```
<PV ORIGIN="Seoul" DESTINATION="Kansas City"  
VALUE="94.95" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Los Angeles"  
VALUE="93.25" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Miami"  
VALUE="107.65" />
```

```
<PV ORIGIN="Seoul" DESTINATION="New York"  
VALUE="107.25" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Richardson"  
VALUE="91.85" />
```

```
<PV ORIGIN="Seoul" DESTINATION="San Francisco"  
VALUE="94.3" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Santa Clara"  
VALUE="93.95" />
```

```
<PV ORIGIN="Seoul" DESTINATION="Seattle"  
VALUE="99.4" />
```

```
<PV ORIGIN="Shanghai" DESTINATION="Atlanta"  
VALUE="134.9" />
```

```
<PV ORIGIN="Shanghai" DESTINATION="Chicago"  
VALUE="143" />
```

```
<PV ORIGIN="Shanghai" DESTINATION="Cleveland"  
VALUE="147.3" />
```

Appendix B

```
<PV ORIGIN="Shanghai" DESTINATION="Dallas"
VALUE="121.3" />

<PV ORIGIN="Shanghai" DESTINATION="Denver"
VALUE="124.6" />

<PV ORIGIN="Shanghai" DESTINATION="Detroit"
VALUE="137.3" />

<PV ORIGIN="Shanghai" DESTINATION="Elmhurst"
VALUE="144.8" />

<PV ORIGIN="Shanghai" DESTINATION="Herndon"
VALUE="140.9" />

<PV ORIGIN="Shanghai" DESTINATION="Houston"
VALUE="127.55" />

<PV ORIGIN="Shanghai" DESTINATION="Irvine"
VALUE="105.05" />

<PV ORIGIN="Shanghai" DESTINATION="Kansas City"
VALUE="134.2" />

<PV ORIGIN="Shanghai" DESTINATION="Los Angeles"
VALUE="109.15" />

<PV ORIGIN="Shanghai" DESTINATION="Miami"
VALUE="132.9" />

<PV ORIGIN="Shanghai" DESTINATION="New York"
VALUE="136.1" />

<PV ORIGIN="Shanghai" DESTINATION="Richardson"
VALUE="121.25" />

<PV ORIGIN="Shanghai" DESTINATION="San Francisco"
VALUE="112.4" />

<PV ORIGIN="Shanghai" DESTINATION="Santa Clara"
VALUE="112.4" />

<PV ORIGIN="Shanghai" DESTINATION="Seattle"
VALUE="125.5" />

<PV ORIGIN="Singapore" DESTINATION="Atlanta"
VALUE="135.95" />

<PV ORIGIN="Singapore" DESTINATION="Chicago"
VALUE="131.95" />
```

```
<PV ORIGIN="Singapore" DESTINATION="Cleveland"
VALUE="142.8" />

<PV ORIGIN="Singapore" DESTINATION="Dallas"
VALUE="123.55" />

<PV ORIGIN="Singapore" DESTINATION="Denver"
VALUE="118.85" />

<PV ORIGIN="Singapore" DESTINATION="Detroit"
VALUE="142.2" />

<PV ORIGIN="Singapore" DESTINATION="Elmhurst"
VALUE="131.9" />

<PV ORIGIN="Singapore" DESTINATION="Herndon"
VALUE="143.1" />

<PV ORIGIN="Singapore" DESTINATION="Houston"
VALUE="124.75" />

<PV ORIGIN="Singapore" DESTINATION="Irvine"
VALUE="108.2" />

<PV ORIGIN="Singapore" DESTINATION="Kansas City"
VALUE="123.9" />

<PV ORIGIN="Singapore" DESTINATION="Los Angeles"
VALUE="107.2" />

<PV ORIGIN="Singapore" DESTINATION="Miami"
VALUE="136.4" />

<PV ORIGIN="Singapore" DESTINATION="New York"
VALUE="139.65" />

<PV ORIGIN="Singapore" DESTINATION="Richardson"
VALUE="123.45" />

<PV ORIGIN="Singapore" DESTINATION="San Francisco"
VALUE="107.9" />

<PV ORIGIN="Singapore" DESTINATION="Santa Clara"
VALUE="142.85" />

<PV ORIGIN="Singapore" DESTINATION="Seattle"
VALUE="114" />

<PV ORIGIN="St. Louis" DESTINATION="Atlanta"
VALUE="19" />
```

```
<PV ORIGIN="St. Louis" DESTINATION="Boston"
VALUE="17" />

<PV ORIGIN="St. Louis" DESTINATION="Chicago"
VALUE="4" />

<PV ORIGIN="St. Louis" DESTINATION="Cleveland"
VALUE="12" />

<PV ORIGIN="St. Louis" DESTINATION="Dallas"
VALUE="19" />

<PV ORIGIN="St. Louis" DESTINATION="Denver"
VALUE="33" />

<PV ORIGIN="St. Louis" DESTINATION="Detroit"
VALUE="6.5" />

<PV ORIGIN="St. Louis" DESTINATION="Houston"
VALUE="19" />

<PV ORIGIN="St. Louis" DESTINATION="Kansas City"
VALUE="14" />

<PV ORIGIN="St. Louis" DESTINATION="Los Angeles"
VALUE="33.5" />

<PV ORIGIN="St. Louis" DESTINATION="Miami"
VALUE="21" />

<PV ORIGIN="St. Louis" DESTINATION="Milwaukee"
VALUE="5.5" />

<PV ORIGIN="St. Louis" DESTINATION="Minneapolis"
VALUE="31" />

<PV ORIGIN="St. Louis" DESTINATION="NYC" VALUE="17"
/>

<PV ORIGIN="St. Louis" DESTINATION="Philadelphia"
VALUE="15" />

<PV ORIGIN="St. Louis" DESTINATION="Phoenix"
VALUE="27" />

<PV ORIGIN="St. Louis" DESTINATION="San Diego"
VALUE="34" />

<PV ORIGIN="St. Louis" DESTINATION="San Francisco"
VALUE="27" />
```

```
<PV ORIGIN="St. Louis" DESTINATION="San Jose"
VALUE="33" />

<PV ORIGIN="St. Louis" DESTINATION="Seattle"
VALUE="37" />

<PV ORIGIN="St. Louis" DESTINATION="St. Louis"
VALUE="0" />

<PV ORIGIN="St. Louis" DESTINATION="Tampa"
VALUE="30" />

<PV ORIGIN="St. Louis" DESTINATION="Washington D.C."
VALUE="14" />

<PV ORIGIN="Stockholm" DESTINATION="Atlanta"
VALUE="56.95" />

<PV ORIGIN="Stockholm" DESTINATION="Chicago"
VALUE="59.8" />

<PV ORIGIN="Stockholm" DESTINATION="Cleveland"
VALUE="56.4" />

<PV ORIGIN="Stockholm" DESTINATION="Dallas"
VALUE="71.65" />

<PV ORIGIN="Stockholm" DESTINATION="Denver"
VALUE="82.4" />

<PV ORIGIN="Stockholm" DESTINATION="Detroit"
VALUE="57.15" />

<PV ORIGIN="Stockholm" DESTINATION="Elmhurst"
VALUE="58.8" />

<PV ORIGIN="Stockholm" DESTINATION="Herndon"
VALUE="51.65" />

<PV ORIGIN="Stockholm" DESTINATION="Houston"
VALUE="65.7" />

<PV ORIGIN="Stockholm" DESTINATION="Irvine"
VALUE="82.25" />

<PV ORIGIN="Stockholm" DESTINATION="Kansas City"
VALUE="67.5" />

<PV ORIGIN="Stockholm" DESTINATION="Los Angeles"
VALUE="86.75" />
```

Appendix B

```
<PV ORIGIN="Stockholm" DESTINATION="Miami"
VALUE="64.85" />

<PV ORIGIN="Stockholm" DESTINATION="New York"
VALUE="47.635" />

<PV ORIGIN="Stockholm" DESTINATION="Richardson"
VALUE="70.95" />

<PV ORIGIN="Stockholm" DESTINATION="San Francisco"
VALUE="87.7" />

<PV ORIGIN="Stockholm" DESTINATION="Santa Clara"
VALUE="85.35" />

<PV ORIGIN="Stockholm" DESTINATION="Seattle"
VALUE="88.45" />

<PV ORIGIN="Sydney" DESTINATION="Atlanta"
VALUE="97.15" />

<PV ORIGIN="Sydney" DESTINATION="Chicago"
VALUE="118.75" />

<PV ORIGIN="Sydney" DESTINATION="Cleveland"
VALUE="115.75" />

<PV ORIGIN="Sydney" DESTINATION="Dallas"
VALUE="92.25" />

<PV ORIGIN="Sydney" DESTINATION="Denver"
VALUE="103.85" />

<PV ORIGIN="Sydney" DESTINATION="Detroit"
VALUE="114" />

<PV ORIGIN="Sydney" DESTINATION="Elmhurst"
VALUE="118.1" />

<PV ORIGIN="Sydney" DESTINATION="Herndon"
VALUE="100.75" />

<PV ORIGIN="Sydney" DESTINATION="Houston"
VALUE="90.75" />

<PV ORIGIN="Sydney" DESTINATION="Irvine"
VALUE="75.3" />

<PV ORIGIN="Sydney" DESTINATION="Kansas City"
VALUE="113.7" />
```

```
<PV ORIGIN="Sydney" DESTINATION="Los Angeles"
VALUE="78.05" />

<PV ORIGIN="Sydney" DESTINATION="Miami"
VALUE="102.55" />

<PV ORIGIN="Sydney" DESTINATION="New York"
VALUE="106.6" />

<PV ORIGIN="Sydney" DESTINATION="Richardson"
VALUE="94.5" />

<PV ORIGIN="Sydney" DESTINATION="San Francisco"
VALUE="94.3" />

<PV ORIGIN="Sydney" DESTINATION="Santa Clara"
VALUE="88.15" />

<PV ORIGIN="Sydney" DESTINATION="Seattle"
VALUE="96.75" />

<PV ORIGIN="Taipei" DESTINATION="Atlanta"
VALUE="91.3" />

<PV ORIGIN="Taipei" DESTINATION="Chicago"
VALUE="94.2" />

<PV ORIGIN="Taipei" DESTINATION="Cleveland"
VALUE="96.5" />

<PV ORIGIN="Taipei" DESTINATION="Dallas"
VALUE="86.9" />

<PV ORIGIN="Taipei" DESTINATION="Denver"
VALUE="83.5" />

<PV ORIGIN="Taipei" DESTINATION="Detroit"
VALUE="96.5" />

<PV ORIGIN="Taipei" DESTINATION="Elmhurst"
VALUE="94.05" />

<PV ORIGIN="Taipei" DESTINATION="Herndon"
VALUE="96.4" />

<PV ORIGIN="Taipei" DESTINATION="Houston"
VALUE="84.45" />

<PV ORIGIN="Taipei" DESTINATION="Irvine"
VALUE="70.95" />
```

Appendix B

```
<PV ORIGIN="Taipei" DESTINATION="Kansas City"
VALUE="88.2" />

<PV ORIGIN="Taipei" DESTINATION="Los Angeles"
VALUE="70.3" />

<PV ORIGIN="Taipei" DESTINATION="Miami" VALUE="96.9"
/>

<PV ORIGIN="Taipei" DESTINATION="New York"
VALUE="99.1" />

<PV ORIGIN="Taipei" DESTINATION="Richardson"
VALUE="83.1" />

<PV ORIGIN="Taipei" DESTINATION="San Francisco"
VALUE="71.7" />

<PV ORIGIN="Taipei" DESTINATION="Santa Clara"
VALUE="71.1" />

<PV ORIGIN="Taipei" DESTINATION="Seattle"
VALUE="79.25" />

<PV ORIGIN="Tampa" DESTINATION="Atlanta" VALUE="7" /
>

<PV ORIGIN="Tampa" DESTINATION="Boston" VALUE="24" /
>

<PV ORIGIN="Tampa" DESTINATION="Chicago" VALUE="15"
/>

<PV ORIGIN="Tampa" DESTINATION="Cleveland"
VALUE="21" />

<PV ORIGIN="Tampa" DESTINATION="Dallas" VALUE="18" /
>

<PV ORIGIN="Tampa" DESTINATION="Denver" VALUE="26" /
>

<PV ORIGIN="Tampa" DESTINATION="Detroit" VALUE="30"
/>

<PV ORIGIN="Tampa" DESTINATION="Houston" VALUE="26"
/>

<PV ORIGIN="Tampa" DESTINATION="Kansas City"
VALUE="20" />
```

```
<PV ORIGIN="Tampa" DESTINATION="Los Angeles"
VALUE="33" />

<PV ORIGIN="Tampa" DESTINATION="Miami" VALUE="15" />

<PV ORIGIN="Tampa" DESTINATION="Milwaukee"
VALUE="23" />

<PV ORIGIN="Tampa" DESTINATION="Minneapolis"
VALUE="20" />

<PV ORIGIN="Tampa" DESTINATION="NYC" VALUE="19.5" />

<PV ORIGIN="Tampa" DESTINATION="Philadelphia"
VALUE="20" />

<PV ORIGIN="Tampa" DESTINATION="Phoenix" VALUE="28"
/>

<PV ORIGIN="Tampa" DESTINATION="San Diego"
VALUE="35.5" />

<PV ORIGIN="Tampa" DESTINATION="San Francisco"
VALUE="35" />

<PV ORIGIN="Tampa" DESTINATION="San Jose" VALUE="47"
/>

<PV ORIGIN="Tampa" DESTINATION="Seattle" VALUE="46"
/>

<PV ORIGIN="Tampa" DESTINATION="St. Louis"
VALUE="30" />

<PV ORIGIN="Tampa" DESTINATION="Tampa" VALUE="0" />

<PV ORIGIN="Tampa" DESTINATION="Washington D.C."
VALUE="18" />

<PV ORIGIN="Tokyo" DESTINATION="Atlanta"
VALUE="95.35" />

<PV ORIGIN="Tokyo" DESTINATION="Chicago"
VALUE="88.95" />

<PV ORIGIN="Tokyo" DESTINATION="Cleveland"
VALUE="96.9" />

<PV ORIGIN="Tokyo" DESTINATION="Dallas" VALUE="89" /
>
```

```
<PV ORIGIN="Tokyo" DESTINATION="Denver"  
VALUE="72.25" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Detroit" VALUE="97"  
/>  
  
<PV ORIGIN="Tokyo" DESTINATION="Elmhurst"  
VALUE="89.15" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Herndon"  
VALUE="95.75" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Houston"  
VALUE="89.6" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Irvine"  
VALUE="71.85" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Kansas City"  
VALUE="89.9" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Los Angeles"  
VALUE="72.65" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Miami" VALUE="98.3"  
/>  
  
<PV ORIGIN="Tokyo" DESTINATION="New York"  
VALUE="98.55" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Richardson"  
VALUE="89.15" />  
  
<PV ORIGIN="Tokyo" DESTINATION="San Francisco"  
VALUE="65.5" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Santa Clara"  
VALUE="68.9" />  
  
<PV ORIGIN="Tokyo" DESTINATION="Seattle"  
VALUE="72.45" />  
  
<PV ORIGIN="Toronto" DESTINATION="Atlanta"  
VALUE="19.84" />  
  
<PV ORIGIN="Toronto" DESTINATION="Chicago"  
VALUE="7.285" />  
  
<PV ORIGIN="Toronto" DESTINATION="Cleveland"  
VALUE="11.62" />
```

```
<PV ORIGIN="Toronto" DESTINATION="Dallas"
VALUE="18.725" />

<PV ORIGIN="Toronto" DESTINATION="Denver"
VALUE="19.095" />

<PV ORIGIN="Toronto" DESTINATION="Detroit"
VALUE="10.66" />

<PV ORIGIN="Toronto" DESTINATION="Elmhurst"
VALUE="7.33" />

<PV ORIGIN="Toronto" DESTINATION="Herndon"
VALUE="13.685" />

<PV ORIGIN="Toronto" DESTINATION="Houston"
VALUE="21.155" />

<PV ORIGIN="Toronto" DESTINATION="Irvine"
VALUE="40.425" />

<PV ORIGIN="Toronto" DESTINATION="Kansas City"
VALUE="13.56" />

<PV ORIGIN="Toronto" DESTINATION="Los Angeles"
VALUE="39.44" />

<PV ORIGIN="Toronto" DESTINATION="Miami"
VALUE="26.37" />

<PV ORIGIN="Toronto" DESTINATION="New York"
VALUE="13.875" />

<PV ORIGIN="Toronto" DESTINATION="Richardson"
VALUE="19.185" />

<PV ORIGIN="Toronto" DESTINATION="San Francisco"
VALUE="33.075" />

<PV ORIGIN="Toronto" DESTINATION="Santa Clara"
VALUE="33.29" />

<PV ORIGIN="Toronto" DESTINATION="Seattle"
VALUE="33.895" />

<PV ORIGIN="Toronto" DESTINATION="Atlanta"
VALUE="16" />

<PV ORIGIN="Toronto" DESTINATION="Boston" VALUE="9"
/>
```

```
<PV ORIGIN="Toronto" DESTINATION="Chicago"
VALUE="5.5" />

<PV ORIGIN="Toronto" DESTINATION="Dallas"
VALUE="21.5" />

<PV ORIGIN="Toronto" DESTINATION="Denver"
VALUE="17.5" />

<PV ORIGIN="Toronto" DESTINATION="Los Angeles"
VALUE="32.5" />

<PV ORIGIN="Toronto" DESTINATION="Miami" VALUE="23"
/>

<PV ORIGIN="Toronto" DESTINATION="Minneapolis"
VALUE="9.5" />

<PV ORIGIN="Toronto" DESTINATION="New York"
VALUE="6" />

<PV ORIGIN="Toronto" DESTINATION="Philadelphia"
VALUE="7.5" />

<PV ORIGIN="Toronto" DESTINATION="Phoenix"
VALUE="29.5" />

<PV ORIGIN="Toronto" DESTINATION="San Francisco"
VALUE="38.5" />

<PV ORIGIN="Toronto" DESTINATION="Seattle"
VALUE="30.5" />

<PV ORIGIN="Toronto" DESTINATION="Washington D.C."
VALUE="8.5" />

<PV ORIGIN="Vancouver" DESTINATION="Atlanta"
VALUE="43.2" />

<PV ORIGIN="Vancouver" DESTINATION="Chicago"
VALUE="31.125" />

<PV ORIGIN="Vancouver" DESTINATION="Cleveland"
VALUE="43.48" />

<PV ORIGIN="Vancouver" DESTINATION="Dallas"
VALUE="35.985" />

<PV ORIGIN="Vancouver" DESTINATION="Denver"
VALUE="26.425" />
```

```
<PV ORIGIN="Vancouver" DESTINATION="Detroit"
VALUE="43.92" />

<PV ORIGIN="Vancouver" DESTINATION="Elmhurst"
VALUE="31.46" />

<PV ORIGIN="Vancouver" DESTINATION="Herndon"
VALUE="43.5" />

<PV ORIGIN="Vancouver" DESTINATION="Houston"
VALUE="42.73" />

<PV ORIGIN="Vancouver" DESTINATION="Irvine"
VALUE="19.26" />

<PV ORIGIN="Vancouver" DESTINATION="Kansas City"
VALUE="31.535" />

<PV ORIGIN="Vancouver" DESTINATION="Los Angeles"
VALUE="18.15" />

<PV ORIGIN="Vancouver" DESTINATION="Miami"
VALUE="50.2" />

<PV ORIGIN="Vancouver" DESTINATION="New York"
VALUE="46.045" />

<PV ORIGIN="Vancouver" DESTINATION="Richardson"
VALUE="36.415" />

<PV ORIGIN="Vancouver" DESTINATION="San Francisco"
VALUE="14.29" />

<PV ORIGIN="Vancouver" DESTINATION="Santa Clara"
VALUE="13.31" />

<PV ORIGIN="Vancouver" DESTINATION="Seattle"
VALUE="22.23" />

<PV ORIGIN="Washington D.C." DESTINATION="Atlanta"
VALUE="9" />

<PV ORIGIN="Washington D.C." DESTINATION="Boston"
VALUE="8.5" />

<PV ORIGIN="Washington D.C." DESTINATION="Chicago"
VALUE="11" />

<PV ORIGIN="Washington D.C." DESTINATION="Cleveland"
VALUE="6" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Dallas"  
VALUE="19" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Denver"  
VALUE="24.5" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Detroit"  
VALUE="15" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Houston"  
VALUE="22" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Kansas  
City" VALUE="20.5" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Los  
Angeles" VALUE="33" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Miami"  
VALUE="15.5" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Milwaukee"  
VALUE="16.5" />
```

```
<PV ORIGIN="Washington D.C."  
DESTINATION="Minneapolis" VALUE="15.5" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="New York"  
VALUE="2.5" />
```

```
<PV ORIGIN="Washington D.C."  
DESTINATION="Philadelphia" VALUE="2" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Phoenix"  
VALUE="28" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="San Diego"  
VALUE="31.5" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="San  
Francisco" VALUE="33" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="San Jose"  
VALUE="41" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Seattle"  
VALUE="34" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="St. Louis"  
VALUE="14" />
```

```
<PV ORIGIN="Washington D.C." DESTINATION="Tampa"
VALUE="18" />

<PV ORIGIN="Washington D.C." DESTINATION="Toronto"
VALUE="8.5" />

</PRESET_VALUE>

_ <PRESET_VALUE TYPE="ConnectionQuality">

<PV VALUE="Perfect link (off hours)" PacketLoss="0"
LatencyCoe="0" LatencyConst="0" JitterCoe="0"
JitterConst="0" CongestionFreq="0"
CongestionLoss="0" />

<PV VALUE="Normal business hours" PacketLoss="0"
LatencyCoe="1.2" LatencyConst="5" JitterCoe="1.5"
JitterConst="5" CongestionFreq="5"
CongestionLoss="0.5" />

<PV VALUE="Busy business hours" PacketLoss="1"
LatencyCoe="1.6" LatencyConst="10" JitterCoe="2.5"
JitterConst="10" CongestionFreq="2"
CongestionLoss="2" />

<PV VALUE="Worst case" PacketLoss="5" LatencyCoe="3"
LatencyConst="10" JitterCoe="5" JitterConst="10"
CongestionFreq="1" CongestionLoss="7.5" />

</PRESET_VALUE>

_ <PRESET_VALUE TYPE="ConnectionType">

<PV NAME="Small home office (56Kbps)" VALUE="56" />

<PV NAME="DSL home office (256 Kbps)" VALUE="256" />

<PV NAME="Cable modem home office (384 Kbps)"
VALUE="384" />

<PV NAME="Small branch office - 1 - 10 people (512
Kbps)" VALUE="512" />

<PV NAME="Medium branch office - 10 - 30 (768 Kbps)"
VALUE="768" />

<PV NAME="Regional branch office - 30 - 50 people
(T1 - 1544 Kbps)" VALUE="1544" />

<PV NAME="Large branch office (3 Mbps)" VALUE="3072"
/>
```

```
<PV NAME="Unrestricted (no bandwidth limitation)"  
VALUE="0xffffffffUL" />  
  
</PRESET_VALUE>  
  
</PRESET_VALUES>
```

Index

A

- Add Counters 4-1
- Advanced 3-8
- advanced mode 3-4, 3-6
- API 5-1
- API Parameters 5-7
- Application Efficiency 5-2
- automation script 5-5

B

- Basic 3-6
- basic mode 3-6
- business process 3-20
- Business Processes 3-20

C

- cli
 - working with 5-7
- client, overview 1-6, 1-9
- COM Interface 5-6
- components 1-6

D

- debugging 3-5
- dll 5-5

E

- Evaluation License 1-10
- External COM Interface 5-6

H

- HP LoadRunner 5-1
- HP LoadRunner Analysis 4-4
- HP LoadRunner
 - Controller 1-5
- HP Performance Center 1-5, 1-12, 2-6
- HP Performance Center User Site 1-12

- HTML 4-5

I

- Infrastructure Breakdown 5-2
- installing
 - server 2-1
- introduction 1-1
- IP filters
 - exclude 3-17
 - include 3-17

L

- Licensing 1-10

M

- Microsoft Excel 4-5
- mode_advanced 3-8
- MS Word 4-5

N

- Network Editor 1-8
- Network Scenario Files 3-19

P

- PerfMon Counters 4-1
- Performance Center 2-6
- presets B-1

R

- reports 4-1

S

- Samples folder 5-5
- server, about 1-8
- ShunraVEAPI.tlb 5-7
- ShunraVEController.dll 5-5
- startUp project 3-6

T

- Test Plans 3-19
- Time Stamp 5-3
- Transaction Data Report 5-2

Transaction Marker

Report 5-2

Transaction Response

Time 5-3

trial license 1-10

type libraries 5-6

U

Uninstalling 2-17

V

VE Analyzer 1-9

VE Desktop API 5-1

VE Network Catcher 1-9

Visual Studio integration 3-4