VMware Workbench Log Browser Guide

VMware Workbench 3.5

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/suport/pubs.

vmware®

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Copyright © 2014 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

VMware Technical Publications Glossary 5
Using the Workbench Log Browser Wizard to Create a Project File 7
Recording a Log Browser Session 8
Using the Workbench Log Browser GUI 9
Opening an Existing .logx Project 10
Displaying and Using the Activity Graph 10
Command-Line Client Use Cases 13
Where to Find the Command-Line Client-Server 14
- Workbench Log Browser Update Site Plug-in for the Workbench Appliance 14
- Workbench Log Browser Native Eclipse IDE 14

Using the Command-Line with the Workbench GUI 15
Using the Command-Line Client with the Standalone Server 15
Permission Issues when Retrieving a Recorded Log Browser Session 16
Command-Line API Reference 16
- Command Syntax 16
- Getting the List of Available Commands 16
- Getting Help on a Particular Command 16

Use Case Examples 17
- Printing Log Entries to the Console 17
- Downloading Log Files from a Remote Host for Analysis 17
- Discovering Logs on a Host 18
- Downloading All Logs for a Host 18
- Getting the Count of the Total Number of Visible Log Entries 18
- Printing Log Entries to stdout 18
- Applying a Simple Filter 18
- Exporting Entries to a File 19

Known issues/Limitations 21
Advanced Use Cases 21

About This Book

This book, the *VMware® Workbench Log Browser Guide,* provides information about VMware Workbench 3.5 Log Browser.

Revision History

A revision of this book occurs with each release of the product, or as needed. A revised version can contain minor or major changes. Table 1 lists the revision history of this book.

Table 1. Revision History

Revision	Description
09/29/2014	Workbench 3.5 GA for ESXi 6.0 GA

Intended Audience

This book is for VMware partners who will certify applications using the VMware Workbench 3.5

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to http://www.vmware.com/support/pubs.

Documentation Resources

Each VMware Workbench plug-in contains its own help documentation that displays in the **Help > Help Contents** window. PDF copies of these documents may be available from VMware on your partner co-dev web site.

Documentation Feedback

Please send feedback to this documentation through your VMware partners representative.

VMware also welcomes your suggestions for improving our other VMware API and SDK documentation. Send your feedback to: docfeedback@vmware.com.

Technical Support and Education Resources

For all questions relating specifically to the VMware Workbench, see the VMware partner co-dev web site:

http://www.vmware.com/partners/alliances/programs/co-dev/iovp.htm

The following sections describe VMware technical support and educational resources available to you.

Online Support and Telephone Support

To use online support to submit technical support requests, view your product and contract information, and register your products, go to http://www.vmware.com/support.

Support Offerings

To find out how VMware support offerings can help meet your business needs, go to:

http://www.vmware.com/support/services

VMware Professional Services

VMware Education Services courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. Courses are available onsite, in the classroom, and live online. For onsite pilot programs and implementation best practices, VMware Consulting Services provides offerings to help you assess, plan, build, and manage your virtual environment. To access information about education classes, certification programs, and consulting services, go to http://www.vmware.com/services.

Using the Workbench Log Browser

Use the Workbench Log Browser Editor in the Workbench GUI to analyze logs for debugging purposes.

NOTE View cheat sheets for Workbench Log Browser from **Help > Cheat Sheets**. Expand **VMware vLogBrowser** and select the desired cheat sheet.

The document contains the following topics:

- "Using the Workbench Log Browser Wizard to Create a Project File" on page 7
- "Recording a Log Browser Session" on page 8
- "Using the Workbench Log Browser GUI" on page 9
- "Opening an Existing .logx Project" on page 10
- "Displaying and Using the Activity Graph" on page 10

Using the Workbench Log Browser Wizard to Create a Project File

To use the Workbench Log Browser GUI, you must first create a project with one or more `.logx` files in it. A `.logx` file describes what kind of log files to display and where to find them. Use the Workbench Log Browser Wizard to create the project and populate it with one or more `.logx` files.

Create a Workbench Log Browser Project

1 In the Project Explorer, right-click on any existing development kit project.

If you do not want to associate this project file with a development project, right-click on a blank area in the Project Explorer pane to create a Workbench Log Browser project independent of any development kit projects.

2 Select **New > Log Analysis**.

The Workbench Log Browser wizard displays.

3 From the **Source host** drop-down menu, select the **Local** host or the name of the remote system where the logs you want to analyze reside.

If you do not see the host machine you want in the drop-down menu, add it to the list by clicking **Add**. The **New Connection** wizard displays, where you can add a host machine. When finished, select the newly added host from the drop-down menu.

4 Select a **Log Format** from the drop-down menu.

5 In the **Project Container** field, click **Browse** to select a file container name, or type the name for a new file container.

6 Click **Next** to view the **Log File Path Selection page**.

7 (Optional) If you do not want to take the default values displayed, or there are no default values displayed in the **Parent Directory** and **File pattern(s)** fields, then Click **Browse** to choose the **Parent Directory** for the log search.

Workbench Log Browser will search the directory shown in the **Parent Directory** field for all files matching the **File pattern(s)** regular expression.

If the **Show All Files** check box is not dimmed on this page, you can select the check box to view all log files in the directory tree instead of only the ones matching the pattern.

8 Click **Next** to view the **Summary** page where you can see the log formats and `.logx` file names that will be created.

9 (Optional) Deselect the **Create** check box next to a `.logx` file to remove it from the list of files the wizard will create.

The `.logx` files are all enabled by default.

10 (Optional) Deselect the **Open** check box for one or more `.logx` files if you do not want them to open automatically when the wizard finishes.

The **Open** check boxes are selected by default. If you deselected any of the files in the wizard, you can open them later. You can open a `.logx` file at any time by double-clicking it in the Project Explorer.

11 (Optional) Click **Back** to go back to the **Log Host and Format Selection** page and add more `.logx` files to the project.

By continuing to click Next, you can cycle through the following pages repeatedly: **Log Host and Format Selection**, **Log File Path Selection**, and the **.logx File Creation Summary**.

Each time you cycle through, you can create a new `.logx` file that is added to your Workbench Log Browser project. The summary page displays all of the `.logx` files that will be created for this project.

12 Click **Finish** to create the project and return to the Project Explorer.

You see the project in the Project Explorer. Expanding the project displays all of the `.logx` files associated with the project.

13 (Optional) Double-click the `.logx` file in the project to open the logs in a Workbench Log Browser view.

This step is not necessary if you selected the **Open** check box. You see the logs automatically displayed in a Log Browser view in the Workbench GUI.

Recording a Log Browser Session

You can create a recording of some or all of the actions you take in the log browser. The recording is stored as a shell script (`lb-script.sh`) that can be replayed from a command line, edited, or embedded in another script. For more information on the command-line, see Appendix A, “Log Browser Command Line,” on page 13.

Start and stop recording

1 Start the recording of log browser actions by clicking the red start recording icon. 

2 In the **Select a file to save generated script contents** pop-up window, either accept the default name (lb.-script.sh) and location (inside your project folder), or choose your own name and location, and click OK.

If you are attempting to overwrite an existing script with the same name and location, a warning pops up. Click **OK** to continue overwriting an old script, or click **Cancel** to rename or relocate the script.

The icon changes to the stop recording icon.

3 Stop the recording after you have performed the log browser actions you want to record by clicking the stop recording icon:



Using the Workbench Log Browser GUI

The **vLogBrowser** pane displays formatted log entries. Each log entry is formatted into columns. For VMkernel logs, the column headers are: #, **Date**, **CPU**, **World ID**, and **Entry**. For `hostd` and other logs the column headers are: #, **Date**, **Source** and **Entry** (log text).

In the log browser, you can perform the following tasks:

- Double-clicking an entry causes a dialog box to open, displaying other text associated with the log entry. Click **OK** when you are ready to dismiss the dialog box.
- Filter for specific keywords by typing the keyword into the unlabeled text box on the top left of the Workbench Log Browser view and clicking **Filter**. The button is a toggle. If you wish to return to the unfiltered view of the log, click **Unfilter**.
- Before clicking **Filter**, if you want to see the log entries immediately surrounding the entry with the keyword, select the number of surrounding entries to display from the numerical drop-down menu.
- Sort the log entries by any column by clicking the column header once to sort in ascending order and click again to sort in descending order. Restore the original order by clicking the number ("#") column header to get ascending order.
- Synchronize log entries in multiple log browsers by a specific timestamp. Select the **Sync** check box in each log browser that you want synchronized. Select a time stamped entry in one of the browsers. The other log browsers have the log entry closest to that timestamp selected.
- Click the <**log_file_name.logx**> tab at the bottom of the **vLogBrowser** pane to see the `.logx` configuration file (XML formatted).

  You can customize this log view by editing the XML in the `.logx` file. To see the changes, save, close and reload the file.

- Various actions are available in the right-click menu, from the task bar icons, or from the **vLogBrowser** menu in the VMware Workbench menu bar:

  a **Filter** – The **Keyword Filtering with a Named Filter** dialog box displays.

  In the **Choose a Filter Source** area, select one of the radio buttons: **This .logx file**, **Filters file** (and Browse for the file), or **Other open .logx files** (select the file from a drop-down list).

  To create your own named filter, type a name in the **Named Filters** field or select an already existing named filter from the drop-down menu. Edit or accept the keywords for the filter in the regular expression formatting fields that follow. If you want to include lines surrounding the filtered lines, choose the number to include in the **Include # Surrounding Lines** selector.

  To save your new named filter or edited filter, click Save.

  To delete the filter in the Named Filters field, click Delete Filter. To clear the regular expressions form, click Clear Form.

  To save the new or edited **Named Filter** and filter the `.logx` file, click Save and Filter. To unfilter the `.logx` file and clear the regular expression form, click Unfilter and Clear Form.

  b **Find** – The **Find by keyword** dialog box displays. Enter the keyword to search for in the **Keyword** text box. Click Find All to see all entries in the log browser containing the keyword selected. Click Hide to close the dialog box. Click Find next to find the next instance of the keyword in the log browser.

  c **Find Next** – When selected, takes you to the next entry containing the keyword you searched for previously. If you haven't selected keyword, it opens the **Find by keyword** dialog box.

  d **Find or Adjust Date**– The **Find by Timestamp or Adjust All Timestamps** dialog box displays.

  To find a specific timestamp, type the desired timestamp in the **Date Input** field and click Find to go to the first entry in the log with that timestamp.

Or you can construct the timestamp using the calendar and clock controls in the dialog box. Changes you make using the calendar controls are reflected in the timestamp shown in the **Date Input** field. For example, changing the month to `February` changes the timestamp month to `02`. For the time portion of the timestamp, highlight a section of the clock display and click the buttons on the right to increment or decrement the number. For example, highlight the hour, and click the bottom button to decrement the hour number.

To adjust all of the timestamps in the log, set the desired timestamp in the Date Input field, and click Adjust. You can revert to the original timestamp by clicking Clear Adjustment.

e **Highlight** – The **Highlight a keyword** dialog box displays. Type in the keyword to highlight and click Highlight. All instances of the keyword in the log are highlighted. Click Unhighlight to turn off the highlighting. Click Hide to close the dialog box.

To highlight keywords using the Highlight button on the Workbench Log Browser interface, type in a keyword in the unlabeled text box on the right side of the Workbench Log Browser view and click Highlight. The button is a toggle. If you wish to remove highlighting of the keyword, click Unhighlight.

f **Copy –** Copy the selected entry or entries to the clipboard. You can open a text editor and paste the entries into a document.

g **Refresh** – If additional log entries have been added to the log since you started analyzing it, you can select **Refresh** from the menu to bring those additional entries into the view. This works for logs on the local file system as well as logs from the ESXi hosts you are connected to.

h **View Log Info** – A pop-up window displays the list of logs and their locations that you currently have loaded in this Workbench Log Browser project.

i **Add Log Files** – Add local logs files to an existing Workbench Log Browser project (`.logx` file). The **Add local log files to table** dialog box displays. Browse to the log directory and select one or more logs to add. Click Add to add the files to the `.logx` file. The log display refreshes to include the new entries.

Opening an Existing .logx Project

You can open an existing log analysis file that you saved to the Project Explorer by expanding the project name and double-clicking the `.logx` file. The file opens in a Workbench Log Browser view.

Displaying and Using the Activity Graph

The log browser includes an activity graph view that graphs activity from multiple log files over time. This can be useful to quickly identify irregular activity, such as a system reboot or a peripheral that stops responding.

Open the Activity Graph

1 From the Window menu, select **Show View > Other**.

2 From the **Show View** pop-up dialog, expand **VMware**.

3 Select **Activity Graph** and click **OK**.

Display log files in the activity graph

1 Click the plus (“+”) icon at the top right of the view to open the **Add Log File to Graph** dialog box.

2 Select one or more log files that you want to display and click **OK**.

The dialog box displays the log files already open in Workbench Log Browser. Each log is represented by a separate time line with the time scale spread out on the horizontal axis.

3 Adjust the time scale on the horizontal axis in one of three ways:

   a In one of the time lines, click and drag the cursor across an area of the time line that you are interested in expanding. When you release the cursor, the area you highlighted is expanded to fill the entire time line. The time scale is adjusted on the bottom axis.

   b Select one of the **Zoom chart** links at the top of the activity graph: **1min**, **1hour**, **1day**, **1week**, **1month**.

   The time line at the bottom changes to display the activity during that time interval.

   c Click the **Zoom In** and **Zoom Out** buttons to uniformly expand the time scale or to shrink the time scale.



CAUTION There are upper and lower bounds to the zoom scale. You will receive a warning message if you try to Zoom In to less than 100 ms, or Zoom Out to more than 10 years.

4 Use the **Zoom Back** and **Zoom Forward** arrow buttons at the top of the activity graph to move backward and forward through previous zoom settings.

5 To scroll through the entire log, move the **Scroll chart** horizontal slider at the top of the activity graph view to the left or right.

6 Double-clicking on an individual line in an activity graph causes the corresponding entry in the log browser to be highlighted.

Log Browser Command Line

The Workbench Log Browser packages include a command-line client (`vmwb-lbcli`) and server (`vmwb-server`) that allow you to operate on logs, log bundles, and recorded log sessions from the command-line.

Using the command-line client, you can do the following tasks:

- Aggregate logs from multiple sources and combine them.
- Load a subset of log types from a log bundle.
- Count the number of log entries in a given log (unfiltered or filtered).
- Find log entries that match patterns.
- Print log entries to stdout or to output files.
- Print in various formats (`txt`, `xml`, `json`, `csv`).
- Adjust time stamps in logs.

This appendix covers the following topics:

- "Command-Line Client Use Cases" on page 13
- "Where to Find the Command-Line Client-Server" on page 14
- "Using the Command-Line with the Workbench GUI" on page 15
- "Using the Command-Line Client with the Standalone Server" on page 15
- "Permission Issues when Retrieving a Recorded Log Browser Session" on page 16
- "Command-Line API Reference" on page 16
- "Use Case Examples" on page 17
- "Known issues/Limitations" on page 21
- "Advanced Use Cases" on page 21

Command-Line Client Use Cases

This command-line client is meant to enable use cases such as the following:

- Using the log browser during headless automated testing.

  At whatever point is appropriate in your tests you can run a shell script that connects to a remote system, downloads certain types of logs and looks for patterns that indicate that failures or errors occurred.

- Creating scripts of repeated actions.

If you need to do a repeated sequence of actions on a given type of log, for example once for each of a number of different hosts, you can create a script that performs those actions for you, such as connecting to the host, downloading certain types of logs, and applying a filter to that log.

- Using the log browser as a qualification tool for known defects.

  Log analysis is often the primary means of identifying defects. It would be helpful to quickly identify if a log bundle matches a pattern that corresponds to a known defect. The log browser command-line enables this sort of analysis. You can load the log bundle and then apply filters and search the log for matching entries.

- Doing manual log analysis from the command line

  If the system in which you are doing log analysis is a purely command-line system, it is convenient to do log analysis activities like loading a log bundle, applying a filter and printing the results on the screen or to a file right from the command-line without needing to run any sort of GUI application.

Where to Find the Command-Line Client-Server

The Workbench Log Browser command-line client-server feature is available in the following packages:

- "Workbench Log Browser Update Site Plug-in for the Workbench Appliance" on page 14
- "Workbench Log Browser Native Eclipse IDE" on page 14

Workbench Log Browser Update Site Plug-in for the Workbench Appliance

The command-line client and server are automatically installed when you download and install the Workbench Log Browser plug-in in the Workbench GUI (**Help > Install New Software**). When you install the update site, it installs the command-line only log browser server (`vmwb-server`) and the Log Browser command line client (`vmwb-lbcli`). The executables are automatically added to your path in the virtual machine.

Workbench Log Browser Native Eclipse IDE

The install packages for the native Eclipse version of the Log Browser, are self-contained platform-dependent ZIP files. Each platform specific package includes the command-line only log browser server (`vmwb-server`) and the log browser command-line client (`vmwb-lbcli`), and incudes a full working instance of a subset of the Workbench tools (including the Log Browser GUI) that can run on any machine.

The advantage of this package is that it can be used on any pre-existing host of your choosing. The packages are lightweight in the sense that they require no installation. Each package is a ready-to-use archive that you extract to your preferred directory.

To be able to launch the log browser server or command-line client from any shell, you must add them to the path.



CAUTION Log Browser requires 64-bit Java for Mac and Linux and either 32-bit or 64-bit Java for Windows. The JVM is no longer packaged with the toolsuite. Download and install either a 32-bit (for Windows only) or 64-bit JVM for your platform from http://www.java.com/en/download/manual.jsp.

If you are using the 32-bit Windows JVM, be sure to set JAVA_HOME and your PATH locally. If you set the JAVA_HOME variable globally, 64-bit-reliant packages will not find the correct JAVA package.

The following is a list of the available platform-dependent packages:

- rcp-toolsuite-lin64-3.5.0.*.zip
- rcp-toolsuite-mac64-3.5.0.*.zip
- rcp-toolsuite-win32-3.5.0.*.zip or rcp-toolsuite-win64-3.5.0.*.zip

Using the Command-Line with the Workbench GUI

To use the command-line client, you can either enable the server from the Workbench GUI (**VMware > Preferences**) or start the server from the command-line. If you want to move back and forth from the client command-line to the Workbench GUI, enable the server inside the GUI.

The advantage of using the command-line client in the context of the Workbench appliance is the seamless interoperability between the command line client and GUI operations. Logs that you open in the GUI will be available in the command line, and the reverse is also true, that is, logs opened in the client are available in the GUI. In the GUI also, you can record log browser sessions to create scripts (which are just a sequence of log browser commands).

The disadvantage of using the client with the Workbench GUI is that the Eclipse GUI must be running, thus it cannot be started from a shell. Another perceived disadvantage might be that the Workbench uses a fixed port number for the server. This means there can only be one server instance running on a given host. This limitation can be overcome with the standalone server.

Enable the server from the Workbench GUI

1 In the Workbench GUI, select **Windows > Preferences** to open the Preferences page.

2 In the Preferences page, expand **VMware** and select **Web Server** to display the Web Server options.

3 Select the **Enable web server on start up** check box, click **Apply**.

4 (Optional) To check if the web server has started, click **Check Status**.

5 Click **OK** to exit the Preferences page.

Using the Command-Line Client with the Standalone Server

The new command-line only server (`vmwb-server`) can be used to run the server manually from any shell. The server must be provided a path to a location to use for a workspace, that is a place to cache log files that are read during the process of log analysis. This workspace location must be unique to this server instance and cannot be shared with other server instances or with the IDE.

When you start the command-line server, it dynamically allocates ports that clients use for communication. The server records these ports in a file in the current users home directory so that when the vmwb-lbcli command-line client starts, it can read these values and know what ports to connect to. This functionality is seamless from your perspective as a user.

Start the command-line server from the command line

1 Open a terminal inside the Workbench virtual machine.

2 The server can be started in the background, for example:

```
$ vmwb-server /path/to/your/workspace &
```

The IDE currently is configured to use a fixed default port of `12443` for the server. You can start the server at the command in a mode where it also uses this same default port by doing:

```
$ vmwb-server /path/to/your/workspace --default-port &
```

Of course this mode means that you cannot start more than one instance of this server on a given host since it uses the same port number.

If you cannot have the server on the default port, you can manually specify which port(s) you would like it to use. There are actually two ports, one for the web services (`12443` by default) and one for an administration interface (`12221` by default). In the example that follows, the command is given with a value of `0` (which causes the dynamic allocation of the ports) but you can change this value to anything you wish.

```
$ vmwb-server /path/to/your/workspace --https-port=0 --admin-port=0 &
```

The command-line client application for the current user (the user that started the server) will automatically figure out the ports to use.

Permission Issues when Retrieving a Recorded Log Browser Session

By default on the Workbench virtual appliance, the IDE is run as `root` even though you may be logged in as a normal user (for example, the `vmware` user). One side effect of this is that files that are created by the IDE, such as the recorded shell script/batch file, are owned by `root` by default. If you log into Workbench as `root`, you will not have any permission issues in retrieving the recorded log browser session script. However, if you have logged into Workbench as a regular user, such as `vmware`, you will lack permissions to do anything with the script. To fix this, use one of the following commands to change the permissions:

- Change ownership of the file.

  ```
  sudo chown vmware:vmware lb-script.sh
  ```

- Change permissions of the file so that user `vmware` can access it.

  ```
  sudo chmod o+rwx lb-script.sh
  ```

Command-Line API Reference

This section describes the command-line client API and contains the following topics:

- "Command Syntax" on page 16
- "Getting the List of Available Commands" on page 16
- "Getting Help on a Particular Command" on page 16

Command Syntax

The command-line client, `vmwb-lbcli`, has syntax consistent with tools like `git` or `p4` from the command line, that is, you give the executable name, a command, command specific option, and then command specific arguments, if applicable.

There are many options that are common to various commands such as `--project`, which specifies the name of the project you are operating on, as well as `--resource` which is the resource within the project. Many of these commands can inherit the values for the command from the environment, if certain environment variables are defined. For example, if you do not specify the `--resource` option for a command that requires it, the value of the `VMLB_RESOURCE` environment variable, if defined, will be used. See the help for particular commands to see which variables are applicable if you desire to use this facility.

Getting the List of Available Commands

```
$ vmwb-lbcli help
vmwb_lbcli <command> [options]

    define     Define an additional user provided logx file type to the log browser server.
    save       Save the specified filter.
    count      Print the number of log entries currently filtered in.
    select     Select a log to be loaded in the log browser server.
    adjust     Adjust log entry timestamps by the specified time value.
    delete     Delete bundle and cached files for the project host.  Optionally specify
               particular resource.
    download   Download and analyze a log file from the given host.
    print      Print the specified log entries to the console.
    find       Find the index of log entries that match the specified pattern.
    filter     Appy the filter with the specified name.  The filter must have already been added.
    getfilter  Get filters/a single filter if name specified.
    help       Display top level help information.
    version    Display program version number.
```

Getting Help on a Particular Command

For detailed help on a particular command, type the following:

```
vmwb-lbcli help <command>
```

Use Case Examples

The section lists some possible use cases and the client commands:

- "Printing Log Entries to the Console" on page 17
- "Downloading Log Files from a Remote Host for Analysis" on page 17
- "Discovering Logs on a Host" on page 18
- "Downloading All Logs for a Host" on page 18
- "Getting the Count of the Total Number of Visible Log Entries" on page 18
- "Printing Log Entries to stdout" on page 18
- "Applying a Simple Filter" on page 18
- "Exporting Entries to a File" on page 19

Printing Log Entries to the Console

Print the specified log entries to the console.

```
vmwb_lbcli print [options]

   --count <arg>    Optional max count of the number of entries to display.  Defaults to 30.
                    You may also specify 'all' in order to fetch all log entries.
   --first <arg>    Optional 1 based entry number of the first entry to display.
   --format <arg>   Optional format for the output.  Must be one of json,xml,txt.  Default is
                    txt.
   --lbserver <arg> The host name of the log browser server to connect to.  If not provided
                    looks for env variable VMLB_SERVER, and then defaults to localhost.
   --noextra        If given, suppress printing of 'extra' lines in the output, leaving only
                    the first line of a given log entry as the value emitted.  Default is to
                    emit the extra lines.
   --nofilter       If given, the printed results are the UNFILTERED log entries.  By default
                    the entries printed are a result of the currently applied filter.
   --outfile <arg>  Redirect the output to the specified file instead of stdout.
   --project <arg>  The name of the project.  If not provided looks for env variable
                    VMLB_PROJECT.
   --resource <arg> The name of the resource.  If not provided looks for env variable
                    VMLB_RESOURCE.
   --verbose        Verbose display of results if available.
```

Downloading Log Files from a Remote Host for Analysis

The `download` command allows you to connect to a remote host, collect a particular type of log file from it, and open it for analysis:

```
$ vmwb-lbcli download --host=192.168.1.106 --htype=esx --user=root --project=test --type=hostd
[Enter password for connection:]
```

The command shares the workspace of the server it connects to, and by implication it also shares all projects, log analyses, and cached target credentials that the server has.

In this example, the user had not connected to this target yet, and because the command did not explicitly specify the password with the `--password` switch, it prompted the user for it.

For subsequent commands, the password does not have to be entered again because the credentials are cached in the servers secure storage. In the following example, a later download of a different log file type, it is not necessary to specify the user name and password. The cached ones from the connection above are used.

```
$ vmwb-lbcli download --host=192.168.1.106 --htype=esx  --project=test --type=VMkernel
$
```

After downloading the files, if you have the Workbench IDE running, you will see projects that have been created as a result of these commands. The projects include the server it connects to, as well as a new remote systems target for the host that was specified in the command-line.

Discovering Logs on a Host

If you do not know the types of logs that your remote system supports, the `download` command can give you some guidance. Omitting the `--type` argument causes the system to return an error message that lists the allowable types:

```
$ vmwb-lbcli download --project=myProj --host=192.168.1.106 --htype=esx --user=root
               --password=VMwar3rules
ERROR: Log type must be supplied.  Allowable values for this host include
               vmware,VMKernel,hostd,shell,syslog,vpxa,fdm,VMKwarning.
```

Downloading All Logs for a Host

The `download` command supports `--type=all` as a way to load all known log types for the specified host:

```
$ vmwb-lbcli download --project=myProj --host=192.168.1.106 --htype=esx --user=root
               --password=VMwar3rules --type=all
Loaded 8 log file types to project myProj:
    vmware                  192.168.1.106_vmware.logx
    VMKernel                192.168.1.106_VMKernel.logx
    hostd                   192.168.1.106_hostd.logx
    shell                   192.168.1.106_shell.logx
    syslog                  192.168.1.106_syslog.logx
    vpxa                    192.168.1.106_vpxa.logx
    fdm                     192.168.1.106_fdm.logx
    VMKwarning              192.168.1.106_VMKwarning.logx
```

Getting the Count of the Total Number of Visible Log Entries

The `count` command gives you the count of the number of log entries that are visible with the current filter applied (if there is one). There is an additional `--nofilter` command line option that will give you the unfiltered count if you wish to get that without removing the filter. In the example that follows, the number that shows below the command is the answer returned by the system.

```
$ vmwb-lbcli count --project=test --resource=192.168.1.106_hostd.logx
471686
```

Printing Log Entries to stdout

The `print` command, used to print log entries, is flexible in terms of the format of the reporting (`txt`, `xml`, `json`), as well as the number of entries it prints and the starting entry. You must specify at least the project and resource within the project as arguments to the command. Other arguments, such as formatting, are optional.

The following is an example of the default behavior, which is to print the first 30 entries matched by the currently applied filter, if one exists. In this example, there is no filter applied.

```
$ vmwb-lbcli print --project=test --resource=192.168.1.106_hostd.logx
1 |  |  | Section for VMware ESX, pid=2853, version=5.0.0, build=build-623860, option=Release |
               ------ In-memory logs start --------
2 | 2012-07-13 03:23:48.100 -0600 | FFE2BAC0 info 'Default' | Supported VMs 63 |
<snip...>
30 | 2012-07-13 03:23:48.104 -0600 | FFE2BAC0 info 'Default' | Trying hbrsvc |
```

Applying a Simple Filter

The `filter` command supports two kinds of filters: defining a simple filter dynamically with the `--pattern` option (a `regex` pattern), or applying a known named filter. In the following example, a simple filter is applied, the count of filtered entries is retrieved, and the entries are printed in XML format:

```
$ vmwb-lbcli filter --project=test --resource=192.168.1.106_hostd.logx --pattern="Event 157"

$ vmwb-lbcli count --project=test --resource=192.168.1.106_hostd.logx
2

$ vmwb-lbcli print --project=test --resource=192.168.1.106_hostd.logx --format=xml
<logItems>
```

```
   <logItem>
     <_>411049</_>
     <Date>2012-08-03 16:25:32.215 -0600</Date>
     <Source>47B01B90 info 'ha-eventmgr'</Source>
     <Entry>Event 157 : User root logged out</Entry>
     <EntryExtraLines/>
   </logItem>
   <logItem>
     <_>471663</_>
     <Date>2012-08-03 16:25:32.215 -0600</Date>
     <Source>47B01B90 info 'ha-eventmgr'</Source>
     <Entry>Event 157 : User root logged out</Entry>
     <EntryExtraLines/>
   </logItem>
 </logItems>
```

Exporting Entries to a File

The `print` command is used to export entries. This can be done many different ways depending on your needs. Several examples are given.

stdout Redirection

The normal `print` command directs its output to `stdout`. You can redirect this to a file. This particular example also demonstrates the use of the `--count` and `--format` options:

```
$ vmwb-lbcli print --project=test --resource=192.168.1.106_hostd.logx --count=1 --format=json >
               myOutput.json
$ cat myOutput.json
{
    "logItems": [
        {
            "#": "411049",
            "Date": "2012-08-03 16:25:32.215 -0600",
            "Source": "47B01B90 info 'ha-eventmgr'",
            "Entry": "Event 157 : User root logged out",
            "EntryExtraLines": ""
        }
    ]
}
```

The --outfile Option

The `print` command also has an `--outfile` option for directly specifying the output file. This option has the advantage of being able to display a percentage complete status in the console as the download of the entries progresses. This may be of interest in the case of very large log files. The following example downloads a large number of log entries. (The `--count=all` option was used to specify all log entries.)

NOTE In this example, the filter was first removed with the `filter` command before printing. The `--nofilter` option on the `print` command could also have been used to do this without removing the filter.

```
$ vmwb-lbcli filter --project=test --resource=192.168.1.106_hostd.logx --remove
$ vmwb-lbcli print --project=test --resource=192.168.1.106_hostd.logx --format=xml --count=all
               --outfile=myFile.xml
Downloading log entries: 100% complete
Done writing to file /home/aspear/myFile.xml
downloaded 471686 entries in 62.909s (7497 entries/sec)
```

Finding Particular Entries and Printing Them

As an alternative to filtering, you can also use the `find` command to search for particular patterns. The `find` command by default returns the log entry number of the matching entries:

```
$ vmwb-lbcli find --project=test --resource=192.168.1.106_hostd.logx --pattern="Event 157"
411049
471663
```

You can print the entries in any given format using the `--print` option:

```
$ vmwb-lbcli find --project=test --resource=192.168.1.106_hostd.logx --pattern="Event 157"
              --print=txt
411049 | 2012-08-03 16:25:32.215 -0600 | 47B01B90 info 'ha-eventmgr' | Event 157 : User root
              logged out |
471663 | 2012-08-03 16:25:32.215 -0600 | 47B01B90 info 'ha-eventmgr' | Event 157 : User root
              logged out |
```

Finding Particular Entries and Failing if the Count is Not What You Expect

Both the `find` and `count` commands support a couple of additional options that can be useful for use cases such as automated regression testing:

```
   --fail_if_count_equal <arg>     Optional integer argument to check resulting line count
                                   against.  If the count is equal to this number the command
                                   fails.  Mutually exclusive with the fail_if_count_not_equal
                                    option.
   --fail_if_count_not_equal <arg> Optional integer argument to check resulting line count
                                   against.  If the count is NOT equal to this number the command
                                   fails. Mutually exclusive with the fail_if_count_equal
                                    option.
```

In the following example, `Failure` is used in a particular log file in the case of any error, and there are 17 instances of `Failure` that are known issues and always emitted in the log. As a part of an automated regression test, this log file could be checked and would fail if the count is more than 17 (failing in this case means that the `vmwb-lbcli` process returns an exit code of non-zero in addition to printing an error message to `stderr`).

```
$ vmwb-lbcli find --project=test --resource=192.168.1.106_hostd.logx --pattern="Failure"
              --fail_if_count_not_equal=17
1162
6916
(... 17 entry line numbers printed here)
464835
ERROR: Count 18 is not equal to fail_if_count_not_equal value 17.
```

Printing the Contents of a Local Log Bundle File

The `bundle` command allows printing of information on the contents of a log bundle, as well as loading all or a subset of the contained log file types in the bundle.

The following example prints the available log types in a bundle:

```
$ vmwb-lbcli bundle /home/aspear/logfiles/gss/vcsupport-2011-10-26.16479.zip
Examining log bundle /home/aspear/logfiles/gss/vcsupport-2011-10- 26.16479.zip ...
Available log types:
    lookup Server
    imsTrace
    eam
    stats
    vpxd
    jointool
    ssoAdminServer
    sso-service-cfg
    SRM Recovery
    Virgo Server
    install
    cim-dia
    SRM Protected
    sms
```

Loading Log Types from a Log Bundle

You can load log types from a bundle by adding one or more `--load` options to the `bundle` command. The `load` options specify the log types you wish to load. The command requires that you must specify a project name that you wish to load the given log types into. The following is an example of loading a couple of log types:

```
$ vmwb-lbcli bundle --project=test2 --load=vpxd --load=stats
               ~/logfiles/gss/vcsupport-2011-10-26.16479.zip
Loading log bundle /home/aspear/logfiles/gss/vcsupport-2011-10-26.16479.zip to project test2 ...
Loaded 2 log file types:
    stats                   Local_stats_vmsp.logx
    vpxd                    Local_vpxd_vmsp.logx
```

You can also specify `--load=all` to load all types in the bundle:

```
$ vmwb-lbcli bundle --project=test2 --load=all ~/logfiles/gss/vcsupport-2011-10-26.16479.zip
Loading log bundle /home/aspear/logfiles/gss/vcsupport-2011-10-26.16479.zip to project test2 ...
Loaded 14 log file types:
    lookup Server           Local_lookup_Server_vmsp.logx
    imsTrace                Local_imsTrace_vmsp.logx
    eam                     Local_eam_vmsp.logx
    stats                   Local_stats_vmsp.logx
    vpxd                    Local_vpxd_vmsp.logx
    jointool                Local_jointool_vmsp.logx
    ssoAdminServer          Local_ssoAdminServer_vmsp.logx
    sso-service-cfg         Local_sso-service-cfg_vmsp.logx
    SRM Recovery            Local_SRM_Recovery_vmsp.logx
    Virgo Server            Local_Virgo_Server_vmsp.logx
    install                 Local_install_vmsp.logx
    cim-dia                 Local_cim-dia_vmsp.logx
    SRM Protected           Local_SRM_Protected_vmsp.logx
    sms                     Local_sms_vmsp.logx
```

After the bundle is loaded, normal operations can be done against the file, such as finding, printing, and filtering, by referencing the project and resource names (printed in the previous example):

```
$ vmwb-lbcli print --project=test2 --resource=Local_vpxd_vmsp.logx --count=1 --first=10
10 | 2011-10-25 23:46:56.666 -0600 | EAFFF700 info 'ThreadPool' | Thread enlisted |
```

Known issues/Limitations

The log browser server must be running in order for the command to work.

Currently only local log bundles can be loaded. There is no facility to download a remote a log bundle from a Windows host, and also no integration to request a host to create a log bundle from the CLI. If you can copy the Windows log bundles to your localhost, then they can be analyzed.

Recordings don't actually record the content of filters that are defined. They record the name of the filter and the activation of filters, but rely on the filter being persisted on the server side.

Advanced Use Cases

It is possible to connect to a log browser server running on any machine you wish. By default the assumption is that the server is connected to `localhost`, but the `--lbserver` option can be used to connect to a server on a different host. Related to this, you can also specify the port to connect on as a part of the host name, for example `--lbserver="myServer.eng.vmware.com:12444"`. Workbench Web Services are fixed on port `12443`.

Index

C

copy selected entries **10**
creating a project file using the wizard **7**

F

filter keywords **9**
find keywords **9**

H

highlighting keywords **10**

L

log browser **7**
- .logx file **9**
- action buttons **9**
- sorting by column **9**
- using **9**
- viewing additional text **9**
- viewing surrounding entries **9**
- viewing the XML **9**

T

technical support resources **6**

U

unfilter keywords **9**

V

vLogBrowser
- copy entries **10**
- highlighting **10**

W

wizard **7**
Workbench Log Browser **7**
- filter and unfilter for keywords **9**
- find a keyword **9**
- using **9**

X

XML file, log browser **9**