MUSE

Muse Proxy Install

21 June 2013

Document Version 0.0.1.1
Muse 2.6.0.0
Muse Proxy 3.0 Build 03

Notice

No part of this publication may be reproduced stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of MuseGlobal Inc.

Disclaimer

MUSEGLOBAL, INC. MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

Trademarks

MUSE IS A REGISTERED TRADEMARK OF MUSEGLOBAL, INC. OTHER PRODUCT NAMES AND SERVICE NAMES ARE THE TRADEMARKS OR REGISTERED TRADEMARKS OF THEIR RESPECTIVE OWNERS AND ARE USED FOR IDEN TIFICATION ONLY.

www.museglobal.com

One Embarcadero, Suite 500
SAN FRANCISCO, CA 94111 U.S.A.
415-896-6873 FX 415-896-6870 SUPPORT@MUSEGLOBAL.COM

Table of Contents

© 2006-2013 MUSEGLOBAL INC

1.0

Overview

Muse Proxy is a highly customizable program that has many domains of applicability like proxy server, reverse proxy, rewriting proxy, navigation manager. It can be used by libraries to give access from outside their computer network to restricted-access websites that authenticate users by IP address, or by companies to give access to their employees from outside their computer network to internal resources.

© 2006-2013 MUSEGLOBAL INC

2.0

Muse Proxy Install

Muse Proxy can run on any platform on which a Java Virtual Machine can be installed.

2.1 Hardware Requirements

The following are the minimum and recommended hardware specifications for running Muse Proxy. For this purpose Muse Proxy will be capable of handling a minimum of 50 simultaneously connected users with a quick response processing time.

Minimum specification:

- 20 GB hard disk space;
- 512 MB RAM;
- Intel Pentium III (or equivalent) or better processor;
- Processor clock speed of 1 GHz or better (for Pentium and equivalent processors).

Recommended specification:

- 40 GB hard disk space;
- 2 GB RAM;
- Intel Pentium 4 (or equivalent) or better processor / multiprocessors;
- Processor clock speed of 3 GHz or better (for Pentium and equivalent processors).

2.2 Software Requirements

The following environment software products are necessary to operate Muse Proxy. Where a suitable operating system is in use, it is not necessary to have a dedicated computer to run Muse Proxy. It can exist together with other application software on a single computer running a suitable operating system.

All the environment software listed below has been tested with Muse Proxy, and supports it fully. This does not imply that Muse Proxy is not fully functional on other environments not listed down here. If the environment you need to install Muse Proxy on is not listed here, please contact the Muse Proxy providers for further details.

- `Operating system`

  The following operating systems have been tested with Muse Proxy:

  - Microsoft Windows 2000, Windows XP, Windows Server 2003, Windows 7;
  - Sun Solaris x86 , v. 8-10;
  - Linux (x86) – various flavors: Red Hat, Debian, Suse, Mandriva, Free BSD, Caldera, Slackware, Gentoo, Knoppix.

Note: The only shells that are currently supported are: `csh`, `bash` (for UNIX environments) and Windows command prompt (for Microsoft Windows environments). The provided UNIX scripts may also work with `tcsh`, `ash`. However, these are not tested and may not work as expected.

- `Java Virtual Machine Software Development Kit(JDK).`

  MuseGlobal recommends the installation of a Java Virtual Machine Software Development Kit (SDK, or also referred to as JDK) package because they contain tools not available in the Java Runtime Environment (JRE) package that can be used for monitoring and troubleshooting issues related to the Java Virtual Machine and the Muse Proxy software. For example, the `jstat` tool available in a JDK package displays performance statistics for an instrumented HotSpot Java Virtual Machine (JVM). The Java Virtual Machine Software Development Kit contains the JRE.

  The following virtual machines and platform combinations have been tested with Muse Proxy:

  - Intel x86/Win32 (Windows 2000 / Windows XP / Windows Server 2003 / Windows 7), JDK 6;
  - Intel x86/Solaris, JDK 6;
  - Intel x86/Linux, JDK 6.

Note: IT is recommended to install JDK 1.6 since it was throughly tested. Note that JDK version must be 1.6 or higher, Muse Proxy does not work with JDK 1.5 or lower. You must obtain the Java Virtual Machine only from SUN or IBM. Muse Proxy works only if the Java Virtual Machine is from Sun or IBM. Note that there is no IBM JDK for Windows platform.

Vendor	URL Address
Sun	http://www.oracle.com/technetwork/java/javase/downloads/index.html

© 2006-2013 MUSEGLOBAL INC

IBM	http://www.ibm.com/developerworks/java/jdk/

Windows Platform Installation

Starting from the web address indicated above download the appropriate `Windows (all languages, including English)` version - either for a 32 bit architecture or 64 bit. The downloaded file is a self-extracting archive presented as an executable file. After running the archive the only information for the installation process which will be required is the directory location in which the JVM should be placed.

If you run into any problems during the installation, please check the `Installation Instructions` section at:
`http://www.oracle.com/technetwork/java/javase/install-windows-189425.html.`

Some Java programs require the definition of a supplementary environment variable called `JAVA_HOME` pointing to the JDK installation. If this is required, introduce the following, assuming the destination directory is `c:\jdk1.6.0` :

```
set JAVA_HOME=c:\jdk1.6.0
```

Setting the `JAVA_HOME` variable is mandatory on the Windows 2003 Server platform.

On some Windows distributions there were encountered cases when the Muse Proxy Windows service fails to start and an error message like below is found in the Event Viewer:

```
The LoadLibrary function failed for the following reason: The specified
module could not be found.
```

The fix consists in adding into the `Path` system environment variable the path to the Java `bin` folder. For example, if your Java installation folder is `C:\Program Files\Java\jdk1.6.0_27` then add the following string to the `Path` environment variable: `C:\Program Files\Java\jdk1.6.0_27\bin` . Note that the separator between the entries from the `Path` variable is the semicolon (;) . Details for how to update the Path variable for some Windows distributions can be found here: http://www.java.com/en/download/help/path.xml

UNIX Platform Installation

There are separate JDK Environments for Linux and Solaris. Each is accessible following the appropriate link from the web address indicated above: `Linux self-extracting file` or `Solaris SPARCTM/x86 self-extracting file.`

If you run into any problems during the installation, check the `Installation Instructions`

section: `http://www.oracle.com/technetwork/java/javase/install-142943.html` .

For Solaris, there are two versions of JDK available, according to the installation platform: SPARCTM or IntelTM x86 platform.

If you run into any problems during the installation, check the `Installation Instructions` section:
`http://www.oracle.com/technetwork/java/javase/install-solaris-139361.html`.

If you install the JDK Environment into a system-wide location such as `/usr/local`, you must first become `root` to gain the necessary permissions. If you do not have `root` access, install the JDK Environment into your home directory or a subdirectory that you have permission to write to.

Some environment variables such as `PATH` and `CLASSPATH` (indicating the place where the Java files were installed) must be modified or set to use JDK efficiently.

Although this is not a strictly Java-related issue any longer, some programs require the definition of a sup plementary environment variable called `JAVA_HOME` pointing to the JDK installation. Introduce the following, assuming the destination directory is `/usr/java` you should type:

```
export JAVA_HOME=/usr/java/jdk
```

Add the line above in the file `/etc/profile` to make this setting permanent on Linux or Solaris. The variable becomes globally available after the next login.

2.3 Install Muse Proxy

Muse Proxy is installed as a standalone product using Muse Proxy Setup. You need a License Key File that has enabled the entry associated with the Muse Proxy product.

Although the Muse Proxy Setup installation package can be installed on virtually any platform that supports Java, there are a number of platform-specific issues that keep this platform-neutral installation package from providing a simple and reliable user experience.

Muse Proxy Setup installation package provides platform packs for:

Launcher Name	Platform Distributions
`muse-proxy-linux-x86.bin`	Red Hat, Debian, Suse, Mandriva, Free BSD, Caldera, Slackware, Gentoo, Knoppix
`muse-proxy-solaris-x86.bin`	Sun Solaris x86 v. 8-10

 © 2006-2013 MUSEGLOBAL INC

`muse-proxy-win32.exe`	Microsoft Windows 2000, Windows XP, Windows Server 2003, Windows 7
`muse-proxy-generic-unix.bin`	Generic Unix; should be used for UNIX platforms where no other specific install packages are available
`muse-proxy-setup.jar`	Java Virtual Machine 1.6 or above on the supported platforms

Muse Proxy can run on virtually any platform that supports Java. If the platform you need for your Muse Proxy installation is not listed above, please contact your Muse Proxy provider for further details.

It is recommended to install Muse Proxy as root (on Unix/Linux systems) or Administrator (on Windows systems) to avoid performing additional steps for registering the Muse Proxy services. This also allows you to bind Muse Proxy to ports lower than 1024, such as the standard HTTP port 80 and HTTPS port 443.

The following issues should be noted if you are installing Muse Proxy as a normal user (without `root`/ administration rights):

The user installing Muse Proxy must have all permissions in the parent directory that are required to create the Muse Proxy installation directory and all the Muse Proxy files. By default Muse Proxy in stallation directory is `/opt/muse/proxy`. For example, if the user John is installing Muse Proxy under the `/opt/muse/proxy` directory, then John must have write and execute rights on the `/opt` directory. If rights to the parent directory cannot be given due to security reasons, change the Muse Proxy in stallation directory to a directory where rights can be assigned.

The lack of `root`/administrator rights may generate errors in the Muse Proxy Service Panel File. These can be ignored as the Muse Proxy Service can be installed later. After installing Muse Proxy as a normal user (without `root`/administration rights), run Muse Proxy Service Setup as a `root/administrator` user (see Section 2.3.6, “Running Muse Proxy Service Setup”). The Muse Proxy installation is started in different ways, depending upon your operating system and hardware architecture.

`Muse Proxy Setup` will automatically look for an acceptable Java Virtual Machine for the installation. The launcher will terminate if none of the recommended Java Virtual Machine can be found.

As an alternative to native platform launchers, a Java `.class` file you can execute from the command line is compressed in an appropriate `muse-proxy-setup.jar` file. Use this installation method if none of the recommended Java Virtual Machines can be found and the launcher terminates.

This alternative installation can run in graphical mode using Swing by default, or in console mode.

The graphical user interface (GUI) consists in a series of panels that display to the user. To start the Muse Proxy Setup in graphical mode using the Swing interface use the following command line:

```
java -cp muse-proxy-setup.jar run
```

(or alternatively `java -jar muse-proxy-setup.jar`*)*

When the `Muse Proxy Setup` wizard is run in console mode, all messages and information are sent to the console instead of appearing on graphical panels. While running a wizard in console mode, you can move forward in the wizard by pressing `Enter`. To start the Muse Proxy Setup in console mode use the following command line:

```
java -cp muse-proxy-setup.jar run -console
```

Command Line Options available for all types of installations (graphical, console, silent , and from native executable):

`-log !<log-file> @ALL` Logs all the events in the specified log file.

Note: On some UNIX shells the character `!` is interpreted by the shell and is not passed to the program. For those shells this character must be escaped by replacing it with `\!`.

`-Dis.debug=1` Specifies to display all debug messages that occur at runtime on the Java console. For example:

```
java -Dis.debug=1 -cp muse-proxy-setup.jar run
```

2.3.1 Native Launcher Installation

A native launcher wraps the Java archive file inside a platform-specific executable file.

Starting the native launcher acts the same as launching the command:

```
java -jar muse-proxy-setup.jar -options-record
<user_home>/muse-proxy-options.txt -options
<user_home>/muse-proxy-options.txt
```

Where `<user_home>` will be replaced with the path being the home directory for the current user.

For successful Muse Proxy installation an options file will be recorded and will preserve user's selections and input data. This file will be used and recreated for successful upgrades in the future.

Note: The options above require write access in the local directory where the Muse Proxy options file is to be created.

Note: If an options file existed previously under `<user_home>/muse-proxy-options.txt`, it will be deleted when Muse Proxy Setup starts and will only be written upon a successful installation.

 © 2006-2013 MUSEGLOBAL INC

Command Line Options Available for Native Launchers:

- `-is:log <file name>` is useful for native launchers that hide the Java console. When this option is specified, all output that occurs in the Java console is logged to the specified file.
- `-is:version` reports the version of the native launcher itself. When this option is specified, the launcher simply reports the version and exits without launching the application.
- `-is:javaconsole` makes the Java console visible, overriding the value specified for the `"Show Console"` property when the launcher was built. Always use this option along with `-console` when you want to run the Muse Proxy Setup in console mode using the native launcher.
- `-is:tempdir <directory>` allows the end user to specify a directory to which the launcher will write its temporary files. If the specified location either does not exist or is not a directory, the launcher will use the system temp directory instead, and will not give an error message. If this option is used for a launcher that executes a wizard (either an installer or an uninstaller), then the wizard will use the same directory for its temp files. The temporary directory used by the launcher must have all permissions for the user running the Muse Proxy Setup (`rwx` for UNIX/Linux systems or Full Control on Windows systems). The temporary directory is usually located in `"C:\Documents and Settings\<username>\Local Settings\Temp"` for Windows systems and `"/tmp"` for UNIX/Linux systems.
- `-is:silent` prevents the display of the launcher UI to the user. This does not execute the wizard itself in silent mode.
- `-is:javahome <Java home directory>` allows the end user to specifically tell the launcher the home directory location of the JVM. This option works only when executing a launcher for an installer or uninstaller wizard. In order for it to work, the JVM installed in the specified directory must be one of the JVMs specified in the "JVM Search Instructions" property of the launcher distribution.

2.3.2 Graphical Mode Wizard

To start the Muse Proxy Setup in graphical mode use the following command line:

```
java -cp muse-proxy-setup.jar run
```

(or alternatively `java -jar muse-proxy-setup.jar`*)*

For Windows systems:

To run this package run:

- `"muse-proxy-win32.exe"` for graphic mode.

For Linux systems:

Before running the setup package make sure you have executable rights for it, otherwise use the `chmod` command for changing the permissions:

"chmod u+x muse-proxy-linux-x86.bin".

Then use the following command in a Linux command prompt:

- "./muse-proxy-linux-x86.bin" for graphic mode.

For Solaris systems:

Before running the setup package make sure you have executable rights for it, otherwise use the `chmod` command for changing the permissions:

"chmod u+x muse-proxy-solaris-x86.bin".

Then use the following command in a Solaris command prompt:

- "./muse-proxy-solaris-x86.bin" for graphic mode.

To advance in the Muse Proxy installation procedure click the `Next` buttons. A `Back` button is available to review previous steps. The `Cancel` button, when available, cancels the installation process.

Muse Proxy Setup is very simple to use Setup program built in a Wizard-like fashion, thus to advance to the next step of the installation you must read the information on screen, eventually adjusting the default configuration options, then pressing the `Next` button in order to advance.

Once the last step is reached, the user has to press the `Finish` button to close the Setup. After Muse Proxy was successfully installed you can start it and access Muse Proxy Welcome page:

```
URL: http://localhost:{proxy_port}
```

Or you can start administer it by accessing Muse Proxy Console:

```
URL: http://localhost:{proxy_port}/admin
User: administrator
Password: the password you set
```

where `{proxy_port}`is the port on which Muse Proxy listen.

This information is listed on the last Muse Proxy Setup panel as well:

 © 2006-2013 MUSEGLOBAL INC

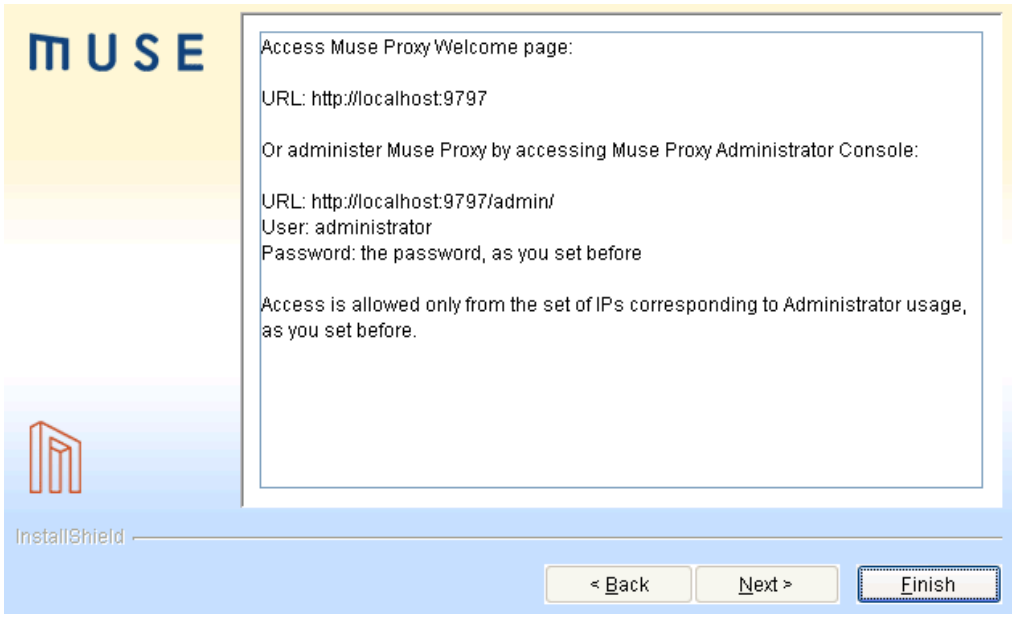


Figure 1. Muse Proxy Setup – Finish panel – Windows and Linux Systems

The most important steps of the Setup are:

- The panel that asks you for License Key Panel - any License Key File belonging to you that has the entry for Muse Proxy activated will be accepted here.

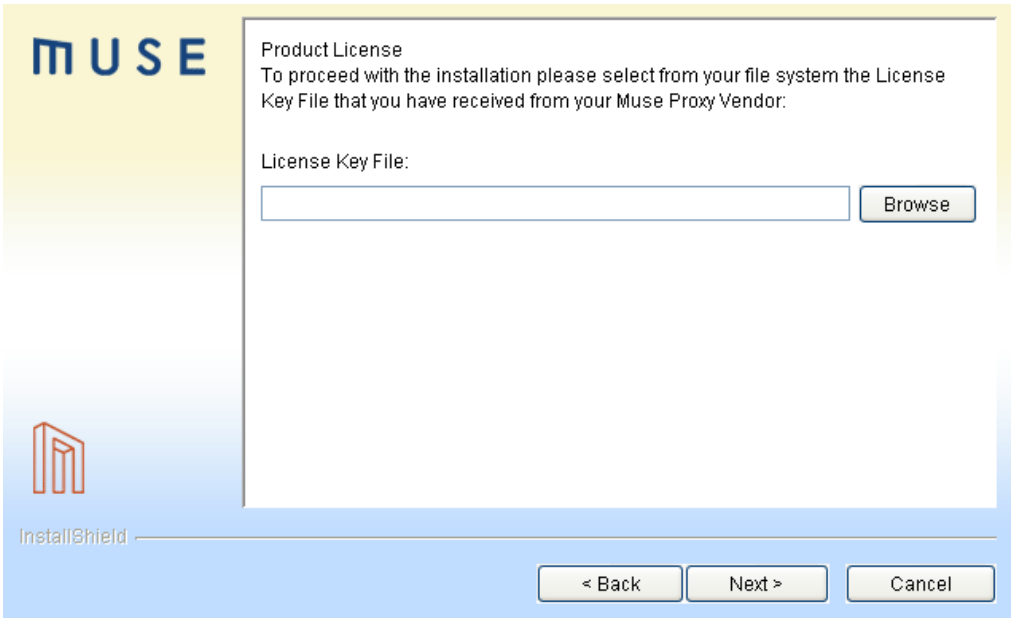
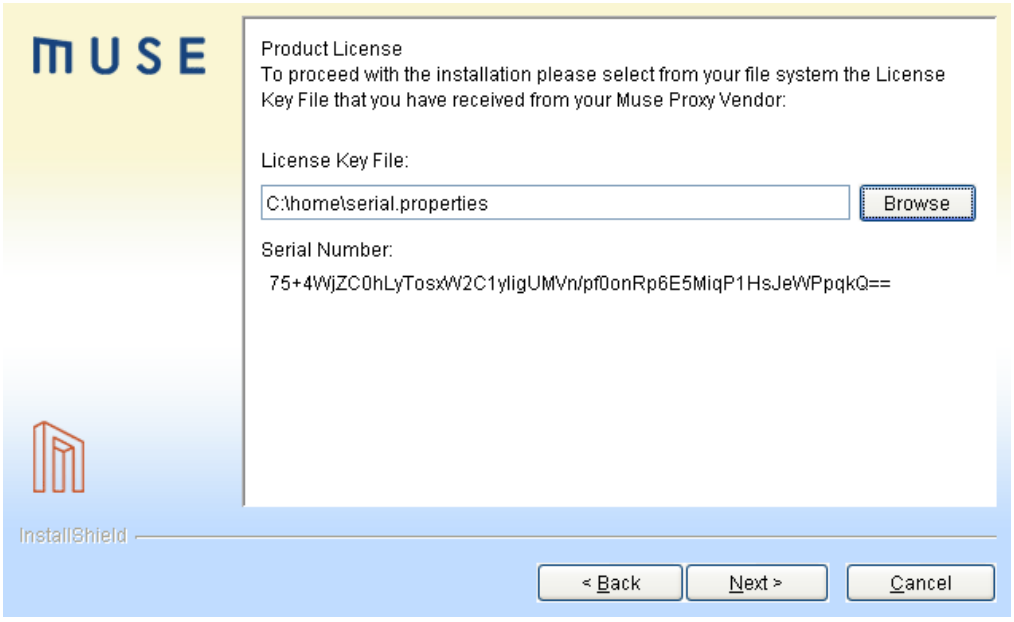


Figure 2. Muse Proxy Setup – License Key panel – Windows and Linux Systems

Once you click **Browse** button and select the License Key File that you received from your Muse Proxy Vendor this panel will look like below:



© 2006-2013 MUSEGLOBAL INC

Figure 3. Muse Proxy Setup – License Key panel – Valid License Key File – Windows and Linux Systems

where Serial Number is an encrypted string used to continue the Muse Proxy installation.

If the License Key File that you selected is not a valid one, (the Serial Number cannot be detected), this panel will look like below and you have to contact your Muse Proxy distributor for more details.

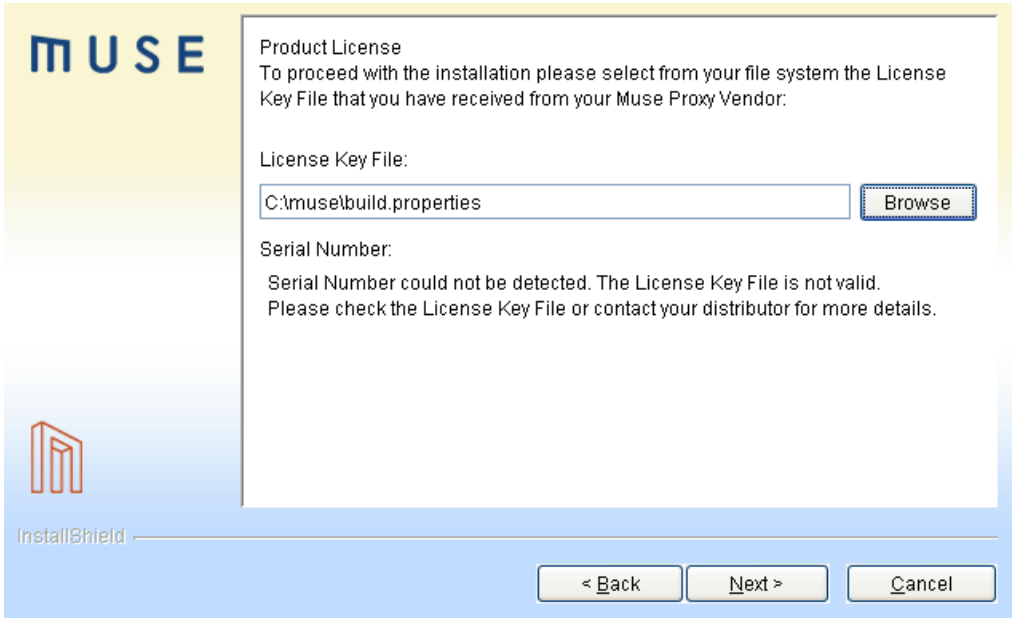


Figure 4. Muse Proxy Setup – License Key panel – Valid License Key File – Windows and Linux Systems

- The panel that asks you for the Muse Proxy installation directory:

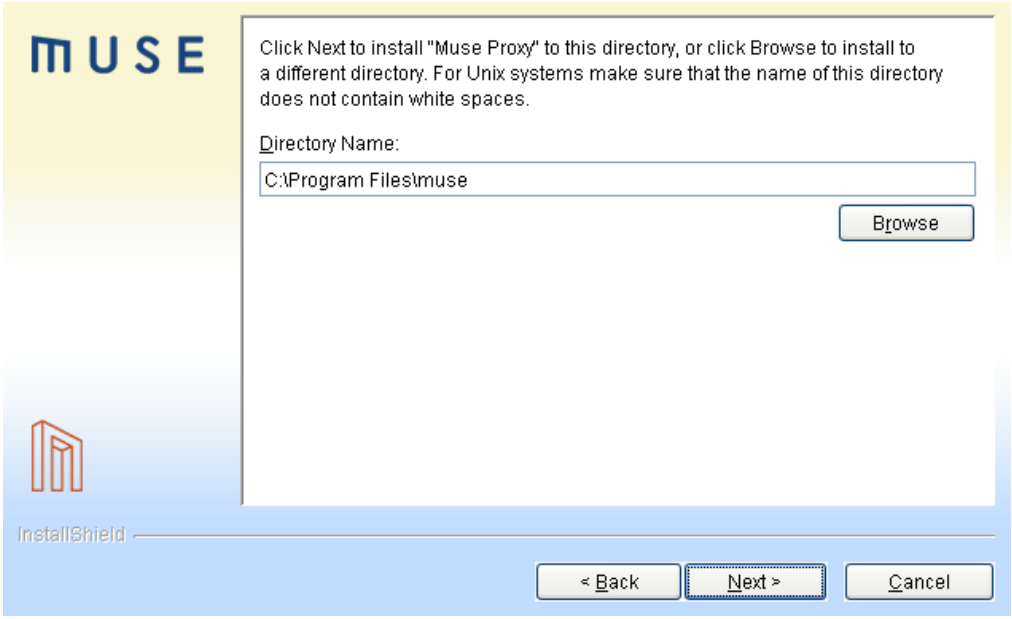


Figure 5. Muse Proxy Setup – Installation Directory – Windows Systems

Note: On Unix systems, you must make sure that the name of the directory where Muse Proxy is installed does not contain any white space characters.

You can choose an existing path in your system, or just use the default one.

- The panel that let you choose the type of installation:

- `Typical`: The `Typical` installation selects all licensed Muse Proxy Setup components by default. This is selected by default.
- `Custom`: The `Custom` installation allows the user to select the Muse Proxy Setup components (these are high level components) to install, as applicable for their licensed Muse Proxy products.

  Note: A `Typical` installation is recommended.

© 2006-2013 MUSEGLOBAL INC

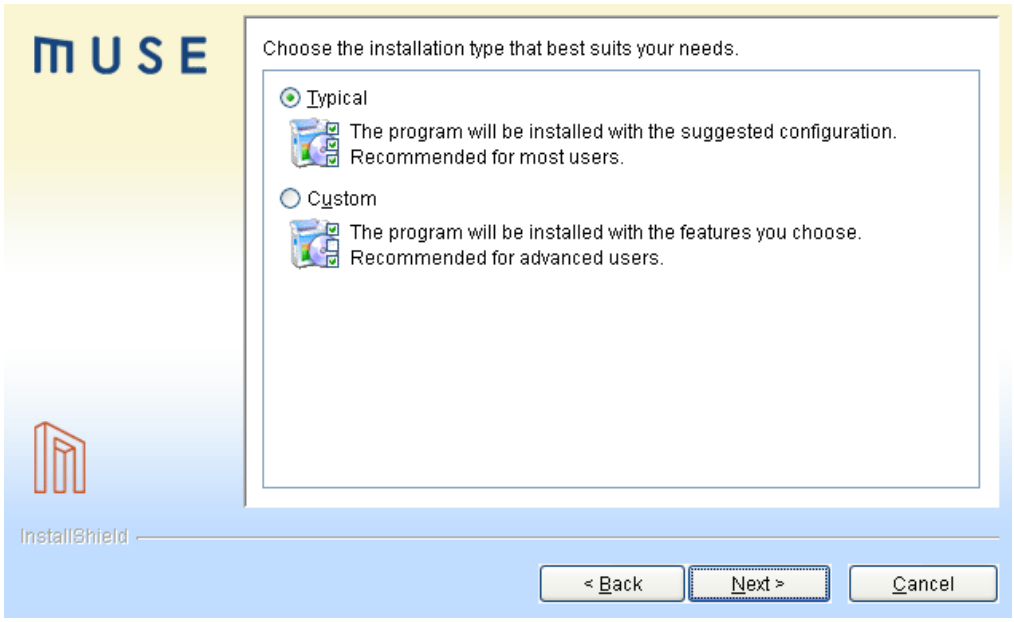


Figure 6. Muse Proxy Setup – Choose Installation Type – Windows and Linux Systems

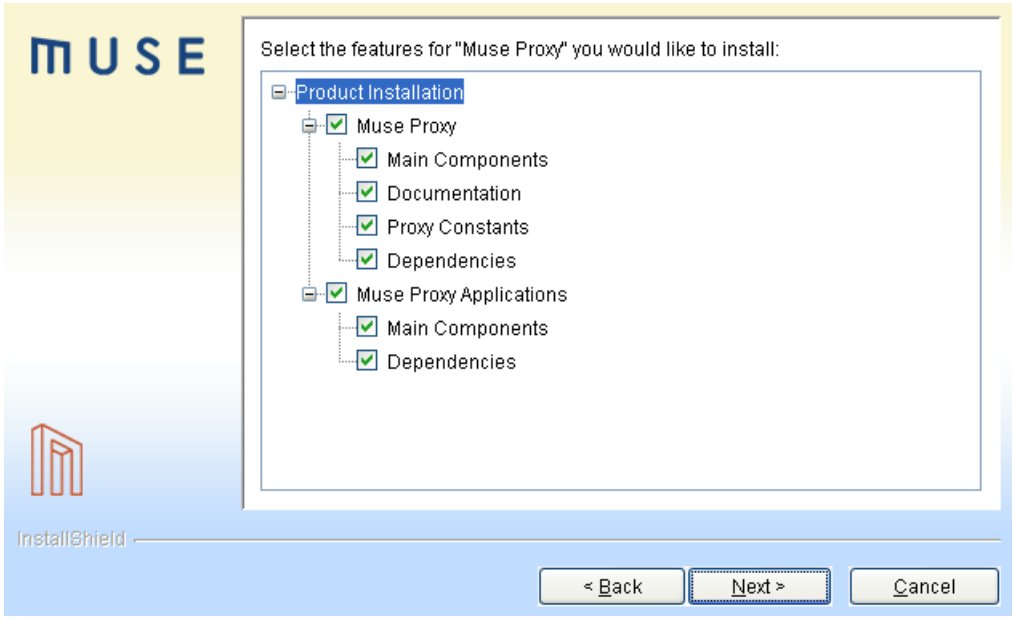


Figure 7. Muse Proxy Setup – Custom Installation Type – Windows and Linux Systems

Currently, `Muse Proxy Setup` package includes the following Muse Proxy products:

- `Muse Proxy`: Muse Proxy is a highly customizable program that has many domains of ap plicability like proxy server, reverse proxy, rewriting proxy, navigation manager. It can be used by libraries to give access from outside their computer network to restricted-access websites that au thenticate users by IP address, or by companies to give access to their employees from outside their computer network to internal resources.
- `Muse Proxy Applications`: Muse Proxy Applications provide a fully configurable interface allowing administrators to setup remote or local Data Sources ("Sources") and where end users enjoy a single sign-on for all of the subscribed data. Library Administrators can add Sources to the Muse Proxy Application depending of licensing rights. Each Source stores the necessary code and credentials to get to the target Data Source. In this way, Muse Proxy enables end users to reach all configured targets with a single sign-on and conceals target authentication credentials.

  Note: To install Muse Proxy Applications you must have a License Key File having the entry associated with this product enabled, otherwise Muse Proxy Applications will not be installed.

  Note: If you do not have Muse Proxy Applications licensed, in the `Custom Installation Types` panel, there will be no checkboxes associated with Muse Proxy Applications.

- The panel with a short summary of the Muse Proxy Setup components to be installed:

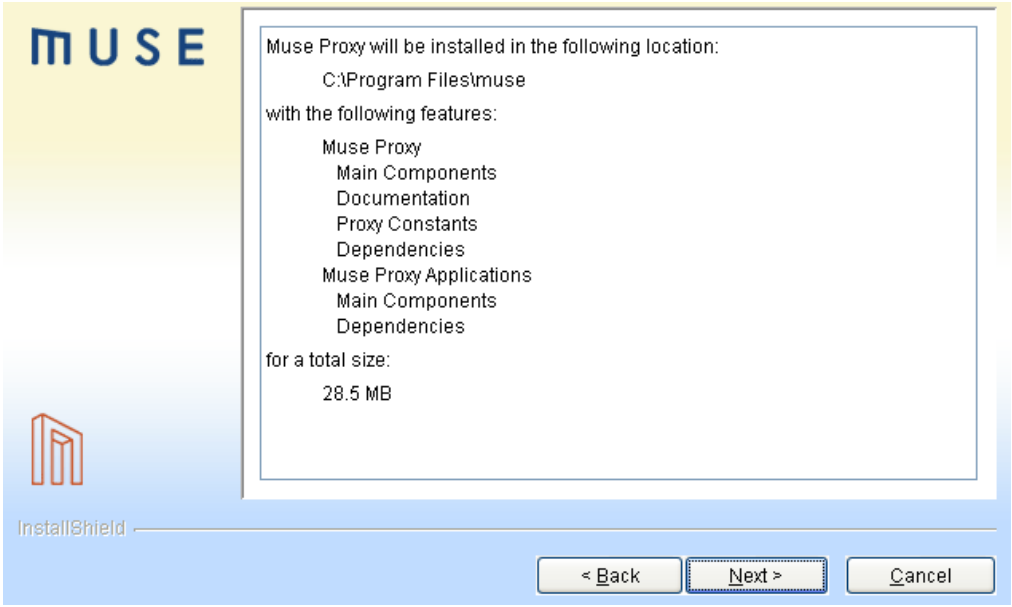


Figure 8. Muse Proxy Setup – Custom Installation Type – Windows Systems

© 2006-2013 MUSEGLOBAL INC

-The panel that lets you initially configure the set of IPs you want Muse Proxy to be accessed from. Start with a secure, yet comprehensive set corresponding to Administrator usage and end-user usage. You will later be able to set differently and selectively the IP rules for various usage from the Muse Proxy Ad ministrator Console. The following users will be allowed to logon/connect to Muse Proxy exclusively from the IPs you add here: `"administrator"`, `"services"`, `"navigationManager"`, `"default"`, `"guest"`, that can connect and use Muse Proxy.

The set of the vendo's secure IP(s) are automatically added. Also the entire set of the IPv6 and IPv4 associated with the machine Muse Proxy is installed on are automatically added.

Note: Make sure that the IPs you add are meaningful for your places you want to administer and test from as only browsers originating from the IPs in this list will be allowed to connect to Muse Proxy Ad ministrator Console and Muse Proxy Welcome page.

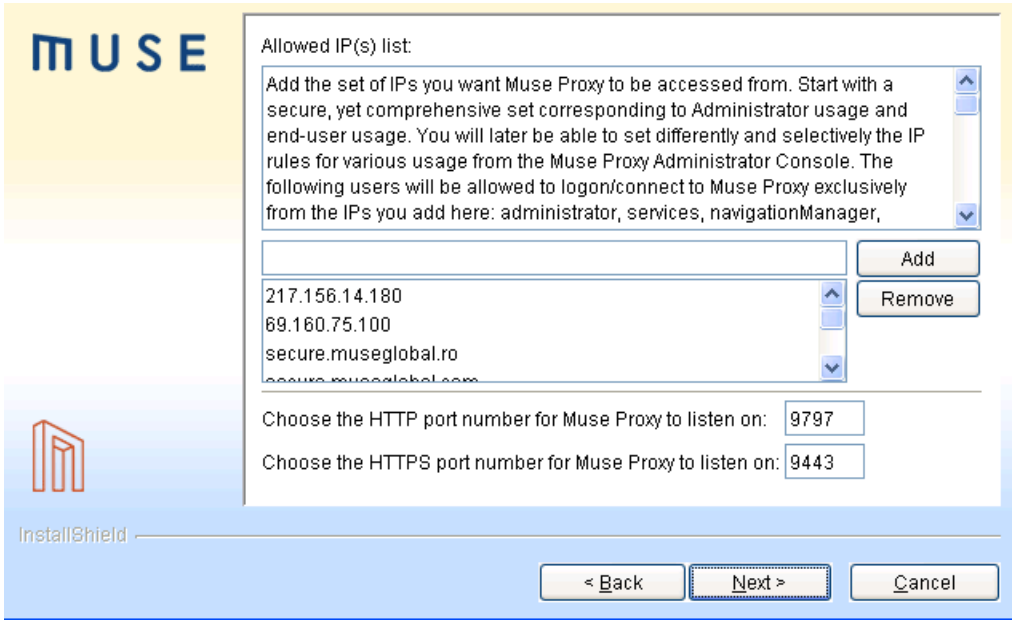


Figure 9. Muse Proxy Setup – Allowed IP List for Muse Proxy access – Windows and Linux Systems

Note: Besides changing the list of allowed hosts, you can also change the default HTTP/HTTPS port for Muse Proxy. In case the port(s) you chose is/are already in use, the Muse Proxy Setup will notice you.

Note: This panel only appears for fresh installations of Muse Proxy, it does not appear for upgrades.

- The panel that summarizes your actions in the previous panel:

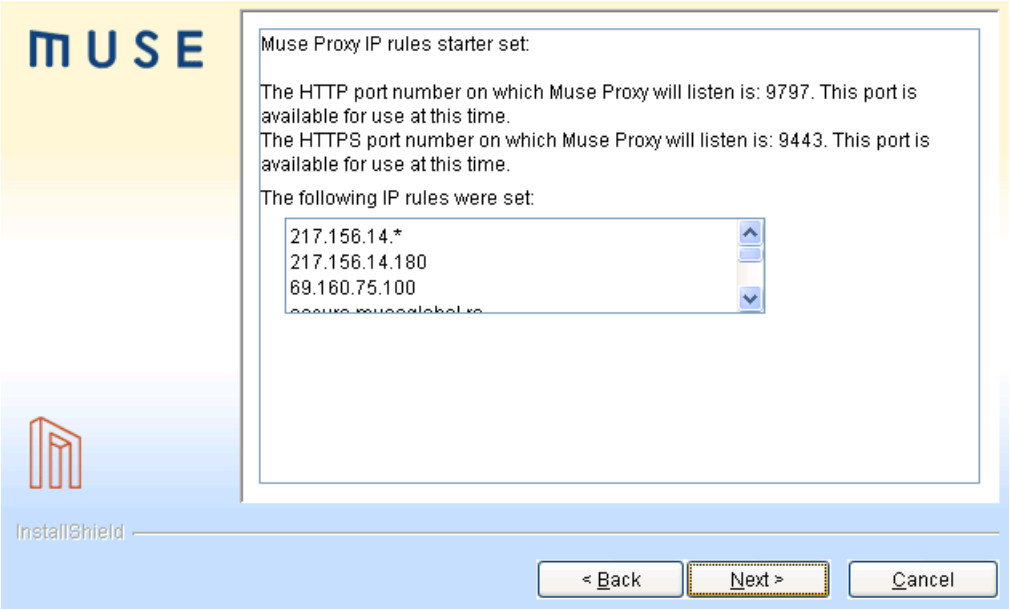
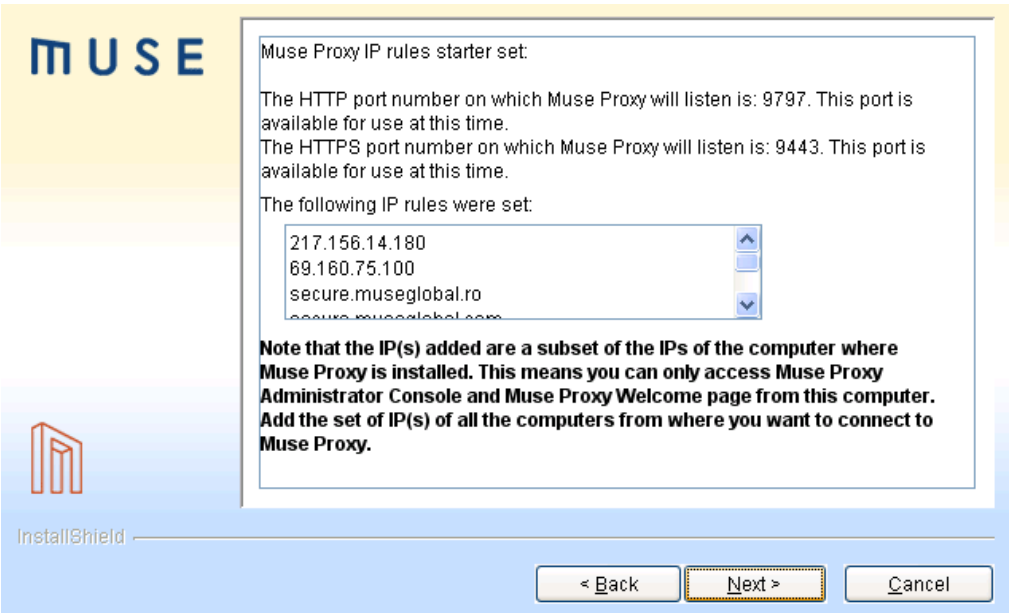


Figure 10. Muse Proxy Setup – IP rules Starter Set – Windows and Linux Systems

If, in the previous panel, you did not add IPs an warning like in the picture from below appears. You must add the IP(s) of all the computers from where you want to connect to Muse Proxy.



© 2006-2013 MUSEGLOBAL INC

Figure 11. Muse Proxy Setup – IP rules Starter Set – Warning Message – Windows and Linux Systems

- The panel that asks you to type in the password for Muse Proxy users: `"guest"`, `"administrator"`, `"services"`, `"navigationManager"`, `"control.jmx"`, `"monitor.jmx"`. At this point it will be the same password for all users. Later you can change it from Muse Proxy Administrator Console.

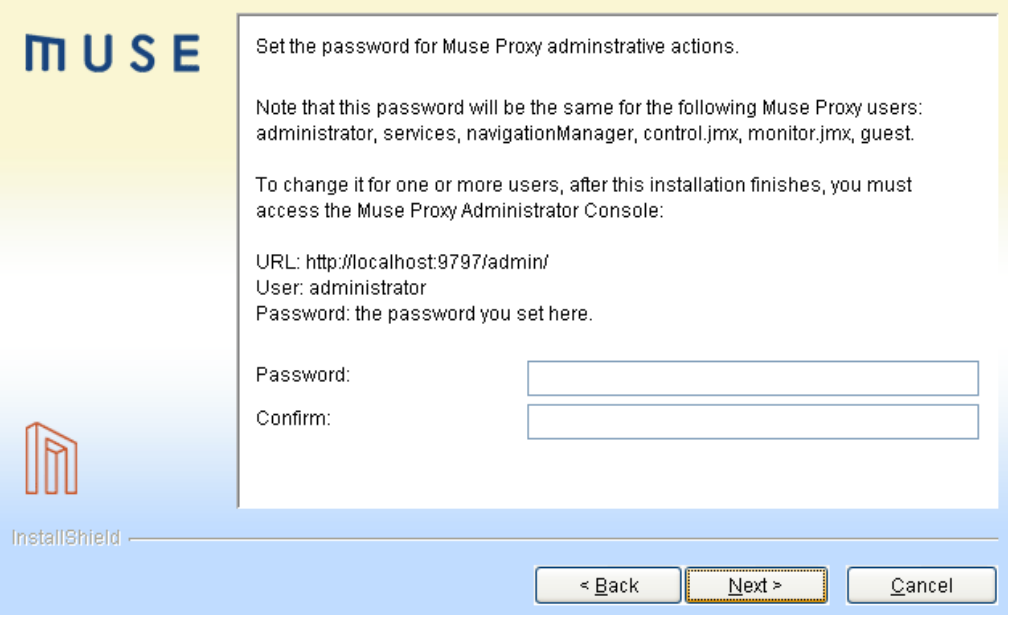


Figure 12. Muse Proxy Setup – Passwords for Muse Proxy Users – – Windows and Linux Systems

Note: This panel only appears for fresh installations of Muse Proxy, it does not appear for upgrades.

- The service installation panel:

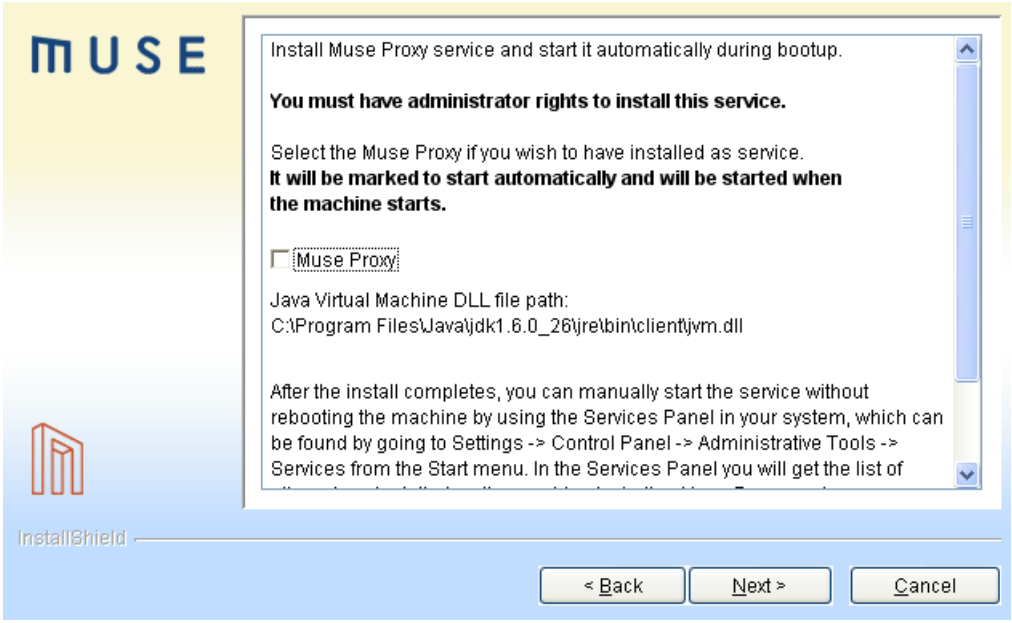


Figure 13. Muse Proxy Setup – Service Installation – Windows Systems

This panel will allow you to install (by checking the shown checkbox) Muse Proxy as a System Service (this will set Muse Proxy to be started automatically when your machine is booted).

- On the last panels you can start Muse Proxy:

 © 2006-2013 MUSEGLOBAL INC

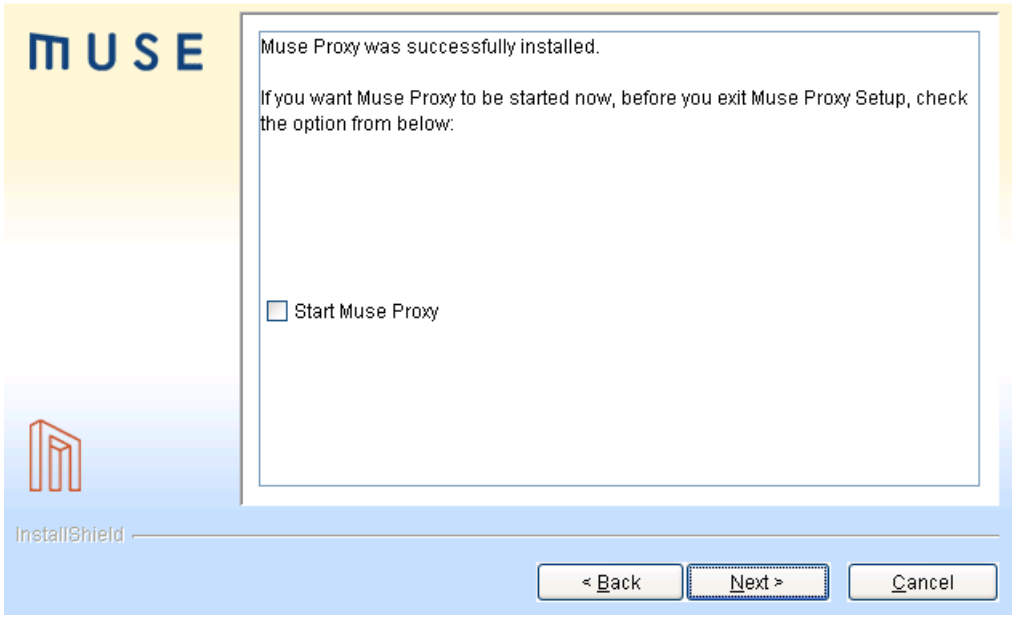


Figure 14. Muse Proxy Setup – Start Muse Proxy

2.3.3 Console Mode Wizard

To start the Muse Proxy Setup in console mode use the following command line:

```
java -cp muse-proxy-setup.jar run -console
```

The console mode translates into text the data presented in graphical panels. While running the wizard in console mode, you can move forward in the wizard by pressing `Enter` key.

Follow the on screen instructions and read the equivalent Graphical Mode instructions.

For Windows systems:

To run this package, use the following command in a Windows command prompt:

- `"muse-proxy-win32.exe -console"` for console mode.

For Linux systems:

Before running the setup package make sure you have executable rights for it, otherwise use the `chmod` command for changing the permissions:

`"chmod u+x muse-proxy-linux-x86.bin".`

Then use the following command in a Linux command prompt:

- "`./muse-proxy-linux-x86.bin -console`" for console mode.

For Solaris systems:

Before running the setup package make sure you have executable rights for it, otherwise use the `chmod` command for changing the permissions:

"`chmod u+x muse-proxy-solaris-x86.bin`".

Then use the following command in a Solaris command prompt:

- "`./muse-proxy-solaris-x86.bin -console`" for console mode.

2.3.4 Silent Mode Wizard

To start the Muse Proxy Setup in silent mode, use the following command line:

```
java -jar muse-proxy-setup.jar -options muse-proxy-options.txt -silent
```

The silent mode install does not interact with the user unless an error occurs. If an error occurs, a message displays and the Muse Proxy Setup exits.

The silent mode install uses a pre-built `muse-proxy-options.txt,` which contains the options needed by the wizard to run, just like the user would introduce them.

Note: This mode will override the built-in path (`<user_home>/muse-proxy-options.txt`) of the options file with the path that you supply.

To start the Muse Proxy Setup, use the following command line to generate the options file:

```
java -jar muse-proxy-setup.jar -options-record muse-proxy-options.txt
```

Note: This will override the built-in `path (<user_home>/muse-proxy-options.txt)` of the options file with the path that you supply at this time.

This runs a normal Muse Proxy Setup where all the installation options are recorded into the `<options file>` as specified.

There is an additional parameter regarding the creation and use of options files.

```
java -jar muse-proxy-setup.jar -options-template muse-proxy-options.txt
```

 © 2006-2013 MUSEGLOBAL INC

All the properties that can be specified to the wizard will be written in the specified file. This file is a template file and the user must set the proper values for the properties. Later this file can be used as an input options file.

Note: All the options above require write access in the directory where the options file will be created.

2.3.5 Options File Method

The Muse Proxy Setup package gives the possibility of recording (by the means of an options file) any input that was supplied during an installation. Once such an options file is produced, it can be used in other further installations of Muse Proxy (even on other computers) in an automatic and even silently manner.

- `Producing the options file`

By default, any method of installing Muse Proxy (native launchers or running the setup from the `.jar` file) produces an options file at the following location: `<user home>/muse-proxy-options.txt`. The `<user home>` will be replaced with the path which is the home directory for the current user doing the installation.

You can overwrite this default path of the options file and have the options recorded at a different location other than the default `<user home>/muse proxy-options.txt` by giving a command line parameters with a different path of the options file. To generate an options file at a location you specified you can run the setup with the `options-record` command line option:

```
java -jar muse-proxy-setup.jar -options-record <desired path to options file>
```

Note: The option above requires write access into the local directory where the options file will be created.

Note: If an options file existed previously under `<desired path to options file>`, it will be deleted when Muse Proxy Setup is started and will only be written on a successful installation.

- `Running the setup using an options file`

This allows you to re-use the selections you made at a previous Muse Proxy installation. To use this feature you need to run the Muse Proxy Setup with the `options` command line option: `java -jar muse-proxy-setup.jar -options <path to options file>`

By default, when you run the Muse Proxy Setup without the `options` command line parameter Muse Proxy Setup will try and load an options file from the `muse-proxy-options.txt` file in the user's

home directory: `<user home>/muse-proxy-options.txt.`

Example of using the options file

An example of using the options file is when Muse Proxy needs to be installed on many machines with the same options (same installation directory, same IP addresses that are allowed through the proxy, etc.). Using the options file can greatly automate the process of installing Muse Proxy. Combined with other options allowed for the Muse Proxy Setup program, such as the `silent` option, this is a really powerful tool to automate the installation of Muse Proxy.

First run the setup to create the options file:

```
java -jar muse-proxy-setup.jar  options-record muse-proxy-options.txt
```

Install the Muse Proxy as normal and every configuration you make during the installation will be recorded in the options file. After successfully completing the installation you will find in the current directory a file named `muse-proxy-options.txt` that contains the options you made during in stallation. If you want the same options to be used on a different machine, you will have to copy on the machines that you need to install Muse Proxy, both the Muse Proxy Setup and the `muse-proxy-options.txt` options file. When running the Muse Proxy Setup on these machines you will need to specify the options file to be used during installation. Additionally since all the options are already supplied in the options file you can add the `silent` command line option to minimize the interaction you need to have with the Setup during installation. `silent` gives no user interaction, and all the options are read from the options file. So, on the other machines you need to run the Muse Proxy Setup as:

```
java -jar muse-proxy-setup.jar  options muse-proxy-options.txt  silent
```

Wait for the install to be completed and unless any errors are displayed Muse Proxy is successfully installed.

2.3.6 Running Muse Proxy Service Setup

If `Muse Proxy Service` was not set during `Muse Proxy Setup` installation, you can still set it up to be automatically started when the system boots up, by running the Muse Proxy Service Setup. Muse Proxy Service Setup may also be used to remove Muse Proxy Service from automatic startup. Ad ministrator rights are required to manage Muse Proxy Service.

Under a graphical environment Muse Proxy Service Setup can be started from Start menu (`Start/Programs/Muse/Muse Proxy Service Setup`).

`Muse Proxy Service Setup` can be started by running `C:\Program`

 © 2006-2013 MUSEGLOBAL INC

`Files\muse\proxy\setup\startMuseProxyServiceSetup.bat` on a Windows machine, assuming the Muse Proxy is installed under the `C:\Program Files\muse\proxy` directory, or by running `/opt/muse/proxy/setup/startMuseProxyServiceSetup` on a UNIX machine, assuming the Muse Proxy is installed under the `/opt/muse/proxy` directory.

If you want to perform Muse Proxy Service Setup in console mode, run the files mentioned above with the `-console` parameter. E.g.:

```
C:\Program Files\muse\proxy\setup\startMuseProxyServiceSetup.bat -console
```

When `Muse Proxy Service Setup` is started in a graphical mode, a panel like the one from below, displays.

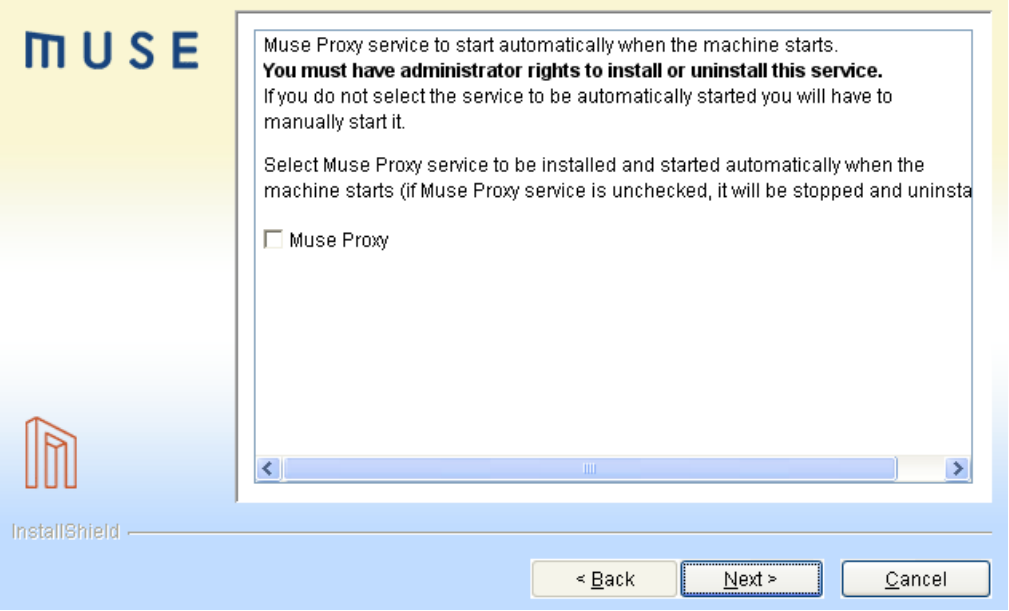


Figure 15. Muse Proxy Service Setup – Windows and Linux Systems

Check (or uncheck) the checkbox near the Muse Proxy Service to indicate whether it should (or should not) start automatically when the system boots up. Click the `Next` button to save the new settings.

Note: The user must have administrator rights to set up the Muse Proxy Service. If the Muse Proxy Setup determines this is not the case, the user will receive information on how to install Muse Proxy Service. The user will need administrator rights to successfully complete Muse Proxy Service Setup.

Note: For Muse Proxy Service to correctly work on Windows 64-bit systems you must either install

Muse Proxy using the `muse-proxy-setup.jar` file or you can use the executable file (`muse-proxy-win32.exe`) to install Muse Proxy but in this case you must set `JAVA_HOME` environment variable to the location of the 64-bit Java. In case, there are both 32-bit JAVA and 64-bit JAVA when installing Muse Proxy using `muse-proxy-win32.exe` file and no `JAVA_HOME` is defined, the 32-bit JAVA will be used and Muse Proxy Service will not work correctly. Otherwise, if `JAVA_HOME` environment is set to the location of the 64-bit JAVA then the JAVA used will be 64-bit JAVA even if you installed Muse Proxy using `muse-proxy-win32.exe` file and on the Windows system there are both 32-bit JAVA and 64-bit JAVA.

The following steps must be performed to set up Muse Proxy Service after Muse Proxy has been installed as a normal user:

1. On Unix Systems:
   - Log in as `root;`
   - In the root's home directory (you should be there after `root` login), create a symbolic link for the Install Shield directory. This is located under the home of the user used for initial Muse Proxy Installation. To do this, launch the following command, substituting `${USER_HOME}` with the full path to the home of the user used for initial Muse Proxy installation:

     ```
     ln -s ${USER_HOME}/InstallShield InstallShield
     ```

   - Launch the following shell script, substituting `${MUSE_HOME}/proxy` with the full path to the Muse Proxy home directory (e.g. `/home/john/muse/proxy`):

     ```
     ${MUSE_HOME}/proxy/setup/startMuseProxyServiceSetup.sh
     ```

2. On Windows Systems:
   - Log in as `administrator` (or as an user having administrator rights);
   - Give administrator rights to the user used for the initial Muse Proxy installation (For example on Windows XP: `Control Panel -> User Accounts -> <<User Name>> -> Change the account type -> Select computer administrator.`;
   - Logoff;
   - Log in as the user used for the initial Muse Proxy installation;
   - Go to the `setup` subdirectory in Muse Proxy installation directory and launch the Muse Proxy Service Setup - `startMuseProxyServiceSetup.bat`;
   - Remove the `administrator` rights for this user and logoff.

2.4 Upgrade Muse Proxy

Muse Proxy can be upgraded either using the Muse Proxy Setup or either by installing a Muse Proxy

 © 2006-2013 MUSEGLOBAL INC

Patch, the latter being mostly used between two Muse Proxy Releases.

When a different Muse Proxy Setup than the one originally installed on your machine is run, the user will go through an upgrade procedure. This is similar to the first-time installation, but when Muse Proxy is installed in the same location more than once, the product components can determine whether or not to replace existing files. During the upgrade you may be prompted to replace or keep some files that were changed during the life of the old installation.

The Muse Proxy packages will overwrite your existing Muse Proxy files. Muse Proxy configuration files are left untouched, so the settings you have added will be safeguarded.

Once the upgrade procedure has been completed, continue with the instructions from the last panel of the Muse Proxy Setup. This lists the manual instructions which must be followed when upgrading from an old to a newer Muse Proxy release version.

Additional steps may be required when upgrading between certain release versions, but these instructions will be provided separately.

Note: It is highly recommended that Muse Proxy Service to be stopped during an upgrade procedure. The Muse Proxy Setup will not continue if an upgrade is performed on an account without administrator privileges and installed Muse Proxy Service is detected.

2.5 Uninstall Muse Proxy

To uninstall Muse Proxy, go to the command line and navigate to the directory in which Muse Proxy was installed, the `uninstall` directory. Under Windows, run `startMuseProxyServer Uninstaller.bat` and for Solaris or Linux run `startMuseProxyServerUninstaller.sh` or `startMuseProxyServerUninstaller.csh` to begin the graphical interface mode uninstall procedure. If you want the uninstall to occur in console mode, run the above files with the `-console` parameter. E.g.:

```
startMuseProxyServerUninstaller.bat -console
```

On other systems, run the `uninstall.jar` file, which can be found in the `${MUSE_HOME}/proxy/uninstall` directory. Use the following command line to begin the graphical interface mode procedure:

`java -cp uninstall.jar run` *(or alternatively* `java -jar uninstall.jar`*)*

Use the following command line to begin the console interface mode procedure:

```
java -cp uninstall.jar run -console
```

On a Windows system, you can select `Uninstall Muse Proxy` from the Muse folder on your `Start/Programs` menu.

Windows 7 UAC (User Account Control) strips away any `administrative` power from applications, tasks, features, or actions that a user performs during routine functionality. There are 4 configurable permissions levels in UAC, the default one being `Notify me only when programs try to make changes to my computer`. When using the default UAC configuration, the uninstall of Muse Proxy will give many errors of the form `C:\Program Files\muse\proxy\wizard.log (Access is denied)`, assuming `C:\Program Files` is the location where Muse Proxy is installed. If you are experiencing such problems, set the `UAC` level to `Never Notify` before uninstalling Muse Proxy. After Muse Proxy is uninstalled you can restore the UAC default setting or whatever needed.

During uninstall you may be prompted to remove or keep some files that were changed during the life of the installation. You also have the option to remove them all without being prompted.

2.6 Starting/Stopping Muse Proxy

Muse Proxy can specify in its startup command line a set of parameters, which can also be specified from the associated configuration file.

Usage:

```
startMuseProxy [options]
```

Options:

```
-c [configuration file name] default: C:\muse/proxy/MuseProxy.xml

-p [listening port] default: 9797

-sslp [ssl port] default: 9443

-j [jmx port] default: 9798

-H [proxy host]

-P [proxy port]

-pac [URL to proxy PAC file]
```

 © 2006-2013 MUSEGLOBAL INC

```
-v display version information.

-h display this message
```

For convenience there are scripts available to start it for Windows and Linux platforms. Before you start it you should review the configurations found in `${Muse_HOME}/proxy/doc/MuseProxy.xml`, and make the necessary changes.

Start Muse Proxy manually using `startMuseProxy.bat` from `c:\Program Files\muse\proxy` assuming the default installation directory.

To stop Muse Proxy use `stopMuseProxy.bat` from `c:\Program Files\muse\proxy` assuming the default installation directory.

On UNIX, if the default directory is used for installation (`/opt/muse`), to start Muse Proxy from the command line, use the following command:

```
/opt/muse/proxy/startMuseProxy &
```

To stop Muse Proxy from the command line, use the following command:

```
/opt/muse/proxy/stopMuseProxy
```

Under a graphical environment Muse Proxy might be started from Start menu `(Start/Programs/Muse/Muse Proxy/Start Up Muse Proxy)` and can be stopped from the same Start menu `(Start/Programs/Muse/Muse Proxy/Shut Down Muse Proxy).`

2.7 Muse Proxy Service

Before starting Muse Proxy Service, make sure that the `MUSE_HOME` environment variable is defined. By default, it takes the following values, depending on the operating system:

- on Windows: `%MUSE_HOME%`is by default `C:\Program Files\muse`. It can be set under `Control Panel -> System -> Advanced -> Environment Variables;`
- on UNIX/Linux: `$MUSE_HOME` is by default `/opt/muse`. It can be set in the user profile in its home directory (if Muse Proxy was installed as a normal user) and/or in the global profile (e.g. `/etc/profile`) if Muse Proxy was installed as a `root` user.

If you want Muse Proxy Service to start/stop at boot time you may have already selected `Install Muse Proxy Service` option from the installation or run the `${MUSE_HOME}/proxy/setup/startMuseProxyServiceSetup` script (see the Section 2.3.6,

"Running Muse Proxy Service Setup").

- on Windows NT, 2000, XP, to start/stop the Muse Proxy use the following commands:
  - use `net start "Muse Proxy Server"` to start the service;
  - use `net stop "Muse Proxy Server"` to stop the service;

  Note: You can also use the `Services` from the `Windows Control Panel` to start, stop the Muse Proxy.

- on Linux, if you would like Muse Proxy Service to automatically start when the machine starts, do the following:

```
ln -s /etc/init.d/museproxy /etc/rc.d/rc3.d/S99Museproxy
```

- on Solaris, if you would like Muse Proxy Service to start or to automatically start when the machine starts, do the following:

```
ln -s /etc/init.d/museproxy /etc/rc3.d/S99Museproxy
```

The `rc3.d` directory used above supposes that your current run level is 3 (multi-user mode). If you want different run level, use the appropriate directory. For example, if you want run level 4 (X11 with KDM/GDM/XDM) use the directory `rc4.d`.

The above command creates a link to `museproxy` script that starts Muse Proxy Service. This will cause Muse Proxy Service to run at system start under the `root` user. The `museproxy` script should be created by the installation program under `/etc/init.d` directory. If the installation is done manually one should be available under `${MUSE_HOME}/muse/proxy`. Be sure to update the required variables inside the `museproxy` script (e.g. the `MUSE_HOME` variable).

The `museproxy` script can also be used to start/stop all Muse Proxy Service at any time after the operating system is booted:

- use `/etc/init.d/museproxy` or `/etc/init.d/museproxy start` to start the service;

- use `/etc/init.d/museproxy stop` to stop the service;

- use `/etc/init.d/museproxy restart` to restart the service;

Muse Proxy Service started at boot time on UNIX/Linux systems using the above methods will start and run as the user owning the Muse install directory (`${MUSE_HOME}`). If you want to start Muse Proxy Service as a different user than the one owning the Muse installation directory, edit `/`

 © 2006-2013 MUSEGLOBAL INC

`etc/init.d/museproxy` and comment the line that automatically detects this user. Then specify the desired user on the line below the self explanatory comment:

```
# who is the owner of the Muse Proxy install directory?
MUSE_OWNER=`ls -dl "${MID}" | awk '{ print $3 }'`
```

If you want to start Muse Proxy Service as another user than the owner of the installation directory, comment the line above (the one that instantiates the `MUSE_OWNER` variable), and add a line like the one below and specify the desired user:

```
MID_OWNER=specify_a_user
```

You may also use this script to start the Muse Proxy Service as an user other than the one you used when you logged in. To do that, run the script:

```
/etc/init.d/museproxy
```

If you are not root and Muse Proxy is installed as an user other than root, the script will prompt you to introduce the password for the user to be used when the script will be run.

© 2006-2013 MUSEGLOBAL INC

3.0

Updating the Muse Proxy License Key File

In order to update the Muse Proxy License Key File after the initial Muse Proxy product installation the following steps must be followed:

- start the Muse Proxy;
- open an Internet browser (e.g. Internet Explorer, Mozilla Firefox etc) and log into the Muse Proxy Administrator Console (see the `${MUSE_HOME}/proxy/doc/Muse Proxy Administrator Console.pdf` document for more details);
- click the `About` link in the top right of the page;
- click the `License Information` tab;
- in the `License Update` section click the `Browse` button and select the License Key File to be installed;
- click the `Install` button to install the new License Key File;
- the selected License Key File is automatically loaded and used. No Muse Proxy restart is needed.

© 2006-2013 MUSEGLOBAL INC

4.0

Muse Proxy Running on Multiple IPs Machine

Muse Proxy can be run on a machine that has multiple IPs. It can be configured to run on all IPs (default behaviour) or only on some of them, this being set in the main configuration file (e.g. `${MUSE_HOME}/proxy/MuseProxy.xml`).

In the `BINDADDRESS` field it is specified a list of IP addresses of IP subsets in Classless Inter-Domain Routing (CIDR) notation separated by semicolon `";"` to listen for connections on.

For example: `192.168.14.224/28;192.168.14.109` - This means that all local IP addresses found in the range 192.168.14.224 to 192.168.14.239 and the 192.168.14.109 IP will be added for incoming connections.

Besides the CIDR notation Muse Proxy allows the user to specify patterns of IP address (regular expressions) in this field.

For example:

- `192.168.14.*` - This means that all local IP addresses found in the range 192.168.14.1 to 192.168.14.255 will be added for incoming connections.
- `192.168.14.24?` - This means that all local IP addresses found in the range 192.168.14.240 to 192.168.14.249 will be added for incoming connections.

In order to be able to "listen" for connections, Muse Proxy creates one or more sockets. A socket can be created either for all the IPs installed on the machine running Muse Proxy or either for one specific IP. For each socket, a Server object is created, in order to manage the incoming connections.

The `BINDADDRESS` configuration option is used when creating the Servers. Pattern(s) may be added/ edited/deleted from the admin section, or using the JMX mechanism.

See the `${MUSE_HOME}/proxy/doc/Muse Proxy Advanced Configuration.pdf` document for more information regarding how there are stored the files in cache based on the value of the `BINDADDRESS` field.

© 2006-2013 MUSEGLOBAL INC

5.0

Muse Proxy Advanced Tunings

This section describes the advanced settings that can be done for the Operating System as well as for the Muse Proxy in order to increase the number of requests handled by Muse Proxy under high usage. The tests showed that for a number of requests over 500 per second, the Operating System and the Muse Proxy need tunings in their default settings.

5.1 Operating System Tunings

There are several tunings that can be made in a Linux Operating System with regard to TCP/IP, file descriptors, sockets that will help the Muse Proxy to handle more requests in environments with high usage.

- File System

  Increase the value for the maximum number of file descriptors because the Muse Proxy handles files on disk such as log files, temporary files, cache files. The maximum file descriptors value can be set in the `${MUSE_HOME}/proxy/configure` file, by changing the value of the `PROXY_MaxOpenFiles` variable. A value of 65536 is recommended for a highly used Muse Proxy.

- TCP/IP

  The following TCP/IP parameters can be tweaked on the Linux system for fast connections and to improve the responsiveness:

  - `net.ipv4.tcp_fin_timeout=6`

    Determines the time that must elapse before TCP/IP can release a closed connection and reuse its resources. During this `TIME_WAIT` state, reopening the connection to the client costs less than establishing a new connection. By reducing the value of this entry, TCP/IP can release closed connections faster, making more resources available for new connections.

- `net.core.netdev_max_backlog=65536`

  Sets maximum number of packets, queued on the INPUT side, when the network interface receives packets faster than the kernel can process them.

- `net.ipv4.tcp_max_syn_backlog=3240000`

  Increases the number of outstanding SYN requests allowed.

- `net.core.somaxconn=3240000`

  Limits the maximum number of requests queued to a listen socket. Default is 128.

- `net.ipv4.tcp_max_tw_buckets=1440000`

  Sets the maximal number of timewait sockets held simultaneously by the Operating System. If this number is exceeded, a timewait socket is immediately destroyed. This limit exists only to prevent simple DoS attacks.

- `net.core.rmem_max=16777216`

  Sets the max OS receive buffer size for all types of connections.

- `net.core.wmem_max=16777216`

  Sets the max OS send buffer size for all types of connections.

- `net.core.rmem_default=8388608`

  Sets the default OS receive buffer size for all types of connections.

- `net.core.wmem_default=8388608`

  Sets the default OS send buffer size for all types of connections.

- `net.ipv4.tcp_rmem=4096 87380 8388608`

  TCP Autotuning setting. The first value tells the kernel the minimum receive buffer for each TCP connection, and this buffer is always allocated to a TCP socket, even under high pressure on the system. The second value specified tells the kernel the default receive buffer allocated for each TCP socket. This value overrides the `/proc/sys/net/core/rmem_default` value used by other protocols. The third and last value specified in this variable specifies the maximum receive buffer that can be allocated for a TCP socket.

- `net.ipv4.tcp_wmem=4096 65536 8388608`

  TCP Autotuning setting. This variable takes 3 different values which holds information on how much TCP sendbuffer memory space each TCP socket has to use. Every TCP socket has this much buffer space to use before the buffer is filled up. Each of the three values are used

© 2006-2013 MUSEGLOBAL INC

under different conditions. The first value in this variable tells the minimum TCP send buffer space available for a single TCP socket. The second value in the variable tells us the default buffer space allowed for a single TCP socket to use. The third value tells the kernel the maximum TCP send buffer space.

- `net.ipv4.tcp_mem=8388608 8388608 8388608`

  TCP Autotuning setting. The `tcp_mem` variable defines how the TCP stack should behave when it comes to memory usage. The first value specified in the `tcp_mem` variable tells the kernel the low threshold. Below this point, the TCP stack does not bother at all about putting any pressure on the memory usage by different TCP sockets. The second value tells the kernel at which point to start pressuring memory usage down. The final value tells the kernel how many memory pages it may use maximally. If this value is reached, TCP streams and packets start getting dropped until we reach a lower memory usage again. This value includes all TCP sockets currently in use.

To enable the above tunings copy the assignments into the system file `/etc/sysctl.conf` . They will be preserved at system boot. To enable them after editing the `/etc/sysctl.conf` file execute the following command:

```
sysctl -p
```

5.2 Muse Proxy Tunings

The following configurations in the Muse Proxy can be made to handle larger number of requests in highly used environments. Also other administrative settings are specified.

- More memory for the Muse Proxy process. To specify more memory for the Muse Proxy, edit the `${MUSE_HOME}/proxy/configure(.bat/.csh)` depending on the Operating System and change the values for the `PROXY_XMS` and `PROXY_XMX` variables. The Muse Proxy default allocated memory should be increased especially when a large number of Muse Navigation Manager rewritten requests are handled simultaneously by Muse Proxy. The Muse Navigation Manager component of Muse Proxy needs much more memory resources for a rewritten request as compared with the handling of the un-rewritten request through the regular proxy component of Muse Proxy.

  The `PROXY_PermGen` and `PROXY_CodeCache` variables should not need any change, but if `OutOfMemory` errors are found in the Muse Proxy logs you may increase these values too, but

with small amounts.

Note: The value of the `PROXY_XMS` must not exceed the value of `PROXY_XMX`. `PROXY_XMS` is the minimum amount of memory that will be used by the Java Virtual Machine which runs Muse Proxy. `PROXY_XMX` is the maximum amount of memory that will be used by the Java Virtual Machine which runs Muse Proxy,

Note: The sum of the `PROXY_XMX`, `PROXY_PermGen` and `PROXY_CodeCache` values must not exceed the physical installed memory on the system. It should be smaller because some memory must be left for the Operating System as well to function.

- Change the default logging characteristics by updating the `LOGGER` multi-level entries from the `${MUSE_HOME}/proxy/MuseProxy.xml` file. Depending on how much usage the Muse Proxy has, it is recommended to change the default logging settings so that the logs aren't rotated too quickly and thus information used for debugging and analysis being lost. The changes must be done for all 3 types of Muse Proxy logs: `debug`, `access` and `statistics` and imply the following characteristics:
  - `LOG_SIZE` . Specify here a value large enough that will store data for the desired amount of time.
  - `LOG_MAX_BACKUP_INDEX` . Specify a larger number of log files to retain.
- Increase the number of threads. To do this, edit the `${MUSE_HOME}/proxy/MuseProxy.xml` file and increase the values for the following XML nodes keeping the initial ratio: `START_THREADS, MIN_IDLE_THREADS, MAX_IDLE_THREADS, MAX_THREADS`. The default values for these properties should be increased especially when a large number of Muse Navigation Manager rewritten requests are handled simultaneously by Muse Proxy. The Muse Navigation Manager component of Muse Proxy uses more threads for a rewritten request as compared with the handling of the un-rewritten request through the regular proxy component of Muse Proxy.

  Note: Do not specify really high values from the beginning, try increasing them with small values until reaching a stable configuration.
- Increase the value for the `READ_TIMEOUT` field from the `${MUSE_HOME}/proxy/MuseProxy.xml` file. This specifies the read timeout, in milliseconds, for connections made by clients(browsers) to Muse Proxy.
- Increase the value for the `KEEP_ALIVE_INTERVAL` field from the `${MUSE_HOME}/proxy/MuseProxy.xml` file. This specifies the keep alive requests timeout, in milliseconds, for connections made by clients(browsers) to Muse Proxy.
- increase the value of the `SOCKET_BACKLOG` field from the `${MUSE_HOME}/proxy/MuseProxy.xml` file. This specifies the socket backlog length for incoming connection indications (a request to connect). If a connection indication arrives when the backlog is full, the connection is refused. If the value passed is equal or less than 0 or missing, then the default value, which is 1024, will be assumed.

© 2006-2013 MUSEGLOBAL INC

Note: After making changes in the `${MUSE_HOME}/proxy/MuseProxy.xml` file, the Muse Proxy must be restarted so that the new values to take effect.

© 2006-2013 MUSEGLOBAL INC