MuseKnowledge Proxy Release Notes

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

_____

Trademarks

_____

www.museglobal.com

_____

MUSE

Table of Contents
_____

1.0

Changes in MuseKnowledge Proxy 4.5 Build 03

Release Date: 2017–10–05

1.1 New Features:

* A MuseKnowledge Proxy application or a proxified source can be used as an External Tool within a Learning Management System (LMS) using the Learning Tools Interoperability® (LTI®) launch requests standard.

  LTI acts as a half way SSO, in the sense that starting from the LMS system the access to the MuseKnowledge Proxy Application is seamlessly. Hence, once a user is authenticated to the LMS (s)he can access a Muse Proxy application or a source being defined as an External Tool. The LTI standard does not allow for callbacks URLs so accessing directly the same MuseKnowledge Proxy Application as standalone requires a distinct authentication group defined with a distinct au thentication method. The sample MuseProxyFoundation application comes with a predefined `Au thenticationGroups.LTI.xml` file which just needs to be renamed into `Authentica tionGroups.xml`. MuseKnowledge Proxy acts as a Tool Provider. Some of the LMS supporting LTI are Moodle, Atutor, Sakai, Blackboard. For helping with LTI configuration, a new button named "LTI Guidlines" was added into the MuseKnowledge™ Proxy Administrator Console, Configuration – SSO Authentication page. It contains detailed guidelines to configure LTI for a MuseKnowledge™ Proxy Application.

  When performing an upgrade from a previous version follow the upgrade steps listed by MuseKnowledge Proxy Setup (the steps can be found in `${MUSE_HOME}/proxy/Upgrades.txt` as well) regarding the file `${MUSE_HOME}/proxy/webcontexts/ssoRWP2/WEB-INF/classes/securityContext .xml`.

* Added support to set a dynamic value for the HTTP response `Server` header by adding the following entry `<SERVER_TOKENS`

`enabled="true|false">${productID}/${version}</SERVER_TOKENS>` in `${PROXY_HOME}/MuseProxy.xml` file. When `enabled="false"` attribute is present, the header field will not be set at all.

- Introduced a new filter which rewrites absolute URLs from a JSON response. To configure a source with this filter the entry `<FILTER id="JSON"/>` must be present in the source profile, before the other filters.

- Control of the cookies sent in a request to the vendor can be obtained by using the entry `<COOKIE_JS_PATTERNS>Cookie name pattern</COOKIE_JS_PATTERNS>` in the source profile which will let only the cookies with name matching the pattern to pass from the browser to the native source as opposed to `COOKIE_JS` that if true lets all the browser cookies pass. If both `COOKIE_JS` set on true and `COOKIE_JS_PATTERNS` set with the cookie names pattern are present, then only the cookies with names matching the patterns will pass through.

- Starting with this version, in source profiles you can refer to special variables (such as the user ID that is also logged, the application URL, etc.) by using a new node `PARAMETER/REF`, which means that the parameter value will be given by a reference to a special variable, for example:

```
<PARAMETER process="submit">
  <NAME>userID</NAME>
  <REF>logUserID</REF>
</PARAMETER>
```

Special variables are for now: `logUserID`, `rootPath`, `appURL`, `groupID`, `sourcesGroupID`. The parameter value can be also processed by standard methods such as `process="urlEncode"` or by a special JavaScript function defined in the `DEF` section – for example, the above `process="submit"` attribute must have a corresponding `function submit(input)` in the `DEF` section.

- Experimental support for ClassLoader with dynamic ClassPath for the JavaScript (ECMA script) so that from the script defined in the `DEF` source section we can call Java code which is loaded dynamically by the local ClassLoader without having to add library jar files to the system ClassLoader. This makes MuseKnowledge Proxy sources ready for using specific client Java API libraries for obtaining, for example, authentication tokens or making particular actions via that API against the remote source. The source level `CLASSPATH` element consists of one or more path elements separated by semicolon (`;`). Each path element may refer to a jar file (if it does not end in slash (`/`) and it is not a directory), a directory of classes (if it ends in slash(`/`)) or a directory of jar files (if it does not end in slash(`/`) but it is a directory).

There is now experimental support for scripting mode (https://wiki.openjdk.java.net/display/Nashorn/Nashorn+extensions) via `DEF scripting="true"` and also for an external location for the script code via `DEF src="..."` in order to cope with Java Security Manager rights. This makes MuseKnowledge Proxy sources ready for using external "agnostic" processes to deal with particular APIs. This can be done through Unix shell like back quote strings. Back quoted strings are evaluated by executing the programs mentioned in the string and returning value produced by the 'exec'-ed program. For example, if `curl` is available on the server, the following fragment would extract a JSON from where a token is further obtained:

```
var str = `curl
"http://demo.museglobal.ro/muse/servlet/MusePeer?action=logon&userID=t
heApp&userPwd=thePass&locale=&useProperties=false&templateFile=json%2F
index.json&errorTemplate=json%2Ferror.json"`
var response = JSON.parse($OUT);
```

the back quote can be used for calling other scripts already available in perl/python/bash if this is more handy than involving Java API. `$OUT` is used to store the latest standard output (stdout) of the process spawned.

If Java Security Manager rights are necessary (this can be observed in `ERROR` in `MuseProxy.log`) then the JavaScript code must be placed in a distinct file and `src` attribute must be considered, for example:

```
<DEF src="WhatIsMyIP.js" scripting="true">
```

Then in `${MUSE_HOME}/proxy/java.policy` the JavaScript file has to be referred and necessary permissions or `AllPermission` added.

```
grant codeBase
"file:${MUSE_HOME}/proxy/webcontexts/Applications/MuseProxyFoundation/
profiles/sources/WhatIsMyIP.js" {
  permission java.security.AllPermission;
};
```

All these settings must first be tried on a test server, not on the production one because the procedure may require restart/refresh policy and more trial and error steps and MuseProxy.log analyses (which can become hard in a production environment due to the large number of requests).

If need be HTTP headers from requests and responses can be processed using one or more `HEADER` elements within a new or existent `FILTER`,

```
<FILTER>
  <PATTERNS>www.example.com/special/path</PATTERNS>
  <HEADER in="response|reply|request"
action="remove|rewrite|unrewrite|set|add" name="Header name"
value="static value">
    <SCRIPT>Optionally a dynamic value given by JS script (use CDATA
to avoid XML escaping)</SCRIPT>
  </HEADER>
</FILTER>
```

which will be added in the source profile. This is useful, mainly, for non-standard headers. Note that `"reply"` and `"response"` have the same meaning. The main core HTTP headers (e.g. `Connection`, `Content-Encoding`, `Content-Length`) should not be involved here because there are other layers in MuseKnowledge Proxy that take care of them and processing them with a filter may result in unexpected results. Usual cases will not involve scripting, rather remove/add, e.g.: `<HEADER in="request" action="remove" name="X-My-Header"/>`.

- For filtering content, the `unrewriteAndPrefix` method can be used with a capturing group besides `unrewrite` as an attribute for the `REPLACE` element. This method will unrewrite the follow-up link from its Rewrite by Path / Rewrite by Host form and will create an application prefixed url using the qurl parameter such as `http://proxy.museglobal.ro/MuseProxyFoundation?groupID=1&qurl=http%3A%2F%2Fwww.example.com`.

- A new option `LOG_USER_ID_MODE` was added in the configuration of the login modules, with the following possible values: `append`, `overwrite` and `putIfAbsent` (the default value is `append`). Its purpose is to establish what happens when there are more login modules in the stack that deals with values that are candidates for representing a user name for logging purposes or for providing it to some vendor via first URLs calls.

- Because there are many configurations for proxy rewriting available only in the Muse Proxy application source profile, such as cookie related logic, filters, when integrating with Muse Search, the Type1 Links can now point to a Muse Proxy application root in order to use a Muse Proxy source profile. This also brings the opportunity to use other authentication types for the links coming from MuseSearch federated connectors in the future. In order to specify an application root for links generated from a certain MuseSearch source, from the Muse Admin Console under the source's Navigation Manager Settings, inside the Link URLs entry, besides the pattern, add "`, appRoot:/RootOfTheApplication`", for example "`, appRoot:/MuseProxyFoundation`". The rewriting pattern can be preserved, but if it is missing the one in the source identified from the application will be used (in case no pattern is present don't start with a comma (`,`)). To avoid any confusion of which source ID will be used from the application then add also the "`, sourceID: IDOfSource`", for example, add "`, appRoot:/MuseProxyFoundation, sourceID:WhatIsMyIP`". The proxy application wrapping the necessary sources must be configured to use the module `ProxyLogin ModuleToken` for a flawless authentication.

- A new `PRE_PROCESSING` element in the `web.xml` of an application can now be used to define the files on which a server side processing is applied before serving - at this time the engine is based on FreeMarker. The `RULES` are similar to the ones used for `FILE_SET`. Note that both the extension of the file and the rules must match for the pre-processing to take place. The new element is described in the comments in `MuseProxyFoundation/WEB-INF/web.xml`.

- A new `FILE_SET TYPE` is available in the application descriptor, `web.xml`, namely '`authen ticationFlow`' which states that these files are available to the already authenticated users while for the non-authenticated users the authentication flow is initiated asking for login credentials and if the authentication is successful the resource will be available. For example:

```
<FILE_SET>
  <TYPE>authenticationFlow</TYPE>
  <RULES>
    <INCLUDE>/pdf/.*</INCLUDE>
  </RULES>
</FILE_SET>
```

If a local login page is used then the authentication form (the one with the `id="redirectParametersContainerForm"` will have to use `request.getPath()` instead of `request.getRootPath()`. For example, in MuseProxyFoundation sample application in the file

`${MUSE_HOME}/proxy/webcontexts/Applications/MuseProxyFoundation/www/login/IPandUserPasswordAuthenticationGroup.html` the line below

```
<form name="redirectParametersContainerForm"
id="redirectParametersContainerForm" action="<#if
(request.getRootPath()?has_content)>${request.getRootPath()}<#else>/</
#if>" method="post" style="display:none">
```

was automatically changed during upgrade into:

```
<form name="redirectParametersContainerForm"
id="redirectParametersContainerForm" action="<#if
(request.getPath()?has_content)>${request.getPath()}<#else>/</#if>"
method="post" style="display:none">
```

⚡ The debug `MuseProxy*log*` file now lists some useful warnings when no `Host` header is present in the requests as well as when the proxified host is intended to arrive on the secured connection and it arrived on the plain one.

1.2 Bug Fixes:

⚡ Introduced support to reuse tunneled connections (HTTPS through chaining proxy) by adding the following configuration entry `<TUNNEL_KEEP_ALIVE>true</TUNNEL_KEEP_ALIVE>` in `MuseProxy.xml`.

⚡ Sources configured with a `REFERER` value will use the configured Referer header field only for the first request (the logic of this header is to provide the previous address where the request was made from).

⚡ For content filtering the value of a `VAR` from `REPLACE` entry is substituted with its plain value even when rewrite or unrewite method is not specified at all.

⚡ Even if the RFC standard allows only one `Location` header in the response there are resources which returns two Location values. The browser de-facto standard is to accept them as long they are identical. To adapt to this deviation the navigation filtering now removes the other Locations, if they exist, rewriting only the first one.

⚡ HTTPS tunnels with global proxy details (from `${PROXY_HOME}/MuseProxy.xml`) were not working for Type 1 Links – this is now fixed.

⚡ Internal connections to the servlet engine are now independent of the global chaining proxy details in `MuseProxy.xml`, so SAML/SSO logins and their administration is working correctly in this case, too.

⚡ The way Spring Security and the servlet web session mechanism works in general with respect to session and message man-in-the-middle protection can lead to the end-user obtaining a blank page if somehow (s)he tries to authenticate in parallel in two or more tabs to a SSO related layer. This

can appear if somehow a proxified link is clicked twice because of a broken mouse or if more proxified links are just opened and only later the end user switches to those tab and start the SP-IDP authentication process in both (it is not necessary to introduce IDP logon details as IDP might see the end-user as authenticated). Note that proxified links can be added in iframe in a portal and that can also be the case, and also if a source is expiring after 30 minutess of inactivity the source can do multiple requests in the same time without the end-user having to click more times. Now if parallel login requests using SAML or OAuth are happening the end-user is informed through a message and has the choice to continue in the current tab.

- The path for serving local resources is now correctly decoded according to the URI RFC 3986, so local files having, for example, space in their name can be correctly served and there is no need to copy the file name under a name involving %encoding. Requiring a file such as `http://proxy.museglobal.ro/MuseProxyFoundation/pdf/Muse%20Proxy.pdf` will correctly serve the file named `"Muse Proxy.pdf"` from the file system.

2.0

Changes in MuseKnowledge Proxy 4.4 Build 02

Release Date: 2017–05–31

2.1 New Features:

✏ Muse Proxy can now create access log files in the same configurable format as those created by standard web servers such as Apache HTTP Server, format which can be set via a `%` style pattern (an extension of the Common Logging format). In order to do this, the `LOG_FORMAT` element should have the `type="apache"` attribute set. To have a good base for statistical information, especially in a multi-tenant environment, we recommend using more items besides Common Logging, by adding the inbound server IP address, Muse Proxy application, user session, content type:

```
<LOG_FORMAT type="apache">%h %A %w %W %u %S %t "%r" "%{Content-Type}o"
%s %b</LOG_FORMAT>
```

On a fresh installation this format is already set up, while on an upgrade the old format is left in place to keep compatibility in case there are external log parsers set. If a different output is needed then more information can be found in the document `${MUSE_HOME}/proxy/doc/Muse Proxy.pdf` in section "7.2 The access Log".

✏ Introduced support in `ProxyLoginModuleSQL.xml` for SQL statements. More details on how to use this feature are present in `Muse Proxy Advanced Configuration.pdf`, section 6.4.5.6 ProxyLoginModuleSQL. The backward compatibility for specifying a table is kept.

✏ Added support for IP ranges in `ALLOW` and `DENY` rules from ProxyLoginModuleIP.xml, and this range will be matched against the IP address the connection is coming from. Both IPv4 and IPv6 are supported. All types of rules can be mixed if need be, for example one allow/deny rule can be a wildcard such as `217.156.14.*`, another rule can be a CIDR rule such as `217.156.0.0/16` and another one can be expressed using the range `217.156.11.0–217.156.15.255`.

✏ Introduced redirection to remote Sources depending on the end-user IP (non-proxied links). This is done via `Sources.xml` file via the new `REDIRECT` section containing `IP_RULES` elements which are applied on a set of sources and, if the request is for a source that matches the `APPLY` pattern and the request's end-user IP satisfies the `ALLOW/DENY` sequences, then the response will

be a native redirect to the source URL.

- Source parameters can be provided via `Sources.xml` level in the `<SOURCE>` element via multiple `<PARAMETER name="">` children. Each parameter can be further referred in the Source Profile in `<URL>` or `<POST_PARAMETERS>` via `${name}` syntax and its value will be resolved to the content of this node, exactly as if it were defined inside the source profile.

- The Single Sign-on Authentication (other than SAML) core was upgraded and part of the new features Central Authentication Service (CAS) is also supported. Upgrades instructions related to how to handle the changes in `securityContext.xml` are provided via the setup.

- Keep up with SHA1 deprecation.

- To avoid DNS limitation of 63 bytes per label in case of proxying `https://` hosts via Rewrite by Host technique using `https://` proxy URLs, there is now the possibility to map the very long FQDN to shorter names by using `HOST_MAPPING` elements in the source profile.

- This release contains an improvement for Multi tenant environments using the same host name for all tenants but individual IPs for each one. A chaining request to proxy itself on the different IP is no longer taking place for each request, rather the outbound IP is set directly to the one of the chaining proxy. To achieve this only the `PROXY_HOST` must be configured to the allocated IP in the source profile or at the application level and the `PROXY_PORT` must remain empty.

- The generation of the Client Session cookie values was improved.

- The generation of the Connection ID values was also improved.

- Implement an improvement when `action=source` is called, and data is read from the `Sources.xml` application file.

- Refine the errors `"Unexpected exception while accessing target source."` to contain more details about what was wrong with the access.

- Added a limit for the size of multipart `POST` requests which will be kept in memory and requests which will exceed this limit will be temporary saved on the file system. The value of the limit is configured in `USE_MULTIPART_TMP_AFTER` tag from `${MUSE_HOME}/proxy/MuseProxy.xml`.

- Added `COOKIE_PASS_PATTERNS` options and cookies with name matching these patterns are passed into the browser even if they have a domain. This configuration must be used with care and only where strictly required. Cookie name patterns can be specified such as `SESSION*`, separated by semicolon (`;`).

- Secured the `/admin` context availability.

- Related to proxy application IP Authentication, the `REVERSE` flag was added to control if reverse DNS is performed for the end-user IP. The trend is to set it `false` in the configuration files from now on. Reverse DNS is too costly and can slow down the authentication process.

- In case of HMAC, Referer and IP login modules there's no ID to check against a database and hence nothing to be written in the log file, however in case of various integrations we could be receiving a special parameter in the request for tracking purposes and we want to keep this for logging. Now these login modules also support the `LOG_USER_ID` configuration entry, for example:

```
<LOG_USER_ID>${token}</LOG_USER_ID>
```

※ Because `JSESSIONID` name is too general, the session cookie names for the embedded Jetty contexts (related to Single sign-on) were changed.

※ Added `encodeURIComponent` and `decodeURIComponent` to be used for reference and parameter process for first source requests (including extract and navigate scenarios). The functions are compatible with the JavaScript ones. The existent `encodeURL` and `decodeURL` are based on JDK URLEncoder/URLDecoder which are using `application/x-www-form-urlencoded` MIME format which is not entirely the same as the URI encoding which, for example transforms space into `%20` instead of `+`, for example and some servers are sensitive to these differences.

2.2 Bug Fixes:

※ Microsoft Azure AD OpenID Connect End Point v2.0 and End Point v1.0 can now be used for authentication without complex workarounds – guidelines and suggestions are available in the Ad ministrator Console in Configuration / SSO Authentication.

※ Security constraints for SAML or SSO authentication when starting from plain `http://` proxy links are by default not enforced.

※ `HttpModuleApache` is now correctly sending post data for the extract and navigate scenarios as URL encoded data.

※ Filtering on the exact Client Session ID in the Administrator Console was corrected.

※ Fixed parameter name encoding for HMAC Link Generator page inside Administrator Console.

※ Avoid redirect loops when a Type 2 (rewrite by path) link expires for SSO2 (OAuth / OpenID) authentication.

※ Correctly persisting Tiny URLs, that are used in some cases for MuseKnowledge Search integration.

※ Fixed memory usage when downloading more log files from the Administrator Console.

※ Quicker release of file descriptors when local resources such as images, javascript or css are served by Muse Proxy.

※ While rewriting `object`, `embed` and `param` elements the protocol relative URLs (the ones starting with `//`) are correctly treated.

※ The starting point Type1 links used for MuseKnowledge Search integration can also be generated on `https://` protocol.

※ Existing navigation session were ignoring the `FIND/REPLACE` filters after a `mnm.jar` update – this is now fixed.

※ A very rare deadlock appearing when a Muse Proxy under extremely heavy usage has its `mnm.jar` updated. This fix is actually carried by `mnm.jar` version `1.513` itself, so old versions of Muse Proxy can be updated without a full upgrade to fix this.

- Interpreting the request `Forwarded` headers was failing if no `proto=` was found in any of these headers. This is now fixed.

- The following rare case was fixed: if the same linkout source is in two different applications and the same end-user accesses both applications from the same browser at the same time it is possible that the source we link to is used via a navigation session from the other application yielding misleading statistics.

- Protocol relative URLs in HTTP redirects could have generated wrongly rewritten links – this was fixed.

- In case no `ENCODING` is defined in the source profile, first source request(s) (such as extract and navigate) are now considering UTF-8 as a default encoding for various processing such as parameter encoding processing or deserialization from gzipped content (also when no charset is present in the `Content-Type` reply).

- All multiple HTTP response header fields are now reaching the browser in the response to the first rewritten URL (by host or by path) request which is initiated by the source request (`?action=source`, `?url=`).

3.0

Changes in MuseKnowledge Proxy 4.3 Build 02

Release Date: 2016-11-24

3.1 New Features:

⚡ In the family of Single sign-on authentication, besides SAML2.0, MuseKnowledge™ Proxy now supports a wide range of OAuth, OAuth2, OpenID Connect SSO based. MuseKnowledge™ Proxy supports connectivity with more than a dozen of OAuth providers and also a generic OAuth client implementation can be configured for authentication to the providers that are not diverging from the usual practices in OAuth requests and responses (e.g. return the access token in JSON as `"access_token" : "{value}"`, return profile in JSON and not XML, use `"code"` and `"state"` parameter names, no additional hashes with the access token). The existent OAuth specific support is for: BitBucket, DropBox, Facebook, Foursquare, Github, Google, LinkedIn, Odnoklassniki, ORCiD, Paypal, Strava, Twitter, Vk, Windows Live, Word Press, Yahoo. Note that Google ensures authentication with both the public gmail.com domain as well as Google hosted institutions via Google Apps for Education, for example. There's also a general con figuration for any CAS server using OAuth protocol and a general support for the providers that are following the usual practices as described previously.

For helping with OAuth configuration, a new section was added into the MuseKnowledge™ Proxy Administrator Console, `Configuration` menu – `SSO Authentication`. It contains a checklist and detailed guidelines to configure OAuh for a MuseKnowledge™ Proxy Application.

⚡ External HTTP Authentication Login Module for MuseKnowledge™ Proxy is available. There may be cases in which we need to authenticate against an existent HTTP service or even a html login form from the intranet. MuseKnowledge™ Proxy front end logon is presented but behind the request to a remote HTTP login end-point is made and a success/fail decision based on elements from the page is taken. There is no special need of introducing extra text in comments as long as there are clear elements that confirms a success/failure. Also the sources group ID can be extractable or inferred based on group names, messages, elements.

⚡ Introduced experimental support for load balancers that do not spoof/masquerade the IP of the end-user and pass it in the protocol layer via HAProxy PROXY Protocol v1 or via `X-Forawarded-For`. The IP of the end-user is needed for authentication and logging purposes.

Use the `ALLOW_PROXY_PROTOCOL` options to specify the IPs or address templates separated by semicolon (';') that are allowed to send HAProxy PROXY protocol v1 information. If MuseKnowledge™ Proxy receives HAProxy PROXY protocol v1 it first checks the source address to see if it trusts it. Note that SSL must be terminated at the load balancer side in case of using HAProxy PROXY Protocol v1. A similar option, `ALLOW_X_FORWARDED_FOR`, is available in case of using `X-Forwarded-For`. Although the protocol layer end-user IP is used for all authentication and access logging purposes, the IP of the load balancer can still be observed under the connection entry log `110` from `MuseProxyStatistics.log` files, because that entry is logged when a socket is opened and not when the protocol is understood. However the statistics entry `210` (used when new data is received on a connection) was extended to include the end-user IP from the protocol layer.

- Category grouping for source layer presentation is now possible. Multiple areas can be defined, including A–Z ones and these are displayed in different tabs. Integration with MuseSearch passthrough is available if `dblist` source attributes are defined.

- Introduced experimental support for follow-up links without authentication (session cookie) such as for the preflight OPTIONS where the standard requires the browser to avoid sending authorization data. Only links that are generated by a valid navigation session are allowed and only if the HTTP method and link matches the new `<ANONYMOUS_PATTERNS method="OPTIONS|GET|POST">` source configuration element value.

- Apache HTTP Client library can now be used for the first source request (extract and navigate scenario). Because the Oracle JDK URLConnection does not allow the control of the outbound IP address up to now we were forced to perform an extra request through MuseKnowledge™ Proxy and this increases the complexity of troubleshouting and authentication configuration and adds an extra request. The Apache HTTP Client allows control over the outbound IP address and there's no need of an extra request. To configure a source to use the Apache HTTP Client edit Sources.xml from proxy application level and instead `<CLASS>com.edulib.muse.proxy.application.sources.modules.impl.HttpModule</CLASS>` use `<CLASS>com.edulib.muse.proxy.application.sources.modules.impl.HttpModuleApache</CLASS>`.

- Added a limit which usually triggers using temporary files for saving streams of bytes in certain cases, for example for performing gzip. Otherwise these operations are performed in-memory, and, although more time-efficient, this can limit the number of concurrent requests. This controlled via the new flag `USE_TMP_FILE_STREAM_AFTER` in `${MUSE_HOME}/proxy/webcontexts/NavigationManager/profiles/NavigationSession.xml`.

- Added a new tool in the MuseKnowledge™ Proxy Administrator Console – `HMAC Link Generator` – for generating HMAC links for testing the login via HMAC (keyed-hash message authentication code) signing. The utility allows specifying all possible parameters and combinations for generating a HMAC link.

- Added a new tool in the MuseKnowledge™ Proxy Administrator Console – `Evaluate Regex` – for evaluating regular expressions. The utility is most usefull for administrators to troubleshoot sources filter configurations for find and replace. It has two forms: `By JDK RegEx` and `By Running Filter`. The `By Running Filter` tool generates the XML snippet that can inserted into the MuseKnowledge™ Proxy source profile.

⚹ For load balanced environments the `ID` value (if defined in `MuseProxy.xml` and if `cookieSuffix="true"` – which is true by default, if missing) is also added to the session cookie name as a suffix (e.g. `MuseProxySessionIDp2`), because this behaves more reliable in certain Load Balancer cases, such that the ones combining routing rules (`/admin` rule or `\.p1\.[a-z]\.http` rule to go to a certain proxy) with the sticky cookies mechanism.

⚹ Log files can be named containing elements of the creation date, based on the new pattern attribute of the `LOG` element. For example to capture the activity on a daily basis in files such as `access-20161123.log` and keeping a maximum of 365 such files the corresponding logger will have to be configured as below in `MuseProxy.xml`.

```
<LOGGER name="access" enable="true" flush="15000">
  <DEBUG>NOTICE</DEBUG>
  <LOG_CLASS>com.edulib.ice.util.log.ICETextLog</LOG_CLASS>
  <LOG
pattern="'access-'yyyyMMdd'.log'">${MUSE_HOME}/proxy/logs/access.log</
LOG>
  <LOG_SIZE>0</LOG_SIZE>
  <LOG_FORMAT>{0, date,yyyy-MM-dd HH:mm:ss.SSS z} {1}: {4}:
{3}</LOG_FORMAT>
  <LOG_MAX_BACKUP_INDEX>365</LOG_MAX_BACKUP_INDEX>
  <LOG_TIME_INTERVAL>0</LOG_TIME_INTERVAL>
  <LOG_SCHEDULED_ROTATION type="daily" hour="0" minute="00"/>
</LOGGER>
```

When the number of access logs exceeds 365 the oldest ones will be deleted with the creation of new ones.

⚹ Switching to a newer version of Freemarker template library, namely 2.3.25, as it has more expressions for dealing with the model structures. Normally, there should be no backward com patibility issues with existent templates after the upgrade.

3.2 Bug Fixes:

⚹ The `PUT` HTTP method was relayed as `POST` and some AJAX implementations may not accept this. This was corrected.

⚹ A fix related to file uploads through rewritten POSTs was considered.

⚹ Digest authentication for remote sources was not working correctly if `qop` (quality of protection) value is not wrapped in quotes. Although this behaviour was according to the RFC, MuseKnowledge™ Proxy is now more permissive in this regard.

⚹ IP rules for the application level ProxyLoginModuleIP login module are treated the same way the rules for `${MUSE_HOME}/proxy/hosts.xml` are, that is the first rule that match counts.

4.0

Changes in MuseKnowledge Proxy 4.2 Build 02

Release Date: 2016-06-10

4.1 New Features:

- Extract and Navigate scenarios with processing via built-in encoding functions (`urlEncode`, `urlDecode`, `xmlEscape`, `xmlUnescape`) and via JavaScript server side processing (for the custom cases) on the extracting sequence.

- Added client side SSL authentication via a JKS KeyStore containing a key pair and also using a KeyStore password that must coincide with the key pair one.

- Introduced Multi-tenant SAML 2.0 Authentication as a Service Provider. All products supporting SAML 2.0 in Identity Provider mode (e.g. ADFS, Okta, Shibboleth, OpenAM, Efecte EIM or Ping Federate) should be compatible with Muse Proxy. Tests were performed with the Shibboleth IDP implementation (with Open LDAP and with Active Directory at the other end), Simple SAML PHP IDP, with SSOCircle IDP and with Shibboleth Discovery Service implementation. Few of the features follows:

  - Includes a local Discovery service.

  - Supports external Discovery.

  - Metadata management supporting adding IDP metadata and generating of SP metadata, pre-validation of IDP metadata to detect the need of certificates, tests for authentication, seeing SAML attributes, guidelines and more.

  - Supports specifying the IDP metadata either by uploading the IDP metadata file or by specifying the IDP metadata URL with a local file backup with periodically refreshes.

  - Supports specifying IDP metadata as a file/URL containing one EntityDescriptor or as multiple EntityDescriptor wrapped in EntitiesDescriptor (e.g. a federation) with filters eliminating conflicts if the SP metadata is also present in the same file.

  - Post-SAML authentication decisions via server side JavaScript on letting the user in the application, choosing a source group, choosing an attribute to be logged into the statistics. These, as well as other settings are grouped in the `ProxyLoginModuleSAML.xml` con

figuration file of the SAML login module.

- Extend the cases in which error status come with content and it is better to show the native error than the MuseKnowledge Proxy error page (besides 403 added 404, 500 and 503 Status Codes).

- Being more permissive with `EOF ZLIB` errors while processing input content; otherwise some input gzip content (such as css-s from ACS) yielded "`java.io.EOFException: Unexpected end of ZLIB input stream`" and nothing was read.

- Server Name Indication - handling both cases of servers needing and servers refusing it in the same MuseKnowledge Proxy instance (Java Virtual Machine can only be globally configured for a single scenario) by a workaround for JDK related to Server Naming Indication extension for SSL. We needed to send an empty host in cases the exception is "`handshake alert: un recognized_name`".

- Make way for global and local custom HTTP headers that are sent in web-like responses from the MuseKnowledge Proxy (such as for login and error pages). Note that `CUSTOM_HTTP_HEADERS` must be specified before `WEB_CONTEXTS` in `WebContexts.xml` file. We are using this for example to add `X-Frame-Options` for avoiding Clickjacking.

- New login module for HMAC (keyed-hash message authentication code) signing of time limited links is securing login links with explicit parameters by restricting their usage from a certain portal for a very brief period (e.g. 30 seconds) which cannot be hijacked and executed after a certain time and eventually from a different User Agent (if configured so) than that of the end-user accessing the portal/web-page where the Muse Proxy links are placed.

- Introduced link-out source definition where rewriting patterns are searched externally in the other source profile as well if they don't match the current one. This eases the configuration process for Discovery sources which contain full-text links to other vendors. Shortly, if a source is defined for `LINK_OUT` then when a link is decided to be rewritten the algorithm is: - is the link matching? If yes rewrite - if no, then was it actually excluded from the main source? If it was excluded don't rewrite at all; if not excluded continue to search through the other source patterns.

- Using browser's User Agent for all the requests if the `USER_AGENT` is missing from the source profile.

- If `SSL_TRUST_ALL` is true an always true host name verifier is used.

- Implemented support to serve the response content encoded using gzip for the content originating to MuseKnowledge Proxy, such as the application interfaces. This is done only if a matching Accept-Encoding is in the request and contains the gzip token. This policy of using Content-Encoding is configurable at the global level.

- Cookie Management - in some cases cookies should reach the browser. Added the `COOKIE_PASS` flag. Currently MuseKnowledge Proxy collects the cookies set by targets and manages them and a `Set-Cookie` header is removed in responses to the browser. During time we tried to inject JavaScript and provide the cookies there through `_rwp` variables but in a very complex single page JS application it is hard to achieve this and it is simpler to just let the cookies pass so that the JavaScript expecting them is running naturally.

- For filter regex rules added `rewriteHostHTTP` and `rewriteHostHTTPS` XML attributes to rewrite the host (without searching for the scheme `http://` or `https://`) for the http/https version of the hostname and substitute it.

The ID of the group of sources can be specified for each user when using the local user/password file.

4.2 Bug Fixes:

- Connect Timeout and Read timeout for SSL connections are now set accordingly before handshake to improve the effects of the settings.

- Unknown content which is finally discovered as being binary was ended after the first 8K block and an invalid response was sent – this was fixed.

- Fixed cases when custom filters could not be applied on text/plain reported content.

- Some of the errors were duplicated in the log file;

- `NAVIGATION_SESSION_TIMEOUT` value from `NavigationSession.xml` file was currently not used. Now this is fixed.

- In case of extractors, if the response is gzipped or deflated we inflate the bytes for the search to be possible. The response bytes that will be sent to the browser will remain gzipped as in the original response.

- Fixed a redirect issue when Location comes with an OK HTTP code.

- Removed internal unused NavigationSession properties.

5.0

Changes in Muse Proxy 4.1 Build 01

Release Date: 2015-06-30

5.1 New Features:

- Introducing shortcuts to Muse Proxy Source Entry Point URLs – Muse Proxy offers now new Entry Points for source navigation without the need to specify the `sourceID` and `action=source` parameters, rather just an `url` parameter (or its encoded form), for example `http://proxy.domain.com/MPApplication?url=http://www.edulib.com/products` as opposed to `http://proxy.domain.com/MPApplication?action=source&sourceID=EduLib[&url=...]`.

  Both approaches have their advantages, the original one being more secure as it can hide the initial URL. However for integration purposes the direct `url` parameter is more suitable. Note that based on the `url` value provided the corresponding sourceID is selected. This is done based on a search mechanism against the defined sources described in the documentation, hence Muse Proxy will not automatically proxy any URL provided in the `url=` parameter. Not that it wouldn't be able, but it would be dangerous.

- Availability of a new Muse Proxy Admin function Utilities / Evaluate Shortcut URL to test what `sourceID` is discovered for a certain application when a certain `url` would be provided as parameter. This is needed as the order and definition of sources matters in choosing the source definition that will be further used.

- Sources can be hidden in the Muse Proxy application interface but usable in Entry Point URLs (either with explicit `sourceID` or implicit shortcut URLs).

- A new `REDIRECT` source configuration element is available and its primary purpose is to be used in a hidden "fallback" source to identify the pattern domains of web sites with free content that do not have to be proxied, and that can still end-up in Shortcut Entry Point URLs generated by external systems. Another usage may, in the future, be for sources that integrates without the need of rewriting which, after a first request, are redirecting with an authentication token to their domain and that must not be rewritten at all.

- Extract and Navigate options are available for a source profile in order to deal with sources requiring more authentication or navigation steps (such as selecting a certain database having a dynamic URL containing session identifiers) before handing the control to the browser. Extract variables via the new `EXTRACTOR` option from content and using them in the next URLs or next Posts are the key to this scenario.

- A new boolean source option, `SHOW_GET_PARAMETERS`, is available in case GET parameters of the source URL need to be revealed when the control is handed over via the follow-up URLs. By default they are still hidden for security reasons.

- In order to troubleshoot, or avoid extra configurations, a new boolean source option, `SSL_TRUST_ALL`, can be used for HTTPS sources which involve certificates signed by CAs not covered by the main proxy trust store which is now a copy of the JDK 1.8.0_45. Up to now for these cases a manual configuration of certificates is required and this may not be straight forward. However, note that trusting source certificates that are not signed by trusted CAs is a security decision.

- Application context mapping was extended to support more contexts for the same application and can now use the host and port in the `CONTEXT_ACTIVATION_RULES/URL_RULES/URL_PATTERN` field. This means that an application can be configured to respond to multiple paths and for hosted environments the DNS name of each organization can be used to distinguish between applications in case each organization has a single application and the same path (e.g. `/rewrite`). By default `MuseProxy Foundation` comes configured with `/MuseProxyFoundation` and `/rewrite` patterns.

- Introduced support for persisting the client sessions and their corresponding navigation sessions, authentication tokens and tiny URLs during a graceful Muse Proxy restart. Persistence is controlled through a new boolean flag, `PERSISTENCE`, in `MuseProxy.xml` main configuration file. Session and authorization data is persisted as long as the shutdown happens gracefully, that is via the Muse Proxy stopping scripts, Windows Service stop, or via `^C` or `SIGTERM` but not through a forced process kill or unexpected machine crash.

- Configurable Find and Replace filters acting on the HTTP body can now be crafted in the XML source profiles and will be interpreted at run-time, without the need to write Java code. There are two types of filters: regular expression based and Muse Proxy token rule based similar to the token rules written in Muse Proxy Java filters. There are simple (just find/replace) and complex filter configurations involving conditions (such as `APPLY_IF_FIRST`) and variables.

- Introducing an alternative Navigation Session storage which uses disk space to spool idle sessions. As Navigation Sessions are in the center of the navigation process and each new entry point URL translates into a Navigation Session in order to make room for more navigation actions without adding substantial RAM Muse Proxy can be configured with Apache JCS – Java Caching System system for a composite LRU cache (memory/indexed disk). By default navigation sessions that weren't used for 5 minutes are spooled to disk (but not expired) and if they are requested until they time out then they are retrieved back in memory. For the indexed disk storage only the keys will be stored in memories, but not the content.

   The usage of the JCS system is possible through a new option `NAVIGATION_SESSION_STORAGE` available for setting in `${MUSE_HOME}/proxy/MuseProxy.xml` file. This new option must be set to `JCS` so that the hybrid storage is used. Otherwise, the default value is `memory`.

The JCS setting is recommended for hosted environments with dozens of institutions in case RAM memory proves to be a limit.

- Application Web Contexts are also visible in Monitoring/Client Sessions section from Muse Proxy Administrator Console.

- Extend the Admin Utilities / Encrypt Password to support the symmetric DES encryption.

- Icon configuration for each source is now available in any `MuseProxyFoundation` based application. If configured, the image will be displayed under the Source name, next to the source description.

- The Client Session ID encoding was changed from Hex String representation into base 36 rep resentation in order to lower it and inherently the space used.

- Cookies set from the JavaScript level which aren't intercepted by Muse Proxy JavaScript `_rwp` wrappers were saved in the navigation session. This is not generally necessary as those cookies are usually used at JS level only and besides this could save memory for storing more navigation sessions in parallel. Flags for controlling the un-intercepted JS cookie polices were added in MuseProxy.xml and each source could actually overwrite these in its profile. The new options are `COOKIE_JS` and `COOKIE_JS_PERSIST`.

- Some of the Navigation Session attributes such as Original URL, Entry Point URL and the associated cookies are stored as byte arrays instead of Strings thus reducing the memory space for a Navigation Session.

- Some sources report the JSON content as `text/json` instead of `application/json` and now Muse Proxy recognizes this, too, although the standard value is `application/json`.

5.2 Bug Fixes:

- Corrected the memory values listed for Client Session sizes as the size accounted for shared con figuration data, too.

- The tilde (~) character was encoded when rewriting location redirects when it should not be as it is not a special character. However tilde (~) is encoded because JDK's URLEncoder considers that browsers (although the old ones such as Netscape) do encode them despite the protocol requirement, the explanation being that "It appears that both Netscape and Internet Explorer escape all special characters from this list with the exception of "-", "_", ".", "*". While it is not clear why they are escaping the other characters, perhaps it is safest to assume that there might be contexts in which the others are unsafe if not escaped". Muse Proxy now fixes this as modern browsers are no longer encoding the tilde (~) character.

- The LDAP login module can now accept a complex query combining more attributes – `<SEARCH_STRING>(&(objectClass=person)(sAMAccountName=${NAME}))</SEARCH_STRING>`

- Location URLs that did not follow the standard and contain un-encoded slashes (`/`) in the query

part (i.e. after **?**) were not successfully rewritten. Although non-standard, Muse Proxy can now cope with them without dropping the query part up to slash (**/**).

- The Rewrite by Host mechanism was expecting a port number after colons (**:**) even if an anchor such as `http://host:/path` is equivalent with `http://host/path` and then navigation of such rewritten URLs was failing as the host part was rewritten to contain "**.p**" without a value. This is now corrected.

- The inbound source address was not used for HTTPS remote connections in case no proxy is configured either at source, application or global level. The default IP address of the machine was used. This is now fixed.

- In Muse Proxy Admin, the Monitoring / Client Session / Navigation Sessions the label "Rewritten URL" was renamed into "Entry Point URL" and its value is now correctly computed for the cases of source navigation (up to now it was working only for links coming from MuseSearch).

- Fixed a concurrency bug for FreeMarker interface template file loading which could have resulted in sporadic "Resources not found" errors.

- Fixed a rare shutdown refusal issue when `stopMuseProxy` script is used.

6.0

Changes in Muse Proxy 4.0 Build 02

Release Date: 2014-12-22

6.1 Bug Fixes:

- Content-type not seen as "gzipable" by Muse Proxy during the content processing operation could end up gzipped twice. Such a case was discovered for the `text/json-comment-filtered` Content type. This is now fixed and if `Content-Encoding` is still present in the reply after Navigation Filter did the processing then the `gzip` action is not performed.

7.0

Changes in Muse Proxy 4.0 Build 01

Release Date: 2014-12-19

7.1 New Features:

- Added logic and a new configuration element for skipping content rewriting. There are cases in which resources have to be requested via the proxy (so their URL has to be rewritten), but their content must not be parsed and modified, being served untouched. For the proxy sources this con figuration element is called `TRANSPARENT_CONTENT_PATTERNS`, while for the Muse Search starting point URLs there are two new rules (`includeT:`, and `excludeT:`) to be defined in the `NAVIGATION_MANAGER_MODE` of the Muse Source profile (editable under Link URLs in the MCAA).

- Added support for Search Widgets and Form Integration via Muse Proxy. For this, an extended Muse Proxy source type URL is used at the application level appending an URL parameter (either encoded or direct, non-standard) and an optional parameter stating if the request parameters are further submitted using `GET` or `POST` methods. Something as the next URL can be used to replace the initial HTML `form action` where the value of the `url` parameter was usually the initial value of the `action` attribute:
  `http://proxy.edulib.com:9797/MuseProxyFoundation?groupID=1&action=sour ce&sourceID=SourceID&nativeParams=POST&url=http://provider.domain.com/ path/etc`. Also, a similar link up to, or including `&url=` can be input to many Provider's Widget Builders. Besides having the default option of sending all the `POST` parameters to the native source, the administrator can have a finer control:

  - Prefix the parameters for Muse Proxy with `_lrwp_` and these will be used locally for Muse Proxy. All the rest of the parameters will be sent to the source. It is assumed that one knows which parameters to prefix for the Muse Proxy.

  - Prefix the parameters for the native source with `_rwp_`. If at least one is prefixed then it is assumed that only the prefixed ones will be sent to the native source – hence all the rest of the parameters which are not prefix will be sent to the Muse Proxy.

  - If there are both `_lrwp_` and `_rwp_` prefixes then parameters with `_lrwp_` will be locally used for Muse Proxy, parameters with `_rwp_` will be used for native sources and parameters

which are not prefixed will be used locally for Muse Proxy.

- Rewrite by Host – Introducing the Rewrite by Host (Proxy by Host) functionality which solves more easily situations where the initial Rewrite by Path mechanism, which stored the proxy markers in the path, altering the path, was colliding with the source scripts assumptions on the URL's path. Although the Rewrite by Path mechanism tried to re-decode the path when needed by the native JavaScript there are cases when it is very hard to achieve this even through a special filter.

  Leaving the path untouched and altering the host section of the URL brings in some cases more advantages, while in other cases there are also disadvantages. That is why this option will be configured on a source by source basis, because most of the sources work on the initial Rewrite by Path and so have the advantages of no wildcard DNS changes and no wildcard SSL certificates.

  The format of the rewritten links that are navigated in the browser, after the starting point are part of the Muse Proxy technical mechanism and not an API being subject to future changes and should not be used for interconnection purposes or entry points. Currently, besides the native host, the navigation session ID, the Muse Proxy ID and the native protocol are stored as well in the host sub-domain. To cope with DNS fully qualified domain name and token restrictions it is advisable that in case of load balancing the `ID` configured in `${MUSE_HOME}/proxy/MuseProxy.xml` must be as short as possible (even one letter), and must not contain dots ('`.`') nor dashes ('`-`').

  Because there is no other technical solution for accessing sub-domains, the DNS server from the network where Muse Proxy is installed must be configured, besides the normal host name entry, with an extra wildcard DNS entry so that all the sub-domains of the proxy FQDN point to the same IP, the IP of Muse Proxy. For a hosting scenario more such entry pairs are necessary.

  For a better performance of the Rewrite by Host and also associated with a hosted solution the main configuration file `${MUSE_HOME}/proxy/MuseProxy.xml` requires to list the proxy fully qualified domain names under the SERVER_NAMES element.

- Added support for Load Balancing HTTPS traffic via SSL termination, so that the load balancer takes care of the SSL traffic, while the connection between the load balancer and the Muse Proxies is done in plain text, assuming a secure network, thus avoiding unnecessary encryption times. This is achieved either by the de-facto `X-Forwarded-Proto` header field or by the RFC 7239's `Forwarded` header field containing "`proto=https`" which make Muse Proxy behave as if the inbound connection was on SSL.

- Added configuration elements to control the logic of deciding the resulting URL protocol, either HTTP or HTTPS. Up to now Muse Proxy could only isolate the protocols of the source from the protocol of the access. However, accessing Muse Proxy via HTTPS when a source is on HTTP is tricky as all the resources must be accessed on HTTPS, or otherwise the browser security will forbid getting the non-secure resources. This make source configuration or the build of some source filters quite hard. The control of the resulting protocol is done through two new options `IF_HTTP` and `IF_HTTPS` that take the value "`proxy`" or "`source`". If missing, the "`proxy`"

behaviour is assumed. The options are available both at the global level and at the source level.

- ✎ If Muse Proxy rewrites an "`http://`" URL, the `IF_HTTP` option gives the resultant rewriting protocol based on the URL protocol and on the entry point of the current navigation session. If the value is "source" then the protocol will always be "`http`", irrespective of the proxy's one. If the value of this option is "`proxy`" then the protocol is either "`http`" or "`https`" depending on the entry point of the navigation session.

- ✎ If Muse Proxy rewrites an "`https://`" URL, the `IF_HTTPS` option gives the resultant rewriting protocol based on the URL protocol and on the entry point of the current navigation session. If the value is "`source`" then the protocol will always be "`https`", ir respective of the proxy's one. If the value of this option is "`proxy`" then the protocol is either "`http`" or "`https`" depending on the entry point of the navigation session.

✎ For hosting more organizations on the same Muse Proxy, without SSL Termination on a load balancer, the case in which each organization needs its particular certificate can now be achieved. Individual KeyPairs (Private Key and Certificate) can be assigned independently for each IP. Because the SSL handshake takes part at the TCP/IP level distinct certificates require distinct IPs for the association to take place; distinct ports wouldn't be enough as certificates has to be assigned as well with host names.

This is possible via the extension of the configuration element `SSL_KEYSTORE_FILE` in the sense that it now allows a new attribute `ip="<IPAddress>"` and the entries with the `ip` attribute can be multiple. If a SSL connection reaches an IP and that IP does not have an associated SSL Java KeyStore then the default KeyStore file (the one with no `ip` attribute) is used.

✎ SSL Protocols (algorithms) that are used both on the server end and on the client end (requests against the sources) are configurable. The default configuration is for example not including `SSLv3` to avoid the recent vulnerabilities. As Muse Proxy runs inside an Oracle Java Virtual Machine the permitted values for the SSL context and enabled protocols must be in accordance with it. Currently the set "`TLSv1; TLSv1.1; TLSv1.2`" is configured and depending on the JVM version the available ones are used. For example in JDK 1.6 only `TLSv1` is supported. Although not recommended there are some sources that explicitly require the configuration of `SSLv3`. Care must be taken when configuring these. That is why the SSL Protocols on the client end can be configured both at the system level and at the source level to ensure that the such sources are isolated.

✎ Implemented support to serve the response content encoded using gzip for the Navigation Manager rewritten pages. This is done only if a matching `Accept-Encoding` is in the request and contains the `gzip` token. This policy of using `Content-Encoding` is configurable at a global level.

✎ Enhanced the Connection Refused error messages – Explicitly added in the log file the URLs generating the errors that are obtained when navigating starting from Type1 request (MuseSearch generated full text entry point URLs). Also for the proxy source links some of the headers for the first requests are displayed (this contains the URL and the next proxy hop). Although Java stack traces may print the host name in some cases, thus duplicating some information, always having the explicit URL should make troubleshooting easier.

✎ Instead of redirecting to the native site, the expired navigation URLs that are initiated starting from

a proxy source URL can now require re-logon in the initial application and authentication group and, after successfully logging in, the source navigation is restored as long as the native source URL is able to function out of the initial session and request context. This is a global configuration option, and also requires that each Muse Proxy Application and each Muse Proxy Source have stable and unique `code`s defined in the Muse Proxy application files `${MUSE_HOME}/proxy/webcontexts/Applications/Applications.xml` and `${WEB_CONTEXT_HOME}/profiles/Sources.xml`. This is required as Muse Proxy is a multi-application server.

- The Muse Proxy Client session Cookie has been modified to `MuseSessionID` and is now set using a domain not a host. This works consistently both for Rewrite by Host and Rewrite by Path.

- Some providers require sending the end-user IP via `X-Forwarded-For` de facto standard field. This is now possible on a source by source configuration, by default the end-user (client) IP not being sent.

- Reload and use main configuration elements from the `${MUSE_HOME}/proxy/MuseProxy.xml` file, via a new admin operation `Refresh Configuration` located under the menu `Advanced/Operations/`. It is not possible to reload all the elements because some of the objects are only created upon Server start-up, but many of them are reloadable without interfering with the live session.

7.2 Bug Fixes:

- Proxying HTTPS URLs as a classic proxy ignored the port and always used the default one, i.e. `443`. This was fixed as being important for the Muse Search scenarios.

- Improved the analyzer for finding the end of `script` tag to be in full accordance with the HTML Specs detection according to the consortia state machine (https://html.spec.whatwg.org/multipage/syntax.html#script-data-state). The end of script detection is a tricky operation, but it doesn't have to do with JavaScript quotes and comments; normally the end of script is the first `</script*>` (* = any char) but inside a HTML `<!--` comment one can start another `<script>` and `</script>` will no longer be the end of the main script. However, if there's only a `</script>` inside the comment, or before a `</script>` end the first occurrence is a `</script>` not a `<script>`, then it represents the end of the main `<script>` outside of the comments.

- Fixed a potential loop when the end-user is IP authenticated to the Navigation Manager and a partial (incompletely rewritten) MNM request is made.

- The URL part of the `content` attribute of the element `<meta http-equiv="refresh"` which is generated directly in the DOM via JavaScript's `document.write` could have resulted in a possible wrong URL containing an extra `'/'` in the end and in a JavaScript Syntax error due to a non-escaped `'` in the cases where `'` and `"` are not used interchangeably in JavaScript's `document.write`.

- Rewrite URLs from the `style` attribute of `<dl...` elements and from the `href` of the `<link rel="alternate stylesheet..."` elements.

- Muse Proxy HTML parser was sensitive to some locales and this is now fixed.

- Corrected a regression bug related to the shutdown class that manifested on systems that have only private IP(s) assigned (e.g. `192.168.C.D`) and the shut down process refused to send the shutdown command.

8.0

Changes in Muse Proxy 3.1 Build 02

Release Date: 2014-04-17

8.1 New Features:

* Implemented a new Muse Proxy Application login module for performing authentication based on a referrer URL. This performs authentication against the client's referrer URL. It is advisable to combine this with the IP authentication module or a custom authentication form for user/ password.

* The REFERER field configured for a source is no longer validated as an URL because there are cases where it is required to have a custom referrer in order for the target source provider to easily track the requests coming from a subscriber.

* Muse Proxy and Muse Proxy Setup are compatible with the recently released Oracle JDK 1.8. Both the native launchers and the generic jar launcher are able to function if the JVM on the target system is JDK 1.8.

8.2 Bug Fixes:

* The source type requests (e.g. http://serverIP:9797/App?groupID=1&action=source&sourceID=XYZ) coming inbound on a certain IP (or its equivalent FQDN) are now going outbound to the target using that certain IP as source IP, without the need for explicit proxy chaining.

* There were some cases for rewritten URLs using HTTPS via proxy chaining where the target system didn't understand absolute URLs – some web servers couldn't identify the resource because they concatenated the resource with the host and it resulted in an invalid URL. This is now fixed by using relative URLs for proxy sources using HTTPS connections via proxy chaining.

9.0

Changes in Muse Proxy 3.1 Build 01

Release Date: 2013-09-27

9.1 New Features:

⚜ The Muse Proxy Administrator Console was updated to no longer load the entire page for every action. Instead, it is updated only a DIV section from the page content with the result obtained after running that action. The actions are run now using AJAX code and the current page loaded is updated dynamically without being needed a full page reload. Many changes were done in the www interface because all the links from the page needed to be transformed into AJAX calls. Added support so that when deleting the last item from the current page, to be displayed the previous page. This was done for the "Client Connections", "Client Sessions", "Tiny URLs", "Server Statistics" sections.

⚜ Created the MuseSourceID filter. The MuseSourceID filter extracts from a rewritten URL of 'Type 1' the 'MuseSourceID' CGI parameter value and stores it in the Navigation Session. For the target site accessed using that Navigation Session there will be activated only the rewriting filters matching that Source ID. Updated the documentation to refer the new Muse Proxy Constant introduced and to describe the functionality for the MuseSourceID filter.

⚜ A Java API for creating rewriting filters was released. It can be used to create rewriting filters java modules. Added the ${MUSE_HOME}/proxy/tools/filters directory where customers can create, build and deploy their own rewriting filters along with the build and deploy of the museproxyfilters.jar file.

9.2 Bug Fixes:

⚜ The "${MUSE_HOME}\proxy\webcontexts\NavigationManager\profiles\Filters.xml" file did not contain the entry corresponding to the com.edulib.muse.proxy.filter.MuseSourceID filter. This was fixed.

⚜ The content handled by the "getResource" action is now returned using UTF-8 encoding. This

encoding is specified in the Content-Type HTTP header of the reply. The content handled by the "sources" action is returned using UTF-8 encoding. This encoding is specified in the Content-Type HTTP header of the reply. When an error is encountered the response is returned using UTF-8 encoding. This encoding is specified in the Content-Type HTTP header of the reply. The FreeMarker templates are read from disk using UTF-8 encoding. When requesting a resource using "getResource" action there will be created a persistent Client Session only if the requested resource is defined as authenticated in the FILE_SETS section from ${WEB_CONTEXT_HOME}/profile.xml file.

- Updated the methods handling the cookies which are stored by Muse Navigation Manager for target sites and which must be expired. These methods now return "Expires" cookie property having as value an expired date instead of "Max-Age=0" cookie property. The update was necessary because "Max-Age=0" is not supported by Internet Explorer browser.

- Updated Muse Proxy to use as owner for the target connections the Handler instance cor responding to the client connection currently handled. Added the TARGET_KEEP_ALIVE field in MuseProxy.xml specifying whether Muse Proxy should use keep alive for target sites connections. Updated the Muse Proxy documentation to refer the new field. Updated the Muse Proxy Install.pdf document to specify the tunings that must be done for the Windows and Linux operating systems in order for Muse Proxy to support many simultaneous connections.

10.0

Changes in Muse Proxy 3.0 Build 04

Release Date: 2013-08-14

10.1 Bug Fixes:

- Implemented a major improvement in Muse Proxy when handling the HTTP "POST" or HTTP "PUT" requests. Previously such requests were read with a delay of 1 second. This happened no matter whether the requests were addressed to the Proxy Component or to the Web Component of Muse Proxy. This was fixed.

11.0

Changes in Muse Proxy 3.0 Build 03

Release Date: 2013-08-08

11.1 New Features:

- Implemented a PHP script that can be used for integrating in a portal the dynamically rewritten links returned by Muse Proxy for the target Muse Proxy Sources accessed. The "6.5 Portal Integration" section from the "${MUSE_HOME}/proxy/doc/Muse Proxy Advanced Con figuration.pdf" document contains detailed information regarding Muse Proxy Portal Integration and how this PHP script should be used in a portal.

- Created the Anonymous application with the following features: – index page selects au thentication method (IP or U/P); – each authentication group has its own sources group; – no javascript or jquery; – pages are simple and with comments to easily identify each zone; – no GET parameters, only POST; – "light" theme.

- Added a new parameter "DELETE_CLIENT_SESSION_ON_LOGOUT", with possible values true/false in the ${WEB_CONTEXT_HOME}/WEB-INF/web.xml file of the Administrator Web Context and in the existing Muse Proxy Applications. This parameter tells the system whether the Client Session must be deleted after a successful 'logout' action. If this field is missing, the default value used will be 'false' meaning that the Client Session will not be deleted after a successful 'logout' action.

- Added the 'PROXY_USED' field in the Source's profile with the possible values: 'NO_PROXY', 'SOURCE_LEVEL', 'APPLICATION_LEVEL', 'GLOBAL_LEVEL'. Depending on the value of this field there will be used the proxy access details from the corresponding level. Added the 'PROXY_HOST', 'PROXY_PORT', 'PROXY_PAC', 'PROXY_AUTHORIZATION_USER_NAME', 'PROXY_AUTHORIZATION_USER_PASSWORD' and 'PROXY_AUTHORIZATION_SCHEME' parameters in the '${WEB_CONTEXT_HOME}/WEB-INF/web.xml' file for Muse Proxy Applications. These proxy access details will be used by a Muse Proxy Source when the 'PROXY_USED' field from the Source's profiles has the 'APPLICATION_LEVEL' value. Previously, when a set of proxy access details were set at global level and a Muse Proxy source did not used a proxy, all the HTTP requests done by the source did not used a proxy, but, when the rewritten 'Type 2' link was

returned, it was chaining with the globally defined proxy. This was fixed and now if a source does not use a proxy then the rewritten link returned will not chain with a proxy either. Previously if a proxy pac returned a set of proxies and the first one of them failed, the Muse Proxy Source used the second one, but the navigation on the rewritten link was tried to be done using the first proxy returned by the proxy pac and the navigation failed. This was fixed and now all the proxies returned by the proxy pac which failed for the source will be ignored also when the navigation will be done on the rewritten link.

- Previously, the JavaScript content included in the rewritten pages was computed internally in the Muse Navigation Manager code. Now the statical and dynamical parts of this JavaScript content are stored in 2 separate files and these files are included in mnm.jar at the build process. The dynamical part is updated with the run-time information before being appended to the JavaScript content.

- Increased the Client Session Timeout value to 35 minutes. This value must be strictly greater than the Authentication Timeout for all of the existing Web Contexts. Also this value must be strictly greater than the Navigation Session Timeout.

- Added a new chapter named Muse Proxy Features in Muse Proxy.pdf that lists the features supported by Muse Proxy.

11.2 Bug Fixes:

- The file defined by the INDEX_PAGE_RELATIVE_PATH parameter from Application's web.xml file is now served on the root request (e.g. http://proxyHost:proxyPort/AppID/) without checking the level of access. Now, if a Muse Proxy Application has a non-void value for this parameter, the workflow is as follows: – If the user accesses http://proxyHost:proxyPort/AppID/ URL it is read the file defined by this parameter, the freemarker from it is run and the output content is returned; – If the user accesses the http://proxyHost:proxyPort/AppID/index.html URL and the file defined using this parameter is not public a 'Not Found' response is returned.

- Fixed Muse Proxy Application behavior regarding loading login in the sources section div. This happened if after login, Muse Proxy was restarted and then, the first request (an Ajax request) was not authenticated and the login page was returned. This page that was loaded in the div where the authenticated information must have been loaded. This behavior was fixed and now a redirect to the logon page is returned.

- The rewriting of HTTPS sites using proxy chaining did not work. This was fixed for the Proxy IP authorization and for the Proxy Basic authorization with user/password. The rewriting of HTTPS sites when chaining with a proxy using Digest User/Password authorization is still not supported.

11.3 Known Bugs:

- Muse Proxy Sources cannot access successfully target HTTPS sites when chaining with a proxy using Digest user/password authentication.

- Muse Navigation Manager cannot rewrite successfully target HTTPS sites when chaining with a proxy using Digest user/password authentication.

12.0

Changes in Muse Proxy 3.0 Build 01

Release Date: 2013–06–21

12.1 New Features:

- Updated the descriptive text for MuseProxyFoundation application displayed in the Welcome Page, to provide the default User/Password access details for this application.

- Updated Muse Proxy web interfaces as follows: Added the latest jQuery & jQueryUI Javascript libraries; The Muse Proxy Applications actions (navigation, filter, sort) are now implemented using AJAX; In the Muse Proxy Applications and the Root Web Context, the HTML attributes written in page using Freemarker code are now HTML escaped; The header and footer in all the logon and error pages were uniformized; The "About" floating panel from the Muse Proxy Application web interface is now displayed using tabs, the "Product Information" tab was added; All the files from the Root Web Context can now be accessed also directly using the getResource action; In Muse Proxy Applications, changed the "Muse" text with a Proxy constant obtained by calling a Freemarker function; Added a version for the Muse Proxy Applications, the Muse Proxy Application version is maintained in a Freemarker variable, in the "application.inc" file.

- Added support for allowing Clients to access Muse Proxy using HTTPS protocol.

- Implemented a friendly edit panel for the "Configuration" >> "Administrative Passwords" section from Muse Proxy Administrator Console.

- Added "json" MIME type. Added the possibility that, when running Muse Proxy Application Actions, to take in consideration the extension of the returned template file in order to set the mime type accordingly.

- Updated the Muse Proxy Applications to insert the MUSE_PROTOCOL_KEYWORD marker in all the rewritten URLs that are formed by Muse Proxy Sources in order to correctly handle HTTPS rewritten sites.

- Created the "Muse Proxy Sources Profiling.pdf" document describing the entire sources profiling process.

- Revised and improved the functionality in the "Configuration>> Server IP(s)" section from Muse Proxy Administrator Console.

- Created a root web context for Muse Proxy that handles the requests addressed to the root web page of Muse Proxy (e.g. http://proxyHost:proxyPort/). The root Web Context is used to implement the Muse Proxy Welcome page. Depending if the user which accesses this Web Context is IP authenticated or not, there will be displayed a different amount of information in the Welcome page. For the non IP authenticated users there is displayed a page containing some general information regarding Muse Proxy. For the IP authenticated users there is displayed a page from which the user can access the Muse Proxy Administrator Console logon page, the Muse Proxy Foundation Application web page, the list of Muse Proxy features, the Vendor Contact information and it can access the Muse Proxy documentation.

- Changed Muse Proxy Server term with Muse Proxy term in the web interface, in code and in the Muse Proxy documentation.

- Implemented a login module for FTP authentication in a Muse Proxy Application.

- Implemented a login module for IMAP authentication in a Muse Proxy Application.

- A Client Session is now counted and made persistent in Muse Proxy memory only for requests that serve authenticated content. For the requests to public resources there are created only internal temporary Client Sessions which are not counted, which are released from memory immediately after the request is handled and which do not return a Client Session cookie to the Client (browser) which performed the request.

- Now the unexpected errors encountered by Muse Proxy Applications and by the Root context are displayed using the ProxyError.html freemarker template, instead of the ProxyError.xsl stylesheet. This assures that a skin defined in a freemarker file can be imported and used in the error page returned.

- Changed the look of the Muse Proxy Administrator Console as follows: changed the skin, changed all the Web Context Administrator pages in order to support the new skin, changed the session timeout window to support the new skin, changed the Shut down page.

- Added the NAME and DESCRIPTION fields in the Sources profiles from Muse Proxy Ap plications.

- Added the AUTHENTICATION_TYPE field in the Source profiles from Muse Proxy Ap plications.

- The Services Web Context has 2 Web Modules. For the cases when these Web Modules au thenticate successfully they are now returning in the Content-Type HTTP header the charset=UTF-8.

- Updated the Muse Proxy configuration files to not contain comments outside of the root node.

- Updated all the Muse Proxy XML configuration files to use the UTF-8 encoding.

- Implemented support to expire a Muse Proxy Application at a certain date.

- Added User-Agent HTTP Header support in sources profile. Updated all the profiles in order to contain this new field. Updated the documentation to document this new field. Updated the JMX component to correctly display this field.

- Added Custom HTTP Header support in sources profiles. Updated all the profiles in order to contain this new field. Updated the documentation to document this new field. Updated the JMX component to correctly display this field. Note that the mechanism implemented will skip the

following headers if added by the user: "User-Agent", "Cookie", "Referer", "Authorization", "Proxy-Authorization", "Content-Length", "Host" and "Connection". These are skipped either because there are different fields that are treating them (this is the case for "User-Agent", "Cookie", "Referer") or the fields must be computed inside the proxy modules and so the user cannot set them.

- Added the "encryption" attribute to the GLOBAL_IB_PASSWORD field in MuseProxy.xml file, containing the encryption of the Global InfoBase password. This attribute may have the 'NONE' and 'SHA1' values. The JMX console was updated, in order to provide "get" and "set" operations for this new attribute. The Muse Proxy Server Administration Console was updated, in order to edit this new attribute. The changes were done in the "Muse Navigation Manager/Update" section. Also the "Save" action for the Global InfoBase Access Details was updated in order to automatically encrypt the new password in accordance with the selected encryption.

- Updated the Muse Proxy login modules in order to write in log the cause of the authentication failure. Updated the Authentication Manager to compute the authentication module that made the authentication process to fail and specify it in the message written in log.

- Updated the Muse Proxy Administrator Console so that all the filtering forms to contain a "Reset Filters" button by means of which the user can reset all the filter fields.

12.2 Bug Fixes:

- The "About >> License Details" section from Muse Proxy Administrator Console, now displays well the License Key File properties containing non-ASCII characters or which contain the quote (\"), \r or \n characters.

- The logout action for Muse Proxy Applications is allowed to be run also when the user is not au thenticated. The javascript timer which counts the period of time until the session expires is now reset when an AJAX call is run.

- Implemented a major improvement in the Muse Proxy rewriting mechanism, especially when rewriting web pages which load many JS files and which perform many AJAX calls. Technically the improvement was related to the usage of the queue concurrency mechanism from the java.concurrent.util package. There are situations with complex web pages for which the rewriting time was reduced from 20s to 3s. With other words the web pages rewriting got faster, the end user waits less when clicking on links which are rewritten through the Muse Navigation Manager.

- Updated the Muse Proxy code in order to update the last access time of the Authentication Token each time a successful authentication using that Authentication Token was performed. Also, each time a Navigation Link is processed, if the Navigation Session associated with that link has an Au thentication Token then that Authentication Token's validity period will be extended.

- Now when a Tiny URL is generated using the "Utilities >> Rewrite URL" section from Muse Proxy Administrator Console the URL is computed entirely directly without being made an internal request to the /TinyURLGenerator service.

- Updated the Muse Proxy code so that all the start-up parameters documented in the "Muse Proxy Install.pdf" document to work well. Removed some of the start-up parameters which were too

particular and they were not needed to be supported by the start-up script, but they can be set in the Muse Proxy configuration files.

- Updated some Muse Proxy stylesheet files in order to bring some improvements to the Muse Proxy Admin interface such as: align the confirmation message in the "Statistics / Server" section; in the "Utilities / RewriteURL" section, the Cookies table, in some cases there was a minor problem with the table color, this was fixed; in the "Maintenance / Servers" section fixed the border dimension of the table.

- When the request to /ProxyInformation is successful, now there is also verified the structure of the content received. This is done to exclude the case when the content received can be a logon page.

- Updated Muse Proxy code to not chain anymore exception messages, when creating the full error message, unless it is important to have them chained. In this way, the error message written in the Muse Proxy log file is better understood by the Muse Proxy Administrator.

- Updated the Muse Proxy Advanced Configuration manual to document the complex FreeMarker objects used at the interface level in Muse Proxy Applications.

12.3 Known Bugs:

- If a Muse Proxy Source accesses a HTTPS site and is configured to use proxy access details (e.g. Proxy Host/Proxy Port or Proxy PAC), the rewritten links returned by that Muse Proxy Source cannot be navigated successfully through Muse Navigation Manager.

13.0

Changes in Muse Proxy 2.6 Build 20

Release Date: 2013-03-11

13.1 New Features:

* Updated Muse Proxy Administrator Console web interface: updated "Login" and "Retry" buttons from the login pages to have the same look and feel as the whole console, added pop-up messages using jQuery code, added default values for the "Rewriting Patterns" and "Muse Proxy Au thentication Token" fields in "Utilities >> Rewrite URL" section, added confirmation pop-up for the "Run Garbage Collector" action in the "Advanced >> Virtual Machine" section.

* Implemented support to reload the Muse Proxy Applications at run-time.

* Implemented support for Muse Proxy Applications.

* Added support in Muse Proxy Administrator Console for editing the ${MUSE_HOME}/proxy/passwords.xml file. Created the "Utilities >> Encrypt Password" section that can be used to generate an encrypted password using the "MD5" or "SHA1" algorithms.

* Extended the "Monitoring >> Client Sessions" section of Muse Proxy Admin. Now for each Client Session that handled access to one of the Web Contexts: "administrator", "naviga tionManager" or "services" there is displayed the authentication meta-information in the Client Session details page. New filters were added for filtering the Client Sessions using the au thentication meta-information fields.

* Extended the "Rewrite URL" section from Muse Proxy Admin to allow the generation of Tiny URLs on demand.

* The "MuseKey" HTTP Header will no longer be used when generating a Tiny URL for all Muse Proxy Server versions 2.6.1.1 or above version. For the previous versions of Muse Proxy Server the "MuseKey" HTTP request header will still be used. The changes are available in modulesutil.jar version 1.2209.

* Removed the Start prefix from the Muse Navigation Manager markers which contain it.

* The "Rewritten Request Initial" term was replaced with the "Rewritten Request Type 1" term,

meaning that all the references of the "Rewritten Request initial" (property names, class name etc.) were replaced with the "Rewritten Request Type 1" term. The "Rewritten Request Other" term was replaced with the "Rewritten Request Type 2" term, meaning that all the references of the "Rewritten Request Other" (property names, class name etc.) were replaced with the "Rewritten Request Type 2" term. The "Rewritten Request Type 2" Web Module now supports its own au thentication process. This means that, unlike the older version, now the Muse Proxy administrator can setup different authentication for the "Rewritten Request Type 1" and for the "Rewritten Request Type 2" Web Modules. In the Web Module configuration file (i.e. ${WEB_CONTEXT_HOME}/WEB–INF/web.xml) it was added the "WEB_MODULE_AUTHENTICATION_RELATIONS" element. By means of this element, Muse Proxy Server administrators can establish different authentication relations between Web Modules (See Muse Proxy documentation for more details). In the Navigation Manager Web Context configuration file (i.e. ${MUSE_HOME}/proxy/webcontexts/NavigationManager/WEB–INF/web.xml), for the re writtenRequestType2 Web Module, it was added a new configuration field, named NAVIGATION_SESSION_CONFIGURATION_FILE, having as default value "${WEB_CONTEXT_HOME}/profiles/NavigationSession.xml", which specifies the path to the configuration file where there are stored the settings related to Navigation Sessions. In this file there was moved the NAVIGATION_SESSION_TIMEOUT parameter of the rewritten RequestType2 Web Module and there were added other parameters which describe how Muse Proxy Server should behave in situations when a rewritten link either does not contain a Navigation Session ID, either it contains a Navigation Session ID which is expired or which does not match the rewritten URL. The Muse Proxy Server access details (the ones which are used for authenticating the user to the Muse Proxy) are now extracted and removed from the requests mapped to 'administrator', 'navigationManager' and 'services' Web Contexts. This process is performed for any request no matter if that request is authenticated directly or if it is authenticated based on the data in the authentication cache. The JMX console MBean structure has been updated to reflect the latest changes. The Muse Proxy Statistics log message structure was also updated.

- Updated the filter classes to not add a prefix (&) and a suffix (=) to marker's name. Updated the MuseProxyUtils and MuseProxyServerUtils classes to correctly process both the markers containing a prefix and a suffix and the markers without a prefix and a suffix.

- Updated the Proxy Admin "Maintenance >> Update" section by adding a "more" link. Created a JQuery help pop-up which is opened when the user navigates on this "more" link.

- Updated the encodeMNMurl methods from the MuseProxyUtils class to not add anymore the "MuseFirst" marker in the rewritten links. Removed the "MuseFirst" marker from several example links from the comments present in several classes.

- Changed the value stored in the authorization markers to no longer contain the "Basic" prefix.

- Done some small changes in Muse Proxy Admin interface in "Rewrite URL" and "Unrewrite URL" sections.

- Transformed the help windows in the Muse Proxy Admin interface into internal JavaScript windows.

- Created a generic mechanism for managing lists of data in Muse Proxy Administrator Console.

- Created an "Advanced" section in the Muse Proxy Administrator Console interface, placed after

the "Utilities" section, which contains the following subsections: the "Operations" subsection and the "Virtual Machine" subsection.

- Extended the "Configuring Multiple IP addresses.pdf" and "Muse Proxy Backup.pdf" documents with the latest functionalities.

- Implemented an authentication token marker in Muse Proxy to be used for authenticating the rewritten links.

- Updated the Request Handlers configuration files to replace the AU THENTICATION_CACHE_INTERVAL node with the AUTHENTICATION_TIMEOUT node. The values of the these nodes have remained the same. Updated all the Web Contexts con figuration files to add the new AUTHENTICATION_TIMEOUT node. This field was left empty by default. Updated the Muse Proxy Server code accordingly. Small updates with regard to the statistics messages: the 211 code was displaying the wrong number of active requests; the 214 code was missing from the log files; the 210 code was supposed to also display the IP address of the server which received the request.

- Updated the Muse Proxy code to not take into consideration the request header when generating a Navigation Session ID, but a Navigation Sessions counter will be used instead.

- Revised the usage of the Httpd class in Muse Proxy.

- Implemented a Muse Proxy Administrator Console time-out reminder pop-up which warns the user that the Muse Proxy Administrator Console session is close to end. The mechanism includes the possibility to configure the time when the popup must appear.

- Extended the #210 statistics code to include the server IP. Now, the format of the message is : <connection_id><server_ip_address> where: #connection_id - The ID of the connection on which the new data was received; #server_ip_address - The server IP address which was accessed for that connection id.

- Revised the "getParameters", "getGETParameters" and "getPOSTParameters" methods from the Request class to correctly work when a parameter has multiple values.

- Changed the entries from ${MUSE_HOME}/proxy/jaas.config file to use "requisite" instead of "required".

- Done some clean-up work for the Handler class.

13.2 Bug Fixes:

- Fixed some small problems found when testing Muse Proxy 2.6 Build 1.2.

- Removed the unnecessary namespaces from all the stylesheet files. Updated the MuseProxyUtils class, in order to avoid download of unnecessary XML entity files when the request to http://${proxyHost}:${proxyPort}/ProxyInformation or http://${proxyHost}:${proxyPort}/TinyURLGenerator returns an error page which refers a ".dtd" file.

- Fixed somes issue that appeared while testing internal Muse Proxy build.

- There was a small problem encountered to the record links rewriting when the ICE Server was not IP authenticated for Proxy Services. This was fixed.

- In the "Admin >> Utilities >> Rewrite URL" section a rewritten URL could not be generated immediately after the update of the mnm.jar file from Global InfoBase and before any navigation on a rewritten link was performed. When the "Generate" button was clicked, there was not displayed any error message in the www interface but the following message is written in the log file: "Handler@17070e7: [connection.id=7253919C5A5D6EC] Muse Proxy Server encountered an unexpected error which prevented it from fulfilling the request.". This was fixed.

- A problem with the cache mechanism was encountered when the resources served by Muse Navigation Manager were saved in cache and they were later served from cache. The encountered problem was fixed. The cache mechanism was updated in order to completely delete the cached files and their references when they are removed from cache. Updated the web modules that are using cache to synchronize the access to the cache.

- Corrected the information written in the MuseProxyStatistics log for 310 and 311 statistics codes. Previously, for these statistics codes, instead of being written in the MuseProxyStatistics log the "user IP address", it was written the "server IP address". This was fixed.

- Updated the Muse Navigation Manager cookie mechanism to support more date formats for expired cookies dates.

14.0

Changes in Muse Proxy 2.6 Build 10

Release Date: 2012-08-24

14.1 New Features:

- Renamed the Client Session cookie name from "MNMSESSIONID" into "CLIENT_SESSION_ID". This cookie is returned by Muse Proxy to the Client when a request to the web component of Muse Proxy is made and that request does not contain this cookie.

- The following changes were done in Muse Proxy:

    - Now the authentication for the Muse Proxy users, for the Muse Navigation Manager users, for the Muse Proxy Admin users and for the Muse Proxy Services (ProxyInformation, Tiny URLGenerator) users is done separately. For example an user who is authenticated to use Muse Proxy as a regular proxy may be set to not be authenticated to perform the http://${PROXY_HOST}:${PROXY_PORT}/ProxyInformation request.

    - Implemented Web Contexts for all the web components of the Muse Proxy: Muse Proxy Admin, Muse Navigation Manager, Public requests, Services requests and requests which must return a Not Found response page. Any request received by Muse Proxy is categorized and directed either to the proxy component or either to one of the existent Web Contexts in order to be processed.

    - The logon to Muse Proxy Admin requires both IP and user/password authentication. The logon to Muse Proxy Admin interface is now handled using a logon page instead of a Proxy Authorization pop-up. Implemented the logoff action in Muse Proxy Admin. In the "Monitoring" -> "Client Sessions" section, the link to the list of Navigations Sessions for a Client Session is now displayed only when the current Client Session contains at least one Navigation Session inside it.

    - The logon to Muse Navigation Manager requires either IP or user/password authentication. By default the requests handled by the Muse Navigation Manager are IP authenticated from any IP (this is similar to the previous configuration from Muse Proxy 2.6 Build 0.0 or older) so the user/password authentication is not used by default. But if one will restrict the IP au thentication rights for the Muse Navigation Manager, the off-campus users will receive a logon page where they can logon using their user/password access details.

- The logon to Muse Proxy Services (ProxyInformation, TinyURLGenerator) requires either IP or user/password authentication. The users who are not IP authenticated will receive a logon page where they can logon using their user/password access details.

- The requests to URLs of the form http://${PROXY_HOST}:${PROXY_PORT}/${unknownPath} where ${unknownPath} is an URL path not handled explicitly by Muse Proxy will return a "Not Found" response page. In the previous versions of Muse Proxy (2.6 Build 0.0 or below) such a request returned a Proxy Authorization pop-up.

- Removed the DOCUMENT_ROOT, STYLESHEETS, PASSTHRU, MULTIPLE_FILTER_INSTANCES fields. The PROXY_KEEPALIVE field was renamed to KEEP_ALIVE to match the name of other configuration fields. The SESSION_TIMEOUT field was moved in the ${MUSE_HOME}/proxy/modules/handlers/RequestHandlerWeb.xml file in the CLIENT_SESSION_TIMEOUT field. The AUTHENTICATION_CACHE_INTERVAL field was removed from ${MUSE_HOME}/proxy/MuseProxy.xml and there were added the following other similar fields: AUTHENTICATION_CACHE_INTERVAL field in the ${MUSE_HOME}/proxy/modules/handlers/RequestHandlerProxy.xml file and the AU THENTICATION_CACHE_INTERVAL field in the ${MUSE_HOME}/proxy/modules/handlers/RequestHandlerWeb.xml file having separate values. The requests for the proxy and web components of Muse Proxy use now separated au thentication mechanisms. The SUPPORTED_FILTERS and ENABLED_FILTERS fields were moved in the ${MUSE_HOME}/proxy/webcontexts/NavigationManager/profiles/Filters.xml file and re structured as multi-level elements.

- Updated the information written in Muse Proxy Statistics log for the following codes: 212, 280, 281. Added the following new codes for the additional information written in the Muse Proxy Statistics log: 213, 214, 215, 216, 284, 311, 391.

- It was implemented a timeout mechanism for the Navigations Sessions separated from the timeout mechanism for the Client Sessions.

- Now the mnm.jar code is obfuscated.

- Added support to Muse Proxy server to not count the traffic with specific IPs (Target or Client). This is useful in order to allow the Muse Proxy administrator to exclude from traffic counting the local IPs for which there is not done real "internet" traffic. For example if the ICE Server and Muse Proxy server run both in a local network, the Muse Proxy server can be set to exclude the traffic counting with the ICE server. The IPs (Target or Client) for which the traffic counting is excluded can be set using the Muse Proxy Admin interface. By excluding the local IPs from traffic counting, the graphs created in Muse Statistics Monitor for the network statistics data exported by Muse Proxy JMX will show the real internet network traffic. Note: The bytes counted by the traffic attributes exported using Muse Proxy JMX are for the effective data without going deeper into the underlying protocols. They are the effective payload of the TCP/IP packets transferred between the Muse Proxy and the hosts involved in the communication. So they actually represent less data than what is effectively transferred over the network.

- Extended the patterns supported by the Navigation Manager Mode field. Added support for the following escape characters:

- * for the asterisk (*) character;

- \? for the question mark (?) character;

- \\ for the backslash (\) character.

- Added AUTHORIZATION_SCHEME_PARAMETER constant and its associated marker in Muse Proxy. This allows Muse Proxy to rewrite successfully pages which are authorized using either "Basic" and "Digest" authorization methods. Previously only the "Basic" HTTP au thorization was supported for the target sites accessed using rewritten links.

- Updated Muse Navigation Manager to add support for quotes in cookies values.

- Now, the functionality of the HTMLObjectTag filter include the functionality of the Embedde dObjectMovie filter and the latter one was deleted.

- There were some code improvements in the NavigationFilter class. These improvements are delivered through a mnm.jar update.

- Updated Muse Proxy Admin as follows:

  - Updated the "Maintenance -> Update" section in order to use a password control for the "Password" field;

  - Updated the "Backup/Restore" section in order to display the creation date in ISO8601 format for the backup files;

  - Remade the functionality of the "Reset" button of the "Edit" window in "Servers -> Pattern" section;

  - In the "Client Sessions" -> "Navigation Session" section, displayed the value for the following properties stored in the Navigation Session: "REWRITING_PATTERNS", "UID", "charset", "ID". The "ID" field is displayed only if this is not void. Also displayed a bullet before each cookie from the "Cookies" section.

- Added the SOCKET_BACKLOG field in the MuseProxy.xml configuration file. Its default value is 1024. The value of this field is used when creating the ServerSocket objects.

- Created a new Muse Proxy Statistics log which logs in detail all the Muse Proxy run-time activity. The numeric codes used by the Muse Proxy Statistics log are detailed in the Muse Proxy manual. This log is very useful when investigating the Muse Proxy activity for various reasons.

- Created new Muse Statistics Monitor statistics graphs and updated the existing ones for the statistics data exported by Muse Proxy JMX.

- Updated the JMX (for both the monitor and control users) to contain functions for retrieving statistical information. Updated the traffic counting mechanism to split the counted "bytes" section, into 6 sections: "TotalBytesIn", "TotalBytesOut", "TotalClientBytesIn", "TotalClientBytesOut", "TotalTargetBytesIn", "TotalTargetBytesOut". These properties are also exported per each IP in part, on which Muse Proxy listens.

- Added support in Navigation Manager to remove some HTTP Headers when forwarding a request to the Target server. The list of HTTP headers that are removed from the request is configurable at run-time through JMX.

- When the request's Content-Type HTTP header contains Unknown as part of it or if it is not

present then the response is no longer rewritten when its Content-Length exceeds a configured value. This value can be configured at run-time through JMX.

- Previously, the extension of the "filename" attribute from the Content-Disposition response header was considered in deciding whether the content will be rewritten. The code was updated in order to no longer take any decision whether the content will be rewritten or not based on the value of the Content-Disposition HTTP header.

- Added an "Utilities" section in Muse Proxy Admin which can be used to Rewrite/Un-Rewrite an URL through Muse Navigation Manager.

- Added the MusePostID filter in Muse Proxy which is used to rewrite through Muse Navigation Manager the HTTP POST requests done to an URL relative to "/", in the Muse Navigation Manager rewritten pages. This improves pages rewriting for cases when HTTP POST requests to URLs starting with "/" are performed from JavaScript or from Flash components embedded in the page.

- Revised the Muse Proxy code so that for all the actions related to Muse Navigation Manager link navigation to be called a section of code from mnm.jar.

14.2 Bug Fixes:

- Various fixes done to Muse Proxy.

- Previously, for each logon to Muse Proxy a string of the following form: user:password–address:remotePort was written in the log. But this was wrong because the password was written in clear in the Muse Proxy log. This was fixed, and now the password is no longer written in the Muse Proxy log.

- Now, the Muse Proxy Admin is implemented using the "administrator" Web Context and this Web Context has dedicated security rules. In the old implementation in order for an user to be able to logon in the Muse Proxy Admin section, he/she was authenticated twice: once using the rules for the "default" user and a second time using the rules for the "administrator" user.

- Updated the Muse Proxy server code, so that a Navigation Session may be used only by the user for which it was created. This means a Navigation Session can be used only as part of the Client Session which created it, and not by any Client Session.

- A Navigation Session is now reused only when all the request headers and attributes have the same values. This excludes the "Cache-Control" and "Pragma" HTTP headers which change for each request.

- Fixed some small Muse Navigation Manager problems related to the handling of the rewriting patterns: If the value of the NAVIGATION_MANAGER_MODE field in the Source Package profile ended with an "exclude:" pattern which had a space before it, that exclude pattern was interpreted as an include pattern but leaving the "exclude:" part as part of the pattern (so such a pattern did not match any URL). Transformed the relative URLs that were not matched by the rewriting patterns into full native URLs.

- Fixed some small problems found while testing the Muse Proxy functionality.

*   Verified all the Muse Proxy code sections where the Iterator or Enumeration classes are used, in order to see if that code could be used in a multi-threaded scenario and if yes added syn chronization to that code.

*   Updated Muse Navigation Manager to no longer rewrite the pages for which there will be performed a redirect to the original site, after the request will be handled.

*   Fixed some Muse Proxy Statistics log inconsistencies such as:

    *   a line had the "client_session_id" and "the client_ip" parameters missing;

    *   a message having code 283 was missing when deleting a Client Session from the Muse Proxy Session admin console.

*   Reduced the memory used by Muse Navigation Manager when the Client requested a large binary file, but reads just a part of it. Also some threads which remain blocked in certain conditions now are released successfully.

*   Every time when there is encountered an error, Muse Proxy displays an error page. The "error" method from the "Handler" class creates the reply and writes it to the client object. The problem was that this method only wrote the reply headers and not also the reply content. This was fixed now and the "error" method was updated to write both the headers and the content of the reply.

*   Updated the "save" method from the "Updater" class in order to save the value of the "day of week" in the correct field "DAY_OF_WEEK" and not in the field "DAY_OF_MONTH" as previously did. Updated the "schedule" method from the "Schedule" class to assign a default value for a field which is not in the list of fields associated with a moment of type. If the field is missing a message is written in the log file.

*   Previously, when a rewriting pattern from the Navigation Manager Mode field was too general a regular expression error was encountered for specific cases. Now the matching code was updated to check the simple cases without regular expressions and to reject them directly if they do not match. Only the expressions which have passed the initial check will be checked using the regular expressions created based on the received rewriting patterns.

*
*   Updated the Muse Proxy manual so that the Muse Proxy filters to have the same order as they have in the MuseProxy.xml configuration file in the SUPPORTED_FILTERS element.

*   Revised some old sections of Muse Proxy code in order to work better.

*   There was done a small update to the algorithm which rewrites the record links through Muse Navigation Manager in order to better identify the markers names in the URL and place them in the rewritten URL (according with the rewritten URL construction algorithm).

*   The getParameter method from the ProxyAdminFilter class is now used to get the value of the URL from request. This method decodes the input parameter before applying the StringTokenizer object over it. If the parameter, the URL in this case, contains "&" then the value of the URL will be truncated to the first encountered "&" character. A new method getParameter has been constructed which has a boolean parameter which specifies if the input parameter must be decoded or not.

*   Improved the Navigation Manager code to better handle the Cookie HTTP header .

*   Previously, when from Muse Proxy Admin interface, in the "Maintenance" -> "Servers" section,

there was updated a pattern with a value which matched one of the previously matched IPs, Muse Proxy server was shutting down. This was fixed.

15.0

Changes in Muse Proxy 2.5 Build 09

Release Date: 2011–08–18

15.1 New Features:

* Created package–info.java classes for the packages from Muse Proxy included in modulesutil.jar.

15.2 Bug Fixes:

* We now add a default Navigation Manager Mode pattern in the Navigation Session even if there is no pattern defined in the Source Package configuration file. Needed by all sources which use TinyURLs and do not have a rewriting pattern in the Source Package configuration file.

16.0

Changes in Muse Proxy Server 2.5 Build 06

Release Date: 2011-06-16

16.1 Bug Fixes:

17.0

Changes in Muse Proxy Server 2.5 Build 05

Release Date: 2011–05–26

17.1 New Features:

- Added a "Reload Patterns" button in the Muse Proxy Admin/Maintenance/Servers section. This feature can be used when new IPs are added to the physical machine where Muse Proxy runs and the system administrator wants that Muse Proxy to start listening on the new IPs matched by the existing patterns without restarting it.

- When mnm.jar is updated from Source Factory using the Muse Proxy Admin console now it is sent to the Source Factory a corresponding code. In this way, the Source Factory will identify correctly the cases when the mnm.jar was updated using the Muse Proxy Admin interface.

- Used the new MusePostID filter when handling TinyURLs, in order to prevent the cases when the POST data of a TinyURL may be consumed from the NavigationSession by other requests, before being used by the current request.

- Previously, the Muse Navigation Manager could not handle the case when a cookie having a domain property was stored from JavaScript in the document.cookie. The native site sends that cookie to the server at the future requests, but the Muse Navigation Manager was losing it. Because of this the functionality was affected. This was fixed by improving the way in which the cookies from JavaScript are handled in the Muse Navigation Manager rewritten pages.

18.0

Changes in Muse Proxy 2.5 Build 04

Release Date: 2011-03-31

18.1 New Features:

⤳ Now, when interpreting the value of the `StartMuseNavigationManagerMode` marker (which contains the value of the `NAVIGATION_MANAGER_MODE` field from the Source Package profile) the matching is also done against the port and the file part of the `URL` to be matched.

19.0

Changes in Muse Proxy Server 2.5 Build 03

Release Date: 2011-02-22

19.1 New Features:

* The IP of the secure.museglobal.com host was changed so the
  ${MUSE_HOME}/proxy/hosts.xml* files were updated accordingly.

* Removed the UNCOMPRESS field from Muse Proxy configuration file. This was done because the proxy component from Muse Proxy must not uncompress the content processed. The Muse Navigation Manager component of Muse Proxy will uncompress only the pages that will be rewritten.

* Removed the unused Muse Navigation Manager filters: DejaVu.java, Edina.java, NetLibrary.java, NewsBank.java, RefWorks.java, WilsonWeb.java.

19.2 Bug Fixes:

* Previously, when processing cookies containing the HTTPOnly attribute a NullPointerException was thrown. This was fixed.

* Updated some images included in the Muse Proxy.pdf document to not contain the title of the Muse Proxy setup window. Also another image which contained the wrongly spelled text "passowrd" instead of "password" was updated.

* Improved the Muse Proxy cookies management by preserving the cookies sequence, as it is received from the native site, when sending them to the native site.

20.0

Changes in Muse Proxy 2.5 Build 00

Release Date: 2010-10-07

20.1 New Features:

- Updated the Muse Proxy start scripts to match the latest ICE Server start scripts which ensure a better way of handling the messages written to the stdout and stderr streams.

- In JMX, under com.edulib.muse.proxy.update.Updater / mnm / Operations / updatePackage, an error was encounterred "Problem invoking updatePackage: java.lang.NoSuchMethodException:". This was fixed. Now, only a newer version of the mnm.jar file will be downloaded/installed.

- When a domain could not be accessed using Muse Proxy either because of network connection problems with that domain, either because that domain did not exist, the Muse Proxy returned the HTTP/1.0 504 Gateway Timeout response code. Now, in case the connection error with the target host is immediately received the response code is HTTP/1.0 503 Service Unavailable instead of HTTP/1.0 504 Gateway Timeout response code.

20.2 Bug Fixes:

- Now, all the accesses to mnm.jar are made exclusively through the Classloader. This was done in order to prevent the Java Bug which leads to a JVM crash.

- Improvements were done to the Muse Proxy Server Admin sections.

21.0

Changes in Muse Proxy Server 2.4 Build 09

Release Date: 2010-09-01

21.1 New Features:

---

* Small changes done for Muse Proxy JMX: removed the method "saveToDisk" from ServersMBean; corrected the stylesheets for JMX, as the XML format of the response was changed; returned also version and comment for "listBackupFiles" in Updater; updated the functions which retrieve information about: session, connection, Tiny URLs and cache files (search filters, sorting, start, per page); added setter for NAVIGATION_ENABLED_FILTERS and ENABLED_FILTERS, the new value set is a list of coma separated filters; registered to JMX the Navigation preferences; made the TINY_URLS_DIRECTORY read only; when a method encounters an error condition and the operation fails an exception is thrown to the JMX client.

* Added support to rewrite the favicon links.

* Updated the Muse Proxy manual to describe the latest features added in Muse Proxy Admin.

* In Muse Proxy Admin interface there was added the Java Virtual Machine tab with information about the Java Virtual Machine.

* The Muse Proxy Admin interface was recreated to have the Muse Consoles look and feel. Filters and sorting were added to various fields. Added a help for each section. Added a section to display the Muse Proxy configuration file.

* Updated the Muse Proxy manual to document all the attributes and methods exposed through JMX.

* Updated Muse Proxy manual to document the Muse Proxy Admin interface. Added subsections to "2.8 Muse Proxy Admin" section.

21.2 Bug Fixes:

---

*

Fixed a problem that appeared when backing up mnm.jar and no write access was present. Now when there is no write access in the backup directory a corresponding error message is returned.

- Improved the Muse Proxy stop mechanism. Now the user/pass for the administrator user are no longer passed as command line parameters in the stopMuseProxy script, but they are read from the ${MUSE_HOME}/proxy/password.xml file.

- Muse Proxy Admin: Various small fixes were done especially for the filters, sorting, navigation and update on various pages. Additional error messages are now displayed in the www interface when an error occurs. A couple new filters were added for some fields.

- When having a non-void value for the RMI_SERVER_ADDRESS field the JMX Server did not start successfully. This was fixed.

22.0

Changes in Muse Proxy Server 2.4 Build 06

Release Date: 2010-06-10

22.1 New Features:

- Now, Muse Navigation Manager serves to the client the binary files as they are read. This improves the speed of image loading in the rewritten pages, the media files are now loaded on the fly. This greatly reduces the memory used by Muse Navigation Manager when serving the binary content since this content is no longer entirely read and stored in memory before serving it.

- More timeout error codes are now returned by Muse Proxy such as: error code 504(Gateway Timeout – The server was acting as a gateway or proxy and did not receive a timely request from the upstream server) and error code 408 (Request Timeout – The server timed out waiting for the request. The client did not produce a request within the time that the server was prepared to wait. The client MAY repeat the request without modifications at any later time.).

- Previously, the JMX port was only used for "RMI Registry" while for "RMI Server" it was automatically assigned a port which could not be accessed in case of a firewall. This problem is now avoided by assigning the same port for both "RMI Registry" and "RMI Server".

- Removed the CONNECT_TIMEOUT parameter from Muse Proxy configuration file. The Muse Proxy runs as a server and it can only have with a client a read timeout (a server waits forever for any client to come so a server does not have a connect timeout).

- All Muse Proxy parameters defined in the Muse Proxy configuration files and/or in the filters con figuration files are now published through JMX. Changing a parameter value through JMX, will be logged and automatically used on the fly. A new field AU THENTICATION_CACHE_INTERVAL was added to Muse Proxy configuration file; this field specifies the period of time in milliseconds while an authenticated user will be cached.

- Reorganized the Muse Proxy clean-up threads. Added CLEANUP_INTERVAL field in Muse Proxy configuration file which defines the period of time when the clean up runs. Removed SESSION_TIMEOUT_CHECK field from Muse Proxy configuration file and removed TIME_OUT_CHECK field from TinyURL.xml, since they are no longer used.

- Added the TARGET_CONNECT_TIMEOUT, TARGET_READ_TIMEOUT and

KEEP_ALIVE_INTERVAL in MuseProxy.xml and exported them through JMX. TARGET_CONNECT_TIMEOUT and TARGET_READ_TIMEOUT set the connect/read timeout, in milliseconds, for connections made by Muse Proxy to the target sites. KEEP_ALIVE_INTERVAL sets the keep alive requests timeout, in milliseconds, for connections made by Muse Proxy to the target sites.

22.2 Bug Fixes:

- Revised the Muse Proxy log messages to outline the necessary information needed for the Muse Proxy investigation.

- Updated the Muse Proxy start/stop scripts to remove the reference to the libraries that are no longer present in proxy/lib directory.

- Improved the JavaScript parser: the parsing of a JavaScript file of about 1MB now takes less than 1 second, also, now, the JavaScript parsing time is directly proportional with the JavaScript input size.

- Added an additional error condition that fixed a Muse Proxy "NullPointerException" error.

- Fixed the processor spikes problem produced by Muse Proxy when a binary file with no Content-Type but with a Content-Disposition header was rewritten through Muse Navigation Manager. More information(regarding the connection id, the client session id, the navigation session id and the url navigated, also the stream size, the stream read duration and the processing duration) is now written in Muse Proxy log. Now, the thread name for page rewriting threads is set to contain the connection id, the client session id, the navigation session id and the url navigated. This information will appear also through JMX because JMX shows the Thread Name in the Threads information section. The javascript parser code was greatly improved. Added a protection for the cases when a binary content will still reach the rewriting code. If no rewriting is done and the output buffer contains the same characters as the input buffer the output characters are no longer written in the stream. In this way binary streams which would reach the rewriting code will be left untouched and will not be corrupted.

- When having just few idle threads the maximum threads created may exceed the MAX_THREADS configured value. This happened when the number of idle threads is less than MIN_IDLE_THREADS. In this case there were created a number of threads to reach MIN_IDLE_THREADS idle threads. But this was done without checking to not exceed MAX_THREADS. This was fixed.