MuseKnowledge Proxy Release Notes

23 April 2024

Version 5.6.0.5

Notice

Disclaimer

Trademarks

www.museglobal.com

260 Madison Avenue, 8th Floor
New York, NY 10016, U.S.A.
+1 (646) 740-4968 SUPPORT@MUSEKNOWLEDGE.COM

Table of Contents

1.0

Changes in MuseKnowledge Proxy 5.6 Build 05

Release Date: 2024-04-23

1.1 New Features:

- Muse Proxy start up time is now improved for systems with many applications by having parts of Muse Proxy initialization run in parallel with the initialization of the embedded Jetty Servlet Engine - in `MuseProxy.xml` by using the `parallelStart` attribute for `SERVLET_ENGINE` element:

  ```
  <SERVLET_ENGINE_ENABLED parallelStart="true">.
  ```

- Improved the Refresh SAML Configuration and Restart SSO time by loading metadata providers concurrently. Multi-tenant SAML installations loading metadata via `https` can now benefit from parallel initialization in order to significantly lower the Refresh SAML Configuration / Servlet Restart time. In `securityContext-metadata.xml` use `<property name="parallel" value="100"/>` for the CachingMetadataManagerFixed bean - add it as a last property, just before the corresponding closing `</bean>` of `<bean id="metadata" class="org.springframework.security.saml.metadata.CachingMetadataManagerFixed">`

- MuseKnowledge Proxy supports geofencing using IP Location at the Country level. This feature can be used for specifying access rules (allow/deny) by the country identifier, instead of specifying entire IP nets/subnets. The IP<->Country maps are provided by the GeoIP database. Countries can now be specified in ProxyLoginModuleIP.xml where IP rules were available, for example:

  ```
  <IP_RULES>
    <ALLOW country="true">The ISO 3166-1 alpha code for the country.
  Also * can be used for all countries.</ALLOW>
  </IP_RULES>
  ```

  There is a dedicated section in Muse Proxy Admin Console Configuration / IP Location which describes the steps to download and configure the database file for GeoIPLocationService.

- Server Sent Events streams can now have the content data filtered for rewriting in a similar way Web-Sockets frames can be rewritten. This capability is also dependent on the load balancer features (a tunnel timeout is also recommended for the load balancer). On short, a source is configured to rewrite the message data by using the new element

  ```
  <SERVER_SENT_EVENTS applyFilters="true"/>
  ```

  and the exact path for the filters. Details can also be found in the Muse Proxy Sources Profiling documentation.

- This version can, experimentally, run under Java 21. Many changes were carried out to support this, especially in the classpath, and startup scripts. Changes are not directly visible for the system administrator for an installation/upgrade, excepting that `proxy/jaas.policy` is now part of `proxy/java.policy` as a singly policy file has to be used with newer versions of JVM. In order to run under Java 21 set `JAVA_HOME` to point to the JDK 21 installation and/or add the java executable in the system path. Because the lack of full compatibility between Java versions Muse Proxy should be run either with Java 8 or with Java 21.

  - In order to run the Muse Proxy setup using the jar file (which is the recommended way) only this command line should be used for Java 21 (actually for any Java version > 8).

    ```
    java -jar muse-proxy-setup.jar [-console]
    ```

    The other alternative, using `java -cp`, will not work. The above command line assumes that `java` points to Java 21, otherwise the full path must be provided. Note that JDK and not JRE must be used.

  - In order to switch to Java 21 after installing/upgrading, the service files must be manually updated (the setup takes care automatically) both for Windows (`Install MuseProxyService.bat`) and for Linux (`museproxy`). For Windows both `JVM_DLL` and `JVM_EXE` variables must be changed. Note the `jvm.dll` is in a different location for Java greater than 8, namely in `%JAVA_HOME%\bin\server\jvm.dll` instead of `%JAVA_HOME%\jre\bin\server\jvm.dll` (basically there is no `jre` directory in the path). On Windows, the services must be uninstalled and installed again to change the Java version, while on Linux they can be just restarted.

  - To make sure which JVM, Muse Proxy runs within, one can use Muse Proxy Admin console, Advanced >> Virtual Machine, and look for `java.specification.version`, which for Java 8 is `1.8`, while for newer version it is directly the main version, e.g. `21`.

  - When running with JVM 21 please ignore the warnings in the `stderr` when starting Muse Proxy. Java team has been removing a lot of useful libraries, and in the future the Security Manager will be removed as well. In JDK 21 which is a LTE one the Security Manager is still fully functional with regard to the needs of Muse Proxy.

    ```
    WARNING: A command line option has enabled the Security Manager
    WARNING: The Security Manager is deprecated and will be removed in
    a future release
    ```

- Besides the known group processes (`rewrite`, `unrewrite`, `domain`, ...) it is now possible to have a custom processing (`grpProcess`) for the capturing group (`grp`) in the Replacer filter (the default Filter with `FIND/REPLACE`). The function is to be defined in a new `SCRIPT` section via

ECMA (JS) Script. For example

```
<FILTER>
  <FIND><![CDATA[access\('([^']+)','(https://[^']+)'\)]]></FIND>
  <REPLACE grp="2"
grpProcess="uriEncode"><![CDATA[access('$1','$2')]]></REPLACE>
  <SCRIPT>
  <![CDATA[
    function uriEncode(input) {
      if (input !== '') {
        return encodeURIComponent(input);
      }
      return input;
    }
  ]]>
  </SCRIPT>
</FILTER>
```

Support for testing this feature in Muse Proxy Admin will be added in future versions.

- Improved the Authentication section for an Application under Muse Proxy Administrator Application:
  - When the order of the login modules is changed by up and down controls the fact that a module moved was not too obvious. Added some interface elements to give visual feedback.
  - If the login module order was changed and not Updated and the administrator presses `+` (add) the changes were lost. A warning dialog is now displayed.
- Muse Proxy Administrator Console: added Edit option for the "Area Categories" items. Currently in Applications >> Sources >> Sources Groups >> Sources Areas >> Area Categories as there was no option to edit the name of an item, only to delete it and manage its sources. An Edit option is now added, similar to the higher level section where the Areas are managed. The changes are summarized below:
  - Added edit for categories with identifier, name and image fields.
  - Added image listing for Categories and Areas.
  - Added image preview in Edit screens for Area and Categories.

  In the same Area Categories, the management of the sources inside one category was improved:
  - Added "(Drag to arrange)" note related to changing the order of the sources.
  - Hidden sources are now grayed.
  - The zone of the Available sources (to be added) table has now a hover effect so that the cor responding "+" buttons are easier to identify.
- Muse Proxy Administrator Console, Monitoring >> Log Files page was enhanced. It now includes Size and Last Modified information and supports sorting, paging and filtering(e.g. `access-202305` - all access logs for May 2023; or `*-202305*.log` - to obtain all logs, except `ext.log` for May 2023).
- Muse Proxy Administrator Console now shows the `ID` of the Muse Proxy server defined in `MuseProxy.xml` - this is useful for load balanced environments to know what Muse Proxy server is administered.

Upgraded Spring Framework to a more recent version for the `adminRWP` web application – the Applications Management section of Muse Proxy Administrator Console. Spring Security was also added.

- Added an advanced source level setting for SSL Client Enabled Ciphers (`SSL_CLIENT_ENABLED_CIPHERS`). This setting allows a finer control over the enabled TLS ciphers to be used when Muse Proxy acts as a client, i.e. on the target (vendor) end. Some sources might perform TLS fingerprinting and deny ciphers that otherwise are send by the Java HTTPS clients – also, they may prefer fewer ciphers rather then all which are sent by default. Other sources might require only weaker ciphers. The ciphers are enumerated using semi-colon (;) as separator, for example:

  ```
  <SSL_CLIENT_ENABLED_CIPHERS>TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384;TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256;TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</SSL_CLIENT_ENABLED_CIPHERS>
  ```

  The names of the cipher suites which could be enabled for use on an SSL connection can be viewed in Muse Proxy Administrator Console under Advanced >> Virtual Machine, Supported Ciphers (they depend on the JVM and the OS platform). Only a subset of these might actually be enabled by default, since this list may include cipher suites which do not meet quality of service re quirements.

  Use this option only if instructed by MuseGlobal Support Team.

- It is now possible to set a maximum period of time for a continuous use of a session for an application source navigation. It is expressed in milliseconds and it is configurable in the application `web.xml` descriptor file using the `<SESSION_MAX_AGE>` element. In case the session does not time out for this application (it is used frequent enough), the end-user could use this application source action and navigation for a maximum of time corresponding to this configuration. This is a security measure to avoid session hijacking and replication to other systems. It must be set greater than 6 hours. The recommended setting is 10 hours, i.e.

  ```
  <SESSION_MAX_AGE>36000000</SESSION_MAX_AGE>
  ```

  The message that is displayed to the end-user is defined using the keys `SESSION_MAX_AGE` and `SESSION_MAX_AGE_MNM` in the localization file `i18n/proxy.properties` and its language/ application variants (e.g. `proxy_es.properties`).

- A semicolon separated list of request headers to be removed from the HTTP request that are sent to the target sources (vendors) can now be specified in MuseProxy.xml via the new `REMOVE_HEADERS` element. These are useful in case extra custom headers (e.g. `X-Forwarded-Host`) are added by reverse proxies/load balancers. `X-Forwarded-For`, `Forwarded` and `X-Forwarded-Proto` should not be added in `REMOVE_HEADERS` as they were already taken care of.

- The legacy Statistics Menu item in Muse Proxy Admin is now hidden as in the light of the new statistics platform this section in MP Admin is misleading. Although there should be no system in production using this legacy statistics, the menu can be restored using the boolean `STATISTICS_MENU` element in the descriptor `web.xml` file for the Administrator context.

- A new configuration section, Expiry Error (`<EXPIRY_ERROR><PATTERNS method="HTTPMethod">...<HEADER name="RequestHeader">...`), is available both at the source level and, globally, at the `NavigationSession.xml` level. In case the value for

ON_EXPIRY is LOGON* (LOGON/LOGON_REDIRECT/LOGON_REDIRECT_ACTIVE_ANONYMOUS), there are certain cases which do not need to trigger the logon flow, such as requesting css, images or using AJAX calls. The logon flow is meaningless for these cases because the browser does not give control to the end-user if the logon were responded with; instead it is a waste of resources, traffic and access log entries. The cases can be selected by using PATTERNS elements and/or HEADER elements. The PATTERNS element accepts URL-like patterns as the ones in REWRITING_PATTERNS and a method attribute to specify the HTTP method (if missing, the patterns apply to all the HTTP methods).

The HEADER element accepts a JDK-like regex for the value and an XML attribute called name to specify the request header name which is to be used to extract the value from.

By default, NavigationSession.xml comes with the following setting:

```
<EXPIRY_ERROR>
  <PATTERNS method="DELETE">*</PATTERNS>
  <PATTERNS method="PUT">*</PATTERNS>
  <PATTERNS method="PATCH">*</PATTERNS>

  <HEADER name="Accept">^text/css|^image/</HEADER>
  <HEADER name="X-Requested-With">^XMLHttpRequest</HEADER>
</EXPIRY_ERROR>
```

This EXPIRY_ERROR setting can also be placed at the source level, and, in that case, that one will take precedence entirely even if it has just a single child element. Note that for the source level setting a message can as well be customized for the end-user, by using the MESSAGE child element under EXPIRY_ERROR. This message will be, in turn, wrapped in the NAVIGATION_SESSION_EXPIRED_SOURCE localized message defined in i8n properties.

Changes related to EXPIRY_ERROR at the source level are taken into account after a while (REFRESH_INTERVAL - defaulting to 5 min) or after performing Application Update Shortcuts in Muse Proxy Administrator Console, Utilities / Evaluate Shortcut URL.

Changes related to EXPIRY_ERROR at the NavigationSession.xml level are taken into account after performing Refresh Non-Application Web Contexts in Muse Proxy Admin Console under Advanced / Operations page.

- Starting with this version, Muse Proxy communicates license key details to a MuseGlobal service, for statistical and registration purposes.
- Adding new _rwpSkipCookie attribute to JS_MODE so that when it is "true" the document.cookie will not be wrapped by the _rwp* JS cookie methods (it defaults to false).

```
<JS_MODE _rwpInclude="true|false" _rwpInvoke="true|false"
_rwpSkipCookie="false|true"/>
```

Set `_rwpSkipCookie` to `true` only when `_rwpInvoke` is `true` as otherwise the `document.cookie` wrappers are already skipped.

Use this setting only if instructed by MuseGlobal Support Team.

- The `COOKIE_PASS_HTTP_ONLY` flag can be used to still send the `HttpOnly` cookies to the browser side in the injected JS code or when `COOKIE_PASS/COOKIE_PASS_PATTERNS` are used for the corresponding source. Use this setting only if instructed by MuseGlobal Support Team.
- The front-end MKPF application interface is now using Font Awesome 5.15.2.
- The front-end MKPF interface brings a new design based on two cards-like Look and Feel variants that can be configured in Muse Proxy Admin >> Applications >> Interface Options >> Look and Feel selection (or directly in `InterfaceOptions.xml LAF/@default` attribute). Basically, one can choose between:
  - `default` for the normal layout, list oriented with source descriptive text and small icons.
  - `cards` for the layout in which the sources are displayed as cards/tiles. A featured image may be displayed at the beginning of the page (if Enable additional header (`ADDITIONAL_HEADER_ENABLED`) is `true`) , followed by the sources listing. When clicking a card/tile, it flips revealing the source description.
  - `cardsdrop` for the layout in which the sources are displayed as cards/tiles, but without the flipping effect. The source description is expanding below the tile. A featured image may be displayed (if Enable additional header (`ADDITIONAL_HEADER_ENABLED`) is `true`) at the beginning of the page, followed by the sources listing.

A detailed description of the available layouts can be found in our Muse Proxy End User Interfaces Blog entry.

- The front-end MKPF interface has now some sections that can be raw edited in the new Branding zone of Muse Proxy Administrator, under Applications >> Interface Options. Login/Logout Pages Header and Footer, Application Contact, Logout Page, `application.inc`, and others can be viewed and edited to include specific organization information and images.

  The MKPF interface footer is now allowing for a 4 region grid for accommodating logos, contact information, copyright, etc.

- The MKPF template application is configured by default with an Intermediary page for expired `action=source` (non-resource requests) - this will apply when a navigational link is used, that one is expired (and not matching the `EXPIRY_ERROR` patterns) and an expired source link is triggered. To enable the display of this acknowledgement page for expired navigation links, prior to initiating the logon flow the new boolean flag (`<EXPIRED_LINK_STOP status="HTTP Status Code">`) in application's web.xml descriptor is to be used.

  This intermediary page is useful because browsers are no longer keeping in memory all the open tabs and in case of expired links there will be more concurrent logon processes initiated which complicates the SSO/SAML logon flow. This also prevents many secondary requests (such as

cookie-less AJAX or images ones) to initiate the logon flow which is meaningless in these cases. The HTML Freemarker template is ExpiredLinkStop.html and the message is defined in ${MUSE_HOME}/proxy/i18n using the `EXPIRED_LINK_STOP` key. The default value for this flag is false.

- Source `REDIRECT` is now possible via a FreeMarker HTML template page instead of a HTTP `302` status response. Use the new attribute `withPage` of the `REDIRECT` source element.

  ```
  <REDIRECT withPage="SourceRedirect.html">true</REDIRECT>
  ```

  or

  ```
  <REDIRECT withPage="SourceRedirect.html"/>
  ```

  (if the `Sources.xml` IP-based `REDIRECTS` rules are in place).

  The file mentioned in `withPage` must be present in application's `www` directory and can use the model `${actionURL}` as the target URL and, for corner cases, `formParameters` structure captures the post parameters (similar to how it is used in `RedirForm.html`) so that a native form submit can be issued. This page may be necessary in case a source must drop Muse Proxy, but further instructions should reach the end-user. Or, in some cases, depending on IP ranges of the end-user, we may want to display a message to the end-user before going natively.

- The markup from the Source `DESCRIPTION` in `SOURCES.xml` is now interpreted in the front-end MKPF interface.
- For SAML authentication `errorConcurrent.jsp` is global per a Muse Proxy instance. We made adjustments to this file so that tenant specific styles can fit in `errorConcurrent.jsp` if this is necessary. Search for `.tenant-UseTheAliasNeedingCustomization` class - these, together with `c:if test` will need to be duplicated for a tenant requiring customization, and the application id (alias) will need to be used instead of `UseTheAliasNeedingCustomization`.

1.2 Bug Fixes:

- For cases where SAML login is using an IDP which is not supporting SingleLogout (such as Google SAML IDP) an error page is displayed when the end-user logs out. The user is now presented with a more friendly page after logging of.
- Corrections in Muse Proxy Admin Console in pages containing tables with sortable rows - the state of the sorting action was not visually expressed (this is fixed). Also, the sorting icons are now near the header label so that there is no confusion on which column the sorting is applied.
- Fixes related to the front-end MKPF interface `DESCRIPTION` length and the indicators >> and << which are shown according to the details level type (brief/details):
  - If the size of the description is less than or equal to the brief size (default 100) no longer display the maximize/minimize icons (>>/<<).

- When the app is configured with the details level type in the Flat View, the Description is now entirely displayed with the Minimize icon (<<).

- This version brings several changes related to cookie handling to align with RFC 6265 conformance, especially with regard to default paths, multiple cookies with the same name, but different paths / different domains (e.g. parent domain, and sub-domain). These changes might affect just corner cases, however in the unlike event of altering one or more existent working sources (profiles) there is both a global (`MuseProxy.xml`) and source level configuration boolean flag, `<COOKIE_LEGACY>true</COOKIE_LEGACY>`, which will ensure the previous behaviour with regard to these cookie changes at either level. If you encounter such cases let MuseGlobal support know about them for further investigations.

  The changes related to RFC 6265 cookie processing are enumerated below:

  - If the same cookie name appeared on different paths for the same domain or on different domain (e.g. sub.domain.com and domain.com) for a source, only one `name=value` was sent to the source in the request `Cookie` header (the more specific the path/domain, the higher the precedence). Now this is in accordance with the RFC 6265 and all the cookies with same name are sent. The order in which they will appear is given by the specific instructions in the RFC:

    ```
    *  Cookies with longer paths are listed before cookies with shorter
    paths.
    *  Among cookies that have equal-length path fields, cookies with
    earlier creation-times are listed before cookies with later
    creation-times.
    ```

    The list of cookies was already stored in this way for a source, however cookies were filtered when the request was crafted according to older RFC/de-facto rules - and with this version they are no longer filtered (unless `COOKIE_LEGACY` is on `true`).

  - Cookie name and path comparisons for equality and for matches are always case sensitive.

  - Default cookie path and cookie path-matching to test against a request-path are also fully respecting RFC 6265 (section-5.1.4 - Paths and Path-Match).

  - Only one cookie in the `Set-Cookie` header can appear - According to RFC 6265 Origin servers should not fold multiple Set-Cookie header fields into a single. Hence the processing is no longer expecting multiple cookies in the same Set-Cookie response header.

  - The decision to send a cookie in the source HTTP request based on the path was not considering the last path segment if that was not ending with `/`. This is now corrected.

  - The max-age attribute for cookies was not treated in expired comparisons if it was just by itself and not with the Expires.

- Deprecated cookie attributes are no longer supported and stored (version, portList, discard, comment, commentURI/URL).
- Added a cookie flag for host-only cookies (for the ones with no explicit domain) - this is used for matching which cookies are sent for a subsequent request and for testing cookie fully equality (name, path, domain or host-only).
- Cookies where the request-host does not domain-match the domain-attribute are no longer stored in navigation session (they weren't sent but just stored - they are now no longer stored). This change is not controlled by `COOKIE_LEGACY`.
- Http-only cookies are not injected in the JS (unless `COOKIE_PASS_HTTP_ONLY` is `true`) - `COOKIE_PASS_HTTP_ONLY` (as opposed to `COOKIE_LEGACY`) can be used if such cookies are needed in the JS code.

- Excluded patterns with Link Out sources - there are some cases in which an excluded pattern of the main source is still searched globally if the source is link out. This happens when the URL is constructed directly (from JS) and is not under the control of deciding whether to rewrite or not a certain href. In this case, when receiving a rewritten request in the Navigation Session Processor we only verified that the pattern is not matching and then we searched it in the rest of the sources without considering the pattern was really excluded. This is a corner case because it is not advisable to have more sources with the same host patterns but differences in paths. Note that if a pattern is locally excluded it will still be searched globally, for LINK_OUT sources.

  In `NavigationSession.xml` the new attribute `loExclude` of the `ON_URL_MATCH_FAIL` element is a backward compatibility attribute. It should be used only if instructed by MuseGlobal Support. By default, its value is `false`. The rewritten URLs that are matching the `"exclude:"` section in the source `REWRITING_PATTERNS` are not to be processed for Link Out (i.e. searched in other source profiles) at all. The standard rewriting process is not rewriting URLs (even if they match via Link Out in other profiles) that matches the "exclude:" patterns in the normal HTML pages, however, there are many URLs that are relative and managed by complex JS code, and these URLs are automatically made absolute by the browser. We are now also excluding these rewritten requests from the process of Linking Out, with the subsequent requests and they will fall under the `ON_URL_MATCH_FAIL` treatment. If, for some reason, this is breaking existent profiles set `loExclude` globally on `true`, or, preferably do it at the source level (same element `ON_URL_MATCH_FAIL` exists at the source level, too). If it is actually desired to rewrite URLs, but treat them by different linking-out profiles, use the `"excludeLocal:"` section in REWRITING_PATTERNS at the source level.

- Security was enhanced for a corner case related to the URL scheme of the Link Out URLs. When a second source `http://` plain URL is rewritten by a link-out source which was initially under a secure scheme (`https://`) or the application has ENFORCE_HTTPS on true, the eventual `https://` (secure) links under the second navigation session ID were rewritten using plain proxy

Type2/Type 3. They are now rewritten using `https://` (secure).

- Sometimes, in Muse Proxy Administrator Console, when in Configuration >> Administrative Passwords, after performing a browser refresh in that page, and pressing edit for a user, the users table is changed to displays rows of `"undefined"`. This is now fixed.
- Fixing the rare case of a malformed Referer header during indirect authentication such as SSO, SAML, RemoteAuth. Referer can also contain non standard HTTP URLs values. When testing it for loops for SAML or other indirect authentication we now consider it as it was missing if it is malformed. This rare cases could affect vendor mobile apps on older Android OS (e.g. 10) that use prefixed links.
- Muse Proxy Shut Down did not function in case TOTP was activated. For the previous version there was a patch available, now the fix is fully included in this version.

1.3 Recommendations:

- If you are performing an upgrade over an existing installation and the files in `${MUSE_HOME}/proxy/i18n` were customized, make a backup before the upgrade and then re-do the customization. Note that new keys, `EXPIRED_LINK_STOP`, `NAVIGATION_INTERRUPTED`, `SESSION_MAX_AGE`, `SESSION_MAX_AGE_MNM`, `ACCESS_TARGET_SSE`, have appeared and must be merged and eventually translated to other languages if your system is using them. Currently, the out-of-the-box languages are English (`proxy.properties`) and Spanish (`proxy_es.properties`).
- Advertise only documented entry points URL in external systems (including social media). Do not copy and paste URLs from the browser while navigating a resource through MuseKnowledge Proxy. The link itself might not even be bookmarkable even if the vendor is accessed directly. If the native link itself is bookmarkable, it is OK to advertise prefixed URLs `https://proxy.example.org/APP?qurl=https%3A%2F%2Fresource.vendor.com%3FbookID%3D123456789` or `https://proxy.example.org/APP?sourceID=VendorDbID` but a follow up, internal Rewrite by Host or Rewrite by Path URL, such as, `https://0f1060xpe-p1-y-https-resource-vendor-com.proxy.example.org/bookID=123456789`, must never be used in external systems.
- If you are performing an upgrade over an existing installation and the `PRIMARY_COOKIE_CONFIG` and `SECONDARY_COOKIE_CONFIG` entries from `MuseProxy.xml` were previously customized (e.g. prefix change), then the customization needs to be performed again. Check the previous file `MuseProxy.xml.bak` where the older entries are preserved.

2.0

Changes in MuseKnowledge Proxy 5.5 Build 05

Release Date: 2022-09-28

2.1 New Features:

- Implemented post-authentication logic for the HMAC login module via `SCRIPT` section similar to what is available for SAML and LDAP cases. The configuration file, `ProxyLoginModuleHMAC.xml` from the template applications, comes with explanatory comments and a small sample.
- Implemented `ENFORCE_HTTPS` for more WebModules. Up to this version, in order to redirect from `http://` to `https://` there were two possibilities:
  1. Do this globally if `PORT enabled="false"` and `SSL_PORT enabled="true"` in `MuseProxy.xml`.
  2. Do this per application basis via `ENFORCE_HTTPS` in the application's `web.xml`.

  To have a finer-grained control (when the global redirect setting is off) `ENFORCE_HTTPS` is now available to be set selectively for `Root`, `Administrator` and `Public` web modules in their cor responding `web.xml` files.
- Custom HTTP headers defined in the `WebContexts.xml` file can now be reloaded without a full proxy restart. This can be done via Muse Proxy Admin using the `Refresh Custom HTTP Headers` button within the page `Advanced/Operations`.
- Also, related to Custom Http Headers, the WebModuleSAML in `Root/web.xml` is now having two new boolean parameters, `DISABLE_CUSTOM_HEADERS_RELAY` and `DISABLE_CUSTOM_HEADERS_LOCAL` in order to avoid passing the custom headers for resources relayed from the ssoRWP* servlets as well as potential error pages generated by Muse Proxy when accessing these servlets. These will allow, for example, for security headers to be set for Root context, but not inferred for the LTI authentication responses steps.
- Reloading non-application Web Contexts configurations is now possible without an entire server restart by using the `Refresh Non-Application Web Contexts` button within the page `Advanced/Operations` from Muse Proxy Administrator Console.

- The Learning Tools Interoperability® (LTI®), version 1.3 standard can now be used to include a MuseKnowledge Proxy Application or proxified source directly within the Learning Management System (LMS) platform as a Tool in a new window. LTI version 1.3 improves upon older versions by moving away from the use of OAuth 1.0a-style signing for authentication and towards a new security model, using OpenID Connect, signed JWTs, and OAuth2.0 workflows for au thentication. LTI 1.3 still acts as a half way SSO, in the sense that starting from the platform (LMS system) the access to the MuseKnowledge Proxy Application is seamlessly. The LTI 1.3 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. Hence, once a user is authenticated to the platform (LMS) (s)he can access a MuseKnowledge Proxy application or source being defined as a Tool.

  Because the new version comes with extra security and this also implies a greater complexity for configuration than previous LTI versions, extensive instructions are given in Muse Proxy Ad ministrator console, in the new page `Configuration / LTI 1.3 Authentication`. MuseKnowledge Proxy acts as a Tool and leverages LTI Core 1.3 and Deep Linking 2.0 (a service part of the LTI Advantage) processes in order to provide seamless integration of Proxy Application interface and proxified sources directly into the LMS platform.

  There are multiple configuration levels, some configuration steps depending on the platform (e.g. Canvas self-hosted, Canvas cloud-hosted, Moodle) and because there are circular dependencies between the tool and the platform multiple iterations are necessary. Briefly the settings to be done consists in:

  - Generating instance level key pair to be used for the integrations - LTI 1.3 is based on JSON Web Key Set (`JWKS`) and this requires key-pairs (private key and public key) for the parties involved, as well as configuring the public JWKS URLs of the peers.
  - Application Level settings consisting in authentication groups with login modules and Iframe related HTTP Security headers. Note that `AuthenticationGroups.LTI13.xml` is backward compatible with LTI 1.0 because the `groupID=2` still points to `ProxyLogin ModuleSSOLTI.xml` and there is a `groupID=3` which points to `ProxyLogin ModuleSSOLTI13.xml`. This means that older LTI1.0 resource-links from the LMS Platform pointing to the Muse Proxy Application will still work. If a different `groupID`, or the `groupID=2` is desired then this can be changed in `AuthenticationGroups.xml` file and then, via, `lti.properties`, for the corresponding instance by using `.group-id=` property, for example:

    ```
    lti.clients.MuseProxyFoundation.definitions[0].group-id=2
    ```

    or change the global `lti.group-id=2` property if this is true for all the tenants and instances, or just the tenant specific via `lti.clients.MuseProxyFoundation.group-id=2`.
  - Settings for the tool in Muse Proxy's LTI 1.3 main configuration file `lti.properties`

depending on the LMS platform and Muse Proxy application; this implies configuring the end-points of the platform and the public JWKS, while for client ids, deployment ids there must be used placeholders which will be replaced after another iteration.

- Settings in the platform where the end-points of the Muse Proxy Application Tool are configured. This is either done manually or, for Canvas, through a metadata-like configuration JSON that can be accessed at the URL `https://proxy.domain.com/ssoRWP3/config/{ApplicationID}/instance/{instance}/canvas.json` , for example: `https://proxy.domain.com/ssoRWP3/config/MuseProxyFoundation/instance/0/canvas.json` .
- After the tool is added in the platform client ID and eventually deployment ID values needs to replace the placeholder.
- There are also advanced configuration for corner cases which are also described in the Muse Proxy Administrator Console, for example, in-line specification of public key set via `.key-set`, instead of the `.key-set-url`.

There are more placements in a platform available for the tool, for example, in Canvas: Account Navigation, Course Navigation, Editor Button, Link Selection. The Placements available for Muse Proxy tools require new window, due to all the browser security. However, for selecting and embedding a link (Deep Linking), the Muse Proxy Application is still launched inside an iframe so make sure the corresponding iframe related security HTTP headers are correctly set/unset and that third-party cookies are enabled in the browser. For modules (Link Selection) you will have to tick the `Load in a new tab` check box.

- In Muse Proxy Administrator console, Manage Applications section one can now search for a source in all the applications and perform modifications to it. The fields to query can be chosen from `Souce ID`, `Source Name` and `Source Description` or a mix of them. The comparison type can be selected as `Exact match` / `Contains` / `Wildcard`.

  Another extension in the console is that Quick Filter for Applications screen and for Application/Sources screen can now be applied selectively on `ID` and `Name`, because some-times searching just through the IDs is more helpful.

- It is possible to configure the whole Muse Proxy instance to extend the navigation session ID for application sources if either application code or source code is greater than `1295` to `aaagsssssss` (3 digits for application and 3 for source codes). The drawback is the increase of DNS label which is limited to `63` and more host mappings might be necessary. Note that the extension of the ID size happens only if necessary - for the same application there could be both `aagsssssss` and `aaagsssssss` navigation session IDs depending on the code of the sources. In order to make this extension possible when codes are exceeding `1295` set the flag `EXTENDED_NAVIGATION_SESSION_ID` in `MuseProxy.xml` on `true` then either restart the

server or perform a Refresh Configuration from `Muse Proxy Administrator Console / Operations` page.

- Pattern matching is now achieved, by default, through the new flag `<JDK_REGEX>true</JDK_REGEX>` in `MuseProxy.xml`, based on the regular expression API from JDK, which is more efficient in the usage of the stack and processor than the legacy ones.

- A new source configuration element, `MIME_MAPPING` will solve cases where the provided content type is unknown to Muse Proxy and would not be interpreted, or when it is wrongly advertised or even missing and we need to direct the rewriting engine to interpret it differently. The structure of the configuration in the source profile is given below

```
<MIME_MAPPING from="Content-Type header value from Reply (can be empty
for cases when there is no Content-Type)" to="a defined category"
patterns="Optional - URL Patterns for this source for when this
mapping is applied.">
```

  For example, the following entry:

```
<MIME_MAPPING from="application/xhtml+xml" to="HTML"
patterns="www.museglobal.com">
```

  will treat the relayed responses from www.museglobal.com domain having the header `Content-Type: application/xhtml+xml` as HTML and applies the corresponding rewriting rules to the HTML scope.

  The values for the attribute `to` are: `HTML`, `JAVASCRIPT`, `XML`, `JSON`, `CSS`, `TEXTPLAIN`, `BINARY`. There can be multiple `MIME_MAPPING` elements in a source profile.

- For some corner cases, also related to Content-Type, the new source level option `GZIP_PATTERNS` can be used to make sure the reply gzip encoded content (Content-Encoding) is inflated so it can be filtered, if needed. Use this option only if instructed by the MuseGlobal Support Team.

- A new boolean flag, `KEEP_HEADER_ORDER`, for global and individual source configuration is available. If the flag is true for a certain source best efforts are made to keep the request headers order when relaying as a rewriting proxy. Normally, according to the HTTP RFC, the order of distinct header fields shouldn't matter but request fingerprinting started to be used more often for security reasons on vendors side. If using a reverse proxy/load balancer it should as well preserve the order, otherwise the effects will not be achieved.

  The flag can be true for a certain source if the global setting `KEEP_HEADER_ORDER` is `true` in `MuseProxy.xml` and there is no source configuration or, if the source level setting is present and is `true`. If the source setting is present but it is not `true` then the order will not be preserved. In other words, the source level setting takes precedence. For the beginning it is recommended to use this flag, isolated, just for the sources needing it and not using the global setting.

- The front end `MKPF` application interface features a new Tools section where `"URL Encode and Prefix"` and `"Un-Rewrite URL and Prefix"` are possible. The prefix bit refers to creating an URL as an entry point to Muse Proxy, for the current application either from a normal vendor link, or back from an URL that results in the process of navigating a certain proxy source (Rewrite by Host or Rewrite by Path), and which should, actually, never be bookmarked or advertised as such.

- Authentication using 2FA (TOTP - time based one time password) for the administrator user is available. This is working with 2FA Time base OTP applications such as Google Authenticator, Authy, TOTP Authenticator. Follow the instruction from Muse Proxy Administrator Console, for the admin users (use the new key icon in the users table). Note that Muse Administrator accounts are also required to be IP authenticated as well, so what is achieved now by TOTP is actually a 3FA.
- New Statistics log codes for source identification were added:
  - `261` - Message code used for source requests with `url/qurl=` when this parameter cannot identify a source. This is the important one for a potential analysis of the Statistics log in order to review these URLs and add corresponding sources for them if the institution has subscription, or the legitimate URLs can be grouped into patterns and added into a special hidden source with `REDIRECT` flag on `true`. However, the revision procedure must ensure that the URLs are safe and point to a real resource needed for the institution.
  - `262` - Message code used for source requests with `url/qurl=` when this parameter cannot identify a source because of an extremely rare case of unexpected exception.
  - `263` - Message code used for source requests when the source is not found for the current source group.
  - `264` - Message code used for showing what `sourceID` was used for a given `url/qurl=`.
- A new global option, `SSL_DISABLED_CIPHERS`, is available. Its purpose is for Muse Proxy instances facing directly the internet to score grade A (hence this is not needed for SSL Termination scenarios) and at the same time use weak ciphers against vendors still requiring weak ciphers. For example, add in `MuseProxy.xml` the following element after `</SSL_PROTOCOL>`

```
<SSL_DISABLED_CIPHERS>TLS_RSA_WITH_AES_128_CBC_SHA;
TLS_DHE_RSA_WITH_AES_128_CBC_SHA; TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA;
TLS_RSA_WITH_AES_128_CBC_SHA256; TLS_DHE_RSA_WITH_AES_128_CBC_SHA256;
TLS_RSA_WITH_AES_128_GCM_SHA256;
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256; TLS_RSA_WITH_AES_256_CBC_SHA;
TLS_DHE_RSA_WITH_AES_256_CBC_SHA; TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA;
TLS_RSA_WITH_AES_256_CBC_SHA256; TLS_DHE_RSA_WITH_AES_256_CBC_SHA256;
TLS_RSA_WITH_AES_256_GCM_SHA384;
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384;
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256;
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384</SSL_DISABLED_CIPHERS>
```

  This was needed because JVM `jre/lib/java.security` does not differentiate between server and client disabled ciphers, and for the other end, client, Muse Proxy should still be able to use some weak algorithms to communicate with vendors not keeping the pace with the security re quirements.
- There are reverse proxies / load balancers that cannot remove the existent `X-Forwarded-For` headers from the request. They just add one, or one value to it in the end. In order to make sure the correct value for client IP is detected in such cases the processor `mode` attribute for `X-Forwarded-For` field can now be specified. It defaults to `"right"`. It tells what will the client address be in case there are more IPs sepparated by ',' (in case there are more X-Forwarded-For fields they are equivalent to a comma separated list, so `"right"` will assume the last one). `mode="left"` is also supported, but not recommended.
- `TRUST_X_FORWARDED_PROTO` is set to `false` in `MuseProxy.xml`. If a load balancer/reverse proxy is used and this is not using the HAProxy PROXY Protocol then set it to `true`.

- SAML Discovery for IDP Selection was extended in order to add support to specify friendly names for the configured IDP's, as well as, leaving room for setting other properties. If, in `securityContext-metadata.xml` you configured the `<bean id="aliases" class="java.util.HashMap">` for Local Discovery (this was provided in an XML comment section), for this new version the bean structure must be updated according to the newest `securityContext-metadata.xml` from a fresh install - the structure is between the comments `<!-- START_LOCAL_DISCOVERY -->` and `<!-- END_LOCAL_DISCOVERY -->`. This includes referencing beans for IDPs where more properties entries, such as `fiendlyName` are specified.
- Created a new default SAML key pair for Muse Proxy `ssoRWP` SAML end-points; the previous one is expiring. The alias is called `only4test`, available in fresh installs as an out of the box con figuration for tests and must not be used in production.
- HTTP headers to avoid browser caching were added for more Muse Proxy responses cor responding to non-static resources, such as the `action=sources` response for the application interface.
- Added the requested URL scheme information in the Debug log (on `NOTICE` level) on the line `"Received request for"` as `[scheme=https]` or `[scheme=http]`. It is useful to have it especially as browsers are discontinuing plain `http://` (step by step) and now in FireFox one cannot mix `https://` with `http://` due to cookie visibility. Note that for HTTP Tunnels the scheme is `http`, because the HTTP plain connection is actually used, for example:

  ```
  2022-09-05T14:45:18.932 EEST NOTICE: Handler@57232559:
  [connection.id=4FD7F1686D409126][client.ip=217.156.14.167][scheme=http
  ] Received HTTP tunnel request for: "support.museglobal.com:443".
  ```

- Login Modules are also logging the `application.id` besides the `connection.id` in the same log line in the Debug log. This is helpful to identify login failures for a certain application in a multi-tenant environment without using a stateful parser.
- The MKPF template contains a workaround for SAML/SSO authenticated proxy to have proxified links/sources or just the application interface included in IFrame, if the IFrames implementation is mandatory required (assuming third-party cookies are enabled). IFrame is not recommended for in tegrations because of the browser security enforced and even vendor protection for this. IDPs are usually denying frame inclusion, and, in case multiple proxified iframes are in the same page, concurrent authentication to SAML/SSO can result in errors because the HTTP session needs to be created by a single flow.

  A new entry point page, `FrameSSO.html`, is to be used for such not recommended cases which detects the authentication status to the proxy (e.g. if authenticated or not) and, if not authenticated, triggers the SAML authentication workflow into a popup window. The end-user must click to open the popup, otherwise the browser blocks the popup. When the authentication is completed, the IFrame(s) will start loading the proper content. This file is used together with `PopupSSO.html` which needs to be un-commented in `authenticationFlow` in the application `web.xml`; also uncomment `<INCLUDE>/static/FrameSSO.html</INCLUDE>` from `PRE_PROCESSING` in the same file.

  A sample prefixed URL to be used as IFrame src is:

`https://proxy.example.com/MKPF/static/FrameSSO.html?url=https://support.museglobal.com/ipaddr.php` where `proxy.example.com` is the instance of the proxy, `MKPF` is the application root and the `url` parameter contains the resource url (in this example the IP display page from `support.museglobal.com`).

The `FrameSSO.html` page also supports `sourceID=` and `qurl=` parameters. Note that this workaround does not apply for LTI1.0/1.3 as these are one-directional.

- Parameters from the source profile are now resolved in the `value` attribute in the HEADER rules in the Replacer `FILTER` via `${PARAM_NAME}` style. The source parameters can also be bound to `SCRIPT` for `HEADER` rule if `SCRIPT`'s attribute `useParameters="true"` is present - they can be accessed as the `parameters` map - access them only as read only to read parameters values. Also, `logUserID` is now bound to the `HEADER SCRIPT` in order to process it.
- The processing of CSS content type was optimized offering better CPU times, especially for large CSS.

2.2 Bug Fixes:

- There are small changes in the JavaScript inserted in the rewritten pages in order to comply with HTML standard. For example, `<SCRIPT LANGUAGE="JavaScript">` was replaced with `<script type="text/javascript">`. In case you are using automatic comparisons tests for detecting rewriting issues please adjust them to comply with these changes.
- `FOLLOW_REDIRECTS` from source profiles is now read correctly, ignoring the case. For this flag if its value wasn't case sensitive `"false"` it was finally set to true. So now if `"False"` is used then this will be treated as false.
- For SAML, SSO, LTI, AdminRWP relaying Muse Proxy is correctly dealing with HTTP HEAD requests with responses not having a body, even if they contains `Content-Length`.
- The following behavior was fixed in MuseKnowledge Administrator Console / Manage Ap plications: when importing a source having an ID different from the profile file name, the import action is saving the profile using the `IDENTIFIER + .xml` instead of using the file name given by the `CONFIGURATION_FILE` in the export metadata. The import seems successful but if you want to access the source, the error `"The profile configuration file was not found."` was yielded.
- After a server restart, loading persisted sessions failed if an application ID with active sessions is no longer available (renamed/deleted/critical configuration issues) before/during a server restart. This is now fixed and the sessions for the rest of the applications are correctly restored.
- In Muse Proxy Administrator console, Manage Applications section the buttons under Raw Edit did not get proper focus on recent updates of Chrome and Edge. This is now fixed.
- Logging to Debug log, Access log and Statistics log is now recovering after disk full events are resolved, even if more scheduled rotation occurs. Up to now the logging recovered only if disk

space became available in the same time window with the initial disk full event (before the log file rotated).

- `DISABLE_TARGET_KEEP_ALIVE` has effect on the first requests, too, but only when using `HttpModuleApache`. If there are multiple URLs in the profile (Extract and Navigate) it is, very rare, possible that the connection is not reused by the server, driving to failure for the sub-sequent URL request.

2.3 Recommendations:

- If you are performing an upgrade over an existing installation and the files in `${MUSE_HOME}/proxy/i18n` were customized, make a backup before the upgrade and then re-do the customization. Note that new keys, `REMOTE_DENY_WITH_MESSAGE`, `INVALID_REWRITTEN_URL`, `UNRECOGNIZED_REWRITTEN_URL`, `HTTPS_REWRITTEN_URL_WRONG_SEPARATOR`, `HTTP_REWRITTEN_URL_WRONG_SEPARATOR`, `IMPROPER_REWRITE_BY_HOST`, have appeared.
- If parsing the Debug log for errors, double-check the parsing scripts first as there are small modifications such as adding `application.id` for log lines related to the ProxyLoginModule-s. Also, extra blank space was removed for more log lines as well as the URL scheme was added. Note that un-documented changes might appear in the Debug log between versions.
- If you are performing an upgrade over an existing installation and the `PRIMARY_COOKIE_CONFIG` and `SECONDARY_COOKIE_CONFIG` entries from `MuseProxy.xml` were previously customized (e.g. prefix change), then the customization needs to be performed again. Check the previous file `MuseProxy.xml.bak` where the older entries are preserved.

2.4 Known Bugs:

- If 2FA (TOTP - time based one time password) was configured for the administrator user then stopping Muse Proxy via `stopMuseProxy(.bat)` script is not working. In order to stop Muse Proxy use the `SIGTERM` signal on Linux while on Windows use the stop service against the Muse Proxy service inside Windows Services console. This has been corrected for the future and a patch is also available via MuseGlobal Support channels for Muse Proxy 5.5 Build 05.

3.0

Changes in MuseKnowledge Proxy 5.4 Build 02

Release Date: 2021-12-21

3.1 Bug Fixes:

- The Apache Log4j 2 library that is used by few components in Muse Proxy has been updated from 2.16.0 to 2.17.0 according to https://logging.apache.org/log4j/2.x/security.html.
- Installing and controlling Muse Proxy as a Windows Service from the Muse Proxy Setup was failing if `MUSE_HOME` directory contained spaces and Java 1.8 update was at least 292. This is now fixed.

4.0

Changes in MuseKnowledge Proxy 5.4 Build 01

Release Date: 2021-12-17

4.1 New Features:

- The TLSv1.3 standard can now be used on both ends of Muse Proxy if running under JVM version 8, at least, update 262. For the server end Muse Proxy is now configured with TLSv1.2 and TLSv1.3 only as enabled protocols. For sources' end TLS v1.3 can be configured if needed, however, there are still many servers not supporting it, hence, by default TLS v1.3 cannot be enabled for all the sources.
- In MuseProxy Admin the `Utilities` / `Un-Rewrite URL` tool is able to un-rewrite Rewrite By Host URLs.
- The Find/Replace filter is able to process (`rewrite`, `unrewrite`, `rewriteHostHTTPS`, etc) strings that represents the Base64 encoded form of an URL/Host and then re-encode it back. By using the attribute `base64` we can achieve decode, rewrite (or any other group process) and then re-encode in Base 64 as long as the replacement group represents the Base64 sole representation of an URL/host.

```
<FIND><![CDATA["(aHROcHM6[^"]+)",]]></FIND>
<REPLACE rewrite="1" base64="true"><![CDATA["$1",]]></REPLACE>
```

4.2 Bug Fixes:

- The Apache Log4j 2 library that is used by few components in Muse Proxy was updated to 2.16.0. Note that the core logging mechanism of the Muse Proxy is based on the in-house built logger. Logs such as `access.log`, `MuseProxy.log`, `MuseProxyStatistics.log` are produced using the in-house logger, and not Log4j. Also, make sure that the entire Library system stay safe by updating any Java based related software used in the portal, authentication services (LDAP, SAML, etc.), log processing, monitoring infrastructure, etc. accordingly: https://logging.apache.org/log4j/2.x/security.html.

- The `LAX_HEADER_PARSE` in MuseProxy.xml was not propagated as a setting. However, it is strongly recommended to avoid using this configuration element.
- The Server Name Indication workaround to support older servers refusing SNI in the same MuseKnowledge Proxy instance (Java Virtual Machine can only be globally configured for a single scenario) was updated. Now we send an empty host in cases the exception is `"handshake alert: unrecognized_name"` or `"received handshake warning: unrecognized_name"`. The cases needing this workaround are extremely rare.

5.0

Changes in MuseKnowledge Proxy 5.3 Build 05

Release Date: 2021-03-18

5.1 New Features:

- An Audit Report can now be generated using the MuseKnowledge Administrator Console / Manage Applications zone for the current selection of applications. The report includes a cumulative section, individual application details, and source profiles details. When more than 10 applications are selected only the cumulative summary is available.

  The Cumulative Summary contains License Details, a source oriented view, representing sources by applications distribution, listing unique sourceIDs (and for each one the applications where it is available) and an application oriented view, listing the sourceIDs available in each application. Other counters such as Number of Sources, Number of Sources Groups and Number of Sources Used In Groups are available. Exceeding the license limits are emphasized through graphical means.

  If individual application details are included there will be a separate tab for each application, each one containing an application summary (ID, Name, Expiry Date, Enforce HTTPS, etc.), information about the authentication groups with the authentication modules and the source groups listing the source IDs contained. If source profiles details are also included then for each source pertaining to a source group its main details will be included. This report may be extended/ modified with future release.

- In the MuseKnowledge Administrator Console / Manage Applications area, the first time loading of application list was improved by minifying the JS files and bundling them into a single one so that the number of HTTP requests is decreased.

- Testing and managing the Authentication for LDAP Configuration in the Administrator console is now possible. In order to ease the LDAP authentication configuration visual management and step by step tests are available. Check Network Parameters, Check Root (Bind) Authentication, discovering Search Bases, searching for the user (or any Search String), entirely testing the whole

chain with a user and password are now possible in an intuitive manner from the same screen allowing visual edit. Raw edit is still available.

- In this release the JavaScript libraries involved in the application interface templates (`MKPF` and `MuseProxyFoundation`), in the static pages and in the Administration console were updated to newer versions.
- Starting with this release the access to a MuseKnowledge Proxy Application is no longer possible through an URL using IP / localhost / short host name (e.g. `http://localhost:9797/MKPF` or `https://217.156.14.167:9443/MKPF`, or `https://proxy:9443/MKPF`), unless the `SERVER_NAMES` is defined in the `MuseProxy.xml` configuration file. Although we list wildcard FQDN and wildcard SSL certificate(s) as pre-requisites, and it is clear that a HTTP server requires some infrastructure, the immediate availability of applications after an installation has proven to create false expectations.
- The Priority cookie attribute is now set to MuseKnowledge Proxy Session and ID cookies. This attribute is interpreted by Chrome, Microsoft Edge and other Chromium based browsers, while the others are ignoring it. This way, in case the per domain cookie capacity (usually 180 cookies) exceeds on the end-user browser side, the proxy session cookie is not evicted hence the session could still be maintained. This is configurable through the `priority="High" XML` attribute of the `PRIMARY_COOKIE_CONFIG` and `SECONDARY_COOKIE_CONFIG` elements in the `MuseProxy.xml` configuration file. Note that these two settings, `PRIMARY_COOKIE_CONFIG` and `SECONDARY_COOKIE_CONFIG`, are overwritten when upgrading, so the customized bits need to be manually reverted.
- `HttpOnly` Cookies are now filtered out when `COOKIE_PASS/COOKIE_PASS_PATTERNS` are used. By default, MuseKnowledge Proxy is managing the cookies with the vendor side and does not send vendor cookies to the browser. However, because the vendor JavaScript might expect certain cookies, the source level elements `COOKIE_PASS` and `COOKIE_PASS_PATTERNS` can be configured to allow vendor cookies to percolate to the browser side. However, it seems better that, if the engine is deciding to send a Set-Cookie to the browser, the cookie should not be `HttpOnly`, because anyway the reason is to have JavaScript code using the cookies and a cookie with `HttpOnly` attribute cannot be used by JavaScript. Sending the `HttpOnly` cookies uselessly increases the number of cookies from the browser level.
- When generating new SP metadata via Metadata Administration in MuseKnowledge Proxy Ad ministrator Console SAML secure, `https://`, SAML end-point links (e.g. the AssertionCon sumerService links) and `entityID` are always suggested even if administrator access is done through `http://` scheme (which is not recommended). A plain (`http://`) base URL for the SAML end-points can no longer be used when generating new SP metadata.
- Expired link corresponding to applications with indirect authentication (SAML, SSO, RemoteAuth) are not triggering the login flow if the HTTP request headers `Purpose: prefetch` or `Sec-Purpose: prefetch` are present.
- The login flow is no longer initiated for expired (missing or expired session cookie) proxy links of vendors still on plain, `http://`, if the application is configured with `ENOFRCE_HTTPS` and without the secure attribute of `PRIMARY_COOKIE_CONFIG` in MuseProxy.xml. This is because browsers are implementing the Schemeful Same-Site behaviour where different schemes `http://` and `https://` of the same site are considered totally distinct parties and non-secure cookies are no longer shared. Hence, the login process cannot be initiated on `https://` and go back to

`http://` without the session cookie for vendors still on plain `http://`, as this will trigger redirect/form loops. Proxying plain `http://` sources was no longer supported in a third-party environment (e.g. iframe, image, widget, POST method) since the previous MuseKnowledge Proxy version. Now, starting with this version proxying plain `http://` vendors in browsers im plementing rigid Schemeful Same-Site can no longer be supported. Without stopping the login flow the result would be a redirect loop capped by the browser or an infinite post form loop (depending on the type of authentication) and this can be more dangerous. Although not recommended, if, for certain reasons, the same behavior as before (no redirect protection) is wanted when `PRIMARY_COOKIE_CONFIG` is not defined or does not have the secure flag, then use the new XML attribute `httpSchemeful="true"` for the `ENFORCE_HTTPS` element in the `web.xml` application configuration file.

- The client facing Keep Alive mechanism was changed so that it is more accurate and does not have corner cases side effects when closing the socket.
- A Request Timeout for reading the entire HTTP request is now available in the `MuseProxy.xml` configuration file. Note that it must be significantly higher than the `KEEP_ALIVE_INTERVAL` and `READ_TIMEOUT` value. In case a load balancer or a reverse proxy is used in front of the MuseKnowledge Proxy they normally take care of this aspect.
- The `logUserID` is now available in the Application FreeMarker model and can be accessed through `session.getLogUserID()` in the FreeMarker templates, in cases of applications which may have clear human readable `userIDs` and no displayName is available in the post-authentication properties or the authentication module does not have such properties.
- The relaying of responses represented as chunked transfer encoding was further improved in two directions:
  - Client Keep-Alive is also ensured for filtered content that was retrieved as chunked Transfer-Encoding. This wasn't happening previously.
  - The memory usage is improved in the cases where binary content was heuristically detected for unknown content type.
- The debug log now adds more information about client connection, especially on some client side errors so that the client / peer IP (end-user / load-balancer IP in case the end-user IP is not encoded by the load balancer) is also part of the same log entry without the need to identify it in previous entries or in entries from other log files. For the vendor side of connection efforts were made so that the `source.ip` and `target.hostPort` are logged in the same error entry for some errors. The `client.ip`, `source.ip` and `target.hostPort` information is, however, not duplicated for each log entry with a `[connection.id=...]`. Hints about HTTP Tunnel connections are now present.

  For `SSLHandshakeException` and `SocketException` we no longer try to send errors to the client because, obviously, the connection is broken or could not have been established (`SSLHandshake`) and these resulted in useless log entries.
- There are more cache control response headers used in some responses, such as the indirect au thentication redirect ones, so that these responses are not cached by the browsers.
- The HTTP request parsing is stricter with regard to the `CR LF` separator. Basically, MuseKnwoledge Proxy sticks more to the RFC: "Although the line terminator for the start-line

and header fields is the sequence CRLF, a recipient MAY recognize a single LF as a line terminator and ignore any preceding CR." [https://tools.ietf.org/html/rfc7230#section-3.5] If, for certain reasons, the previous parsing is needed use the flag `<LAX_HEADER_PARSE>true</LAX_HEADER_PARSE>` in `MuseProxy.xml`. However, it is recommended to avoid using this configuration.

5.2 Bug Fixes:

- The MuseKnowledge Proxy Setup was hanging near the end when run in silent mode with options, i.e. using

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

  Also, some options such as the SSL port were not taken into consideration from the options file. This is now fixed.

- This release contains security fixes related to corner cases for rewritten requests and their responses. Such cases are improbable to happen in modern browsers.

- Very small memory leaks in Spring SAML Security and OpenSAML libraries were detected and fixed. The issues consisted in accumulating:

  - `org.opensaml.saml2.metadata.provider.ChainingMetadataProvider$ContainedProviderObserver` after the metadata were automatically or manually refreshed.

  - `ExtendedMetadataDelegate` and `FileBackedHTTPMetadataProvider` for the metadata that cannot be obtained (e.g. URL connection cannot be achieved due to network for IDP) which also remained in memory after a Refresh Configuration.

  In production the rate of refresh is big enough so that these issues were not having a visible impact. They could have been an issue only if the refresh rate was modified to seconds level which was very unlikely.

- The internal Jetty Servlet Engine was upgraded once again to resolve a meta-space memory cleaning aspect. It seems that metaspace memory wasn't released as quite often as it happened in the past with the first version of Jetty included in MuseKnowledge Proxy.

- With a very small probability, refreshing the SAML Configuration / Restarting SSO could have triggered a deadlock and thus blocking the Jetty Servlet Engine. This could have happened if there is an automatic metadata refresh in progress (depending on the `maxRefreshDelay` values configured in `securityContext-metadata.xml`) and at the same time one Restarted Jetty (press Restart SSO) or Refreshed the SAML Configuration. If the `maxRefreshDelay` is of hours/days magnitude which is the usual in production then running into this issue was very unlikely. The issue was actually found only through a theoretical analysis. With all these, the new version is fixing this issue.

- This version fixes a race condition when both a Refresh SAML Configuration and another SAML administrative operation is run in parallel (a very unlikely scenario) resulting in the metadata ad ministrative structures not being initialized correctly (the metadata manager not being wired into the administrative controller instance). The effective usage of the SAML authentication flow of the

end-user was not affected.

- Restarting the Servlet Engine resulted in file descriptor leaks, especially for jar files of the web application. This is now fixed through configuration of static java and Jetty Jar related classes (this is a known JDK issue `https://bugs.java.com/bugdatabase/view_bug.do?bug_id=JDK-8163449`).
- Increasing the maximum header size for the Servlet Engine to `16384` bytes to cope with cookies piling up.
- The Access log and Statistics log files may not have recorded the very last requests served just before stopping MuseKnowledge Proxy. This is fixed.
- With previous versions, if, by mistake, Muse Proxy is started again while it is already running and the Servlet Engine was and still is enabled, then the SAML, SSO and Manage Applications web contexts stopped responding remaining in an inconsistent state. With the new version this is corrected by an extra check before starting.
- Fixed a corner case when "rubbish" content could have been issued if a Http relay structure was reused after a previous unexpected exception caused by rare race conditions.
- Modifying the `READ_TIMEOUT` to a smaller value than the `TARGET_READ_TIMEOUT` in `MuseProxy.xml` was affecting the SSL CONNECT Tunnels in the standard HTTP (classic) proxy usage and manifested in glitches while rendering complex pages through MuseKnowledge Proxy when used as classic HTTP proxy. This was fixed.
- In case of indirect authentication (SAML/SSO/RemoteAuth), when the last character in the resource vendor URL (`url=` / `qurl=`) or in the POST parameters (for widgets) was minus('-'), an incorrect return state was created and this could have affected locating the desired resource on the vendor platform after the authentication had taken place. This is now fixed, including the case when a TinyURL ending in '-' or with POST data ending in '-' was generated from a MuseKnwoledge Search source.
- Prevent internal `NullPointerException` for cases where local replies do not have a content at all (the case with `HEAD` for `http` -> `https` redirects, was one of them). Note that the reply was served, and there were no side effects, however an extra protection was set to keep the normal flow.
- A concurrency bug for Source Filter configuration was fixed: in case multiple requests happened in parallel and a configurable filter was involved it might not have had the proper configuration. This fix was already deployed for partners under support and maintenance policies via `mnm.jar` being available since 2020.09.15.
- For the Extract and Navigate steps using the HttpModule (which is based on JDK's `URL Connection`), only `gzip` for `Accept-Encoding` is used instead of `gzip, deflate`, because of certain deflate related issues.
- If an I/O error happened when establishing the socket connection to the remote resource, for plain `http://` cases and for classic standard proxy connection, it, sometimes, was possible to retry the request with a different IP from the proxy machine. This was fixed.
- The MuseKnowledge Proxy Setup is now better handling the file permissions on Linux systems.
- The workaround for `SameSite=None`, mainly for old Safari 10-12 browsers which do not

understand this value and treat it as `SameSite=LaX` (`SameSiteNoneRemovalFilter`) was not covering all the `ssoRWP(2) Set-Cookie` scenarios - this happened for cases where no content is written by the servlet and no sendRedirect is used; the `Set-Cookie` header was not intercepted. Such a case is for OAuth and the workaround was updated.

5.3 Recommendations:

- If you are performing an upgrade over an existing installation and the `PRIMARY_COOKIE_CONFIG` and `SECONDARY_COOKIE_CONFIG` entries from `MuseProxy.xml` were previously customized (e.g. prefix change), then the customization needs to be performed again. Check the previous file `MuseProxy.xml.bak` where the older entries are preserved.
- If you are performing an upgrade over an existing installation and the files in `${MUSE_HOME}/proxy/i18n` were customized, make a backup before the upgrade and then re-do the customization. Note that a new key, `APPLICATION_NOT_SERVED_FQDN`, has appeared and other keys have had their text modified.
- Advertise only documented entry points URL in external systems (including social media). Do not copy and paste URLs from the browser while navigating a resource through MuseKnowledge Proxy. The link itself might not even be bookmarkable even if the vendor is accessed directly. If the native link itself is bookmarkable, it is OK to advertise prefixed URLs `https://proxy.example.org/APP?qurl=https%3A%2F%2Fresource.vendor.com%3FbookID%3D123456789` or `https://proxy.example.org/APP?sourceID=VendorDbID` but a follow up, internal Rewrite by Host or Rewrite by Path URL, such as, `https://0f1060xpe-p1-y-https-resource-vendor-com.proxy.example.org/bookID=123456789`, must never be used in external systems.
- Configure `<ENFORCE_HTTPS>true</ENFORCE_HTTPS>` in all the existent applications `web.xml` file. Plain `http://` entry points URLs must no longer be advertised in external systems. Use only `https://` links when addressing to Muse Proxy.
- To have access to MuseKnowledge Administrator Console / Manage Applications section make sure that `SERVLET_ENGINE_ENABLED` is set on `true` in `MuseProxy.xml` and restart the server. Make sure that at least `8GB` of memory are available for the target machine and, in case you are upgrading, make sure that Java Virtual Machine has `PROXY_XMX/XMX` set to at least `1536M` in `configure` (for Linux), `configure.bat` / `JavaService.jvm` (for Windows). These are the minimum resource requirements.
- Safari end-users should uncheck the option Safari > Preferences > Search > Smart search field: > "Preload Top Hit in the background".
- In order to avoid browser caching delete the cache of the browser used to access the `/admin` interface before switching to the new version.

6.0

Changes in MuseKnowledge Proxy 5.2 Build 03

Release Date: 2020-05-15

6.1 New Features:

- Added SIP (Session Initiation Protocol, version 2.0) authentication module. All details about how this login module can be configured are found in the `${APPLICATION_HOME}/profiles/login/ProxyLoginModuleSIP.xml` configuration file, explained through comments. Both Telnet and TCP/IP layers are supported.
- POP authentication Login module is now available for legacy setups to authenticate users through the Post Office Protocol. More details about how this login module is configured can be found in the comments within the `${APPLICATION_HOME}/profiles/login/ProxyLoginModulePOP.xml` configuration file. It is recommended to use this module only with self-managed email services (on-premise) and not with global email providers, where SSO / SAML is actually encouraged.
- For uniformity, `LOG_USER_ID`, `LOG_USER_ID_MODE` and `USED_PARAMS` configuration entries can now be read and interpreted by more of the direct login modules: U/P, FTP, SIP, IMAP, POP.
- Due to the shutdown of GooglePlus API, for Google2Client used in SSO Google OAuth scenario, the profile URL was changed from `https://www.googleapis.com/plus/v1/people/me` to `https://www.googleapis.com/oauth2/v3/userinfo`. The Post-authentication script from the template applications login configuration, `ProxyLoginModuleSSOGoogle.xml`, was updated to work both with the JSON returned by OIDC and by People API.
- The Chrome team changed cookie cross-domain standards, introducing the requirement for `SameSite=None; Secure` attributes; other browsers will soon impose these requirements, too and Muse Proxy needs to configure the way its own cookies are set. For cross-domain integration, for example, in order to work in an iframe, or for SAML/SSO authentication with different parent domains, Muse Proxy needs to have these cookie attributes. Simply adding these cookie attributes for the existent session and sticky cookie for Muse Proxy was not a solution because Muse Proxy is not a normal web server. Most of the time it relays target (vendor) responses and some vendors are

still on plain (`http://`) and dealing with them means that the scheme of Muse Proxy is itself `http://` for which a secure session cookie will not be sent. Also, some browsers such as Safari 10-12 do not recognize the `SameSite=None` attributes and behaves even stricter. To cover these cases, a second cookie that is able to be sent on plain `http://` and to be cast without the `SameSite=None` attribute needs to be configured; this will not have the `Secure` cookie attribute, either. `PRIMARY_COOKIE_CONFIG` and `SECONDARY_COOKIE_CONFIG` are the new elements which are available in the `MuseProxy.xml` configuration file. More details about these can be found in the configuration file or in the `Muse Proxy Advanced Configuration` manual.

The internal Jetty Servlet Engine was upgraded in order to support the Chrome 80 `SameSite=None; Secure` requirement. Its `ServletEngine.xml` file name, `web.xml` Deployment Descriptor Elements for `ssoRWP` and `ssoRWP2` were updated to fit in the new Cookie flag requirements.

The End points for SAML, SSO (OAuth, LTI) and their administrative interface are now available only on `https://` to avoid any possibility of losing track of cookies and entering redirect/re-post loops. Other changed behaviour might be observed for `http://` sources.

Because this is a change in the standard and secure flags are imposed by the cross domain rules, and, in general `http://` usage is strongly discouraged by modern browsers, Muse Proxy must be configured (either directly or through SSL-Termination) with a valid wildcard CA SSL certificate and all of its entry points (such as application, or source URL) must be advertised using the secure `https://` scheme - this means that external systems must always refer to Muse Proxy using the `https://` scheme. This is something that is already in place for most of the installations, but now it becomes mandatory for all. For MuseSearch integration cases with Type 1 Links, the `Navigation Manager Host` should be configured in Muse Management Console (`/mmc`) by prepending it with `https://` scheme and using for the Port the secure one, usually, in production, 443. The only URLs that remain on plain `http://` are the rewritten links (Type2/3 - Rewrite by Path/Rewrite by Host) corresponding to the plain `http://` vendors. Muse Proxy can, theoretically, relay `http://` vendors using `https://` as a front-end but this implies a harder effort for profiling that vendor source and more processing resources because all the native components (CSS, JS, images, fonts, etc) must be rewritten to become accessible on `https://` via proxy.

First steps on implementing localization are done including support for localizing the server messages and also messages reported by configured login modules. These messages are found in the `${MUSE_HOME}/proxy/i18n` directory. The `LOCALE` element was added globally in `MuseProxy.xml` configuration file, but can also be configured at the application level in `${APLICATION_HOME}/WEB-INF/web.xml` configuration file or, for some login modules, which support the `SCRIPT` Post-authentication logic, via the special variable `locale`; there is also some support for using the `locale=` parameter for the application actions to some extent - specifying an explicit locale parameter while login means the whole session will use it, while

specifying for a certain action means direct errors will be yielded in that language (although the source navigation will use the session locale). Best efforts are made in order to provide the error messages in the localized form set by the application/login module or the locale parameter. However, there are cases in which the global locale is still used - errors during relaying/rewriting HTTP requests/responses, where the scope of the session is too inefficient to be retrieved (core errors) or the session expired errors when the locale settings cannot be identified anymore. If need be, an application can have its customized messages, for any locale, by using a corresponding file name pattern in `${MUSE_HOME}/proxy/i18n` - however, this will make the upgrade process in the future more complex. More details about localization, at this point, can be found in the comments within the configuration and localization files. Being only the basic support it will be extended in the future releases to cover interface level localization, on the fly interface and session language switching and documented accordingly.

- Muse Proxy Application expiry test logic was modified in order to benefit from the specific `web.xml` application locale and possibly use the application customized expiry key. The test is now done at the application level and not at the mapper higher level. In the unlikely case in which there are identical matches rules for two applications and the first application is expired, the second application will no longer be chosen (this was just a side effect of the initial implementation).
- The cyclic refresh is now skipping the `Shortcut Mappings` update for the expired applications.
- The following new optional entries are available for the authentication process in the `${WEB_CONTEXT_HOME}/profiles/AuthenticationGroups.xml` configuration file:
  - A Pre-authentication module is available to interpret a set of rules which ensures that the configured login modules are called only if the request is successfully validated against those rules. This can limit the number of login requests against the back-end services for deny-listed user names, because only valid requests will reach the login module. The rules are configured under the new optional `AUTHENTICATION_GROUPS/AUTHENTICATION_GROUP/AUTHENTICATIONS/PRE` configuration entry.

    This addresses mostly the login modules which connects directly to a file or a service and which explicitly receive parameters such as `userName` (e.g. IMAP, U/P File, LDAP, SQL, SIP, ExtHttp). For indirect login modules (SSO/SAML) only the IP or some header values can be used as the `userName` is not known before. Validation based on users identifiers for indirect login modules can be done, Post-authentication, via the `SCRIPT` section in their login module configuration. There are also local modules such as LDAP and SIP which allow for more complex `SCRIPT` Post-authentication validation based on other properties which are known only after authentication - Pre-authentication only applies before the login modules, based on the elements from the initial request. More details about these rules can be found in the `Muse Proxy Advanced Configuration` manual.
  - For a chain of `sufficient` login modules, an explicit sources group identifier can be set per login module using the `SOURCES_GROUP_ID` entry under `AUTHENTICATION_GROUPS/AUTHENTICATION_GROUP/AUTHENTICATIONS/AUTHENTICATION`, and this value will be used if the successfully authenticated login module doesn't provide another value for source group identifier via its own configuration module.

- Similarly, the `PROXY_HOST` and `PROXY_PORT` configurations located inside of `AUTHENTICATION_GROUPS/AUTHENTICATION_GROUP/AUTHENTICATIONS/AUTHENTICATION` can be used as default values if the successfully authenticated login module doesn't provide another values via its own configuration module.

- Changes were made in order to make sure that the Navigation Manager will relay the Server Sent Events, a standard which is now part of HTML5. However, no filtering support is currently available for this type of request. This is quite an incipient feature and sources that make use of Server Events should be proxied for demo purposes only. This capability is also dependent on the load balancer features (a tunnel timeout is also recommended for the load balancer).
- Where the `SCRIPT` Post-authentication part is available for the login modules, the console object is visible with the following methods `console.log()`, `console.warn()`, `console.error()` and `console.debug()` to log the message parameter (a single parameter is accepted) in the `MuseProxy.log` file on the corresponding Debug levels.
- In the Administrator Console, Manage Applications area, the speed of extracting details for all the configured applications was improved; loading hundreds of applications offers now a better experience, especially on multi-core CPUs installations.
- Creating authentication groups from scratch is a new feature added in Administrator Console / Manage Applications area.
- The `Add New Source` feature is available in Administrator Console / Manage Applications to provide support to create and add a new source from scratch based on the usual frequent fields. For more complex scenarios the `Raw Edit` is available after the source is added.
- Starting with this version, the `DEF` configuration entry from an existent source profile can be visually edited from Administrator Console / Manage Applications. The source profile must contain the `DEF` setting for the edit to be available.
- In Manage Applications section from the Administrator Console it is now possible to visually manage the users defined in the `${WEB_CONTEXT_HOME}/profiles/login/ProxyLoginModuleUserPassword.xml` configuration file of an application. The following actions can be done through the admin web interface: manage the file groups, add a new user, edit a user, delete selected users, import users from a CSV file and export selected users as a CSV file.
- For Applications Administration the size of `maxFormContentSize` from `ServletEngine.xml` configuration file was increased to `5000000` in order to cover cases when bigger files need to be modified on raw edit option from `Administrator Console`, `Manage Applications` section.

6.2 Bug Fixes:

- Properly handling the logout action from `MKPF` template application in certain cases - this is related to the timeout window and timeout modal which were displaying incorrect countdown and were sending more logout requests if the initially triggered logout request took too long. This is now

fixed. The behaviour in the older `MuseProxyFoundation` application template was accurate so no fix is needed.

- Used the UTF-8 encoding to sign the messages in Administrator Console - `HMAC Link Generator` . Note that the HMAC login module performed correctly as it was using the UTF-8 encoding from its inception.
- For LDAP authentication (`ProxyLoginModuleLDAP.xml`) the correct element to be involved for different user and password parameters is `USED_PARAMS` and not `USED_PARAMETERS`, hence in the unlikely case of having `USED_PARAMETERS` configured in the LDAP configuration files in applications, replace its element name with `USED_PARAMS`.
- The `READ_TIMEOUT` source configuration is properly set and propagated when `Http ModuleApache` is used for a source profile.
- There were cases when the memory entries in the `Virtual Machine` page from Administrator Console erroneously displayed `0`. This is fixed now.
- Correctly rewrite inline CSS `:url` when it contains entity escaped apostrophes such as: `&#39;`, `&#34;`, `&apos;` or `&quot;`. Such a case is for example: `<div class="Banner" style='background:url(&#39;/img/back.img&#39;);'></div>` .
- If there was an error generated in an application context, when a session was also to be created (such as providing a non-existent ?[q]url= without being authenticated to the application), and the application FreeMarker error template was erroneous by itself, or was missing, then an Unexpected error was presented and the application access was not granted. A default localized text message wrapper, containing the error, is now in place, as a fallback for this corner case.
- Navigation Manager relaying better respects its configuration entry from `hosts.xml`. The same entry also applies to the initial Extract and Navigate scenarios.
- In some sub-cases, when an existing navigation session from the same client session is reused (as for the case of navigation session expiring, but application context still available, or via `TRY_REUSE` with sessions in memory, not in JCS), it is possible that the reused navigation session is cor responding to a different application from the same client session. Although multiple proxy ap plications for the same end-user, at the same time, in the same browser is a rare case, this is fixed now.
- In the Administrator Console / Manage Applications, in case of source list pagination, after editing a source profile the interface `Back` button (not the browser's control) returns now to the cor responding source page number and not to the first page.

6.3 Recommendations:

- Configure `<ENFORCE_HTTPS>true</ENFORCE_HTTPS>` in all the existent applications `web.xml` file. Plain `http://` entry points URLs must no longer be advertised in external systems. Use only `https://` links when addressing to Muse Proxy.
- Safari end-users should uncheck the option Safari > Preferences > Search > Smart search field: > "Preload Top Hit in the background".

- In order to avoid browser caching delete the cache of the browser used to access the `/admin` interface before switching to the new version.

7.0

Changes in MuseKnowledge Proxy 5.1 Build 02

Release Date: 2019-07-25

7.1 New Features:

- URLs using directly Internationalized Domain Names (IDN) instead of their ASCII equivalent can now be identified and rewritten in source pages, if the source profile contains the new `<TRY_IDN>true</TRY_IDN>` boolean configuration entry. Note that the patterns must always be represented in ASCII Compatible Encoding.

  So far MuseKnowledge Proxy was able to rewrite the ASCII equivalent in a HTML page (such as for `http://revistadefilologiaespañola.revistas.csic.es` where inside its pages the resolved FQDN is used: `xn--revistadefilologiaespaola-uoc.revistas.csic.es`), but there can be cases where directly the IDN form is used in `href`s as for example even in the Unicode consortium page, `http://unicode.org/faq/idn.html`, where we can find the Unicode value for `href` directly in IDN: `<a href="http://öbb.at">http:// bb.at</a>&quot;</td>`

  Using `TRY_IDN` on `true`, MuseKnowledge Proxy will be able to identify directly such URLs in the source page and rewrite them by transforming them in the ASCII equivalent via Punycode algorithm resulting in an `xn--`... host. Inside the source profile always use the ASCII equivalent both for specifying URLs and patterns. Only for `FILTER`'s `FIND` rules when identifying an IDN URL directly, use its IDN form. Use online utilities for transforming from the IDN into the ASCII equivalent, or see the transformation the browser does when such an URL is requested. Note that MuseKnowledge Proxy is using the Java JDK IDN.toASCII(input, IDN.ALLOW_UNASSIGNED) method. The future versions will contain an Admin Utility for this translation.

  Because IDNs are transformed into ASCII equivalent by using extra characters (bytes) this is

increasing the size of the host, and in case of `https://` and rewrite by host, considering the extra rewrite markers, the maximum size of `63` length for the DNS label could exceed. In this case use a `HOST_MAPPING` element to map to a shorter name is necessary.

Use of this setting is recommended only when there is such an IDN URL in the vendor pages.

- Added Library Card's Barcode authentication module. More details about how this login module need to be configured can be found in the `${APPLICATION_HOME}/profiles/AuthenticationGroups.Barcode.xml` con figuration file. `NUM`, `NUMMOD10`, `CODE39` and `CODE39MOD43` are the algorithms supported at this moment for the barcode validation.
- A new section, named `Server KeyStores`, was added in the Administrator Console in order to manage the KeyStores used by Muse Proxy when it is configured to run on `https://`. This section contains a page named `Available KeyStores` where all available KeyStores from the disk are displayed and also you have the option to create a self signed Key Pair, to upload a KeyStore or to create a new one by Key Pair import. A very important option is to manage the `SSL_KEYSTORE_FILE` configuration entries from `${MUSE_HOME}/proxy/MuseProxy.xml` configuration file. These management actions can be made through `Assigned KeyStores` page. A server restart is necessary to take in account the changes made in this page.
- A new boolean configuration entry `<USE_FIRST_AUTHENTICATED_GROUP>true|false</USE_FIRST_AUTHENTICATED_GROUP>` was introduced in the `AuthenticationGroups.xml` configuration file in order to use the first authenticated `groupID` from a session if `groupID` is missing from the request. By default, if this configuration is missing, the authentication will work as before.
- Continued the work on integration with LTI standard - starting with this version a Muse Knowledge Proxy system can be certified as being LTI Compliant as an external service tool consumer by IMS Global Learning Consortium. This version can pass all necessary certification tests for LTI 1.0 standard.
- `<JS_MODE _rwpInclude="true|false" _rwpInvoke="true|false"/>` configuration entry was added in the source profile in order to make JavaScript rewriting more configurable.

  `_rwpInclude` attribute value specifies if the JavaScript header defining many `_rwp` methods will be inserted in the page or not

  `_rwpInvoke` attribute value specifies if the JavaScript content will be rewritten or not, i.e., if functions such as `open` are wrapped by their `_rwp` counterpart (e.g. `_rwpOpen`).

  Do not use this setting unless instructed by MuseGlobal Support Team. The backward compatible mode is `<JS_MODE _rwpInclude="true" _rwpInvoke="true"/>` but it does not make sense to set it this way as it just occupies the navigation session memory. Using a configuration such as `<JS_MODE _rwpInclude="false" _rwpInvoke="true"/>` will still include the header and act as if `_rwpInclude` would be `true` as otherwise an improper rewriting would result by calling undefined JS methods.

- `Usage Limits`, a new component which is activated globally and have configurations at the application level was added in order to protect the vendors from an automated process or abusive users that use the proxy services in an inappropriate way. These limits can be set in accordance with a vendor requirements based on the license agreement, for example. More details about how this component needs to be configured can be found in the `Muse Proxy Advanced Con figuration.pdf` document and in the comments from `MKPF/profiles/Limits.xml`. Enabling and involving usage limits requires more memory and puts a higher stress on the proxy server, so it is recommended to be used only for isolated cases. If lower limits are set at the application level these may create an unsuitable user experience.

  Rules for limits can only be set for applications where a `logUserID` is identified during the login process. In order to have a complete solution for this component, a new section named `Usage Limits` was added in the Administrator Console which lets the administrator of the system view all the usage limits rules defined for all applications, to delete and to export as CSV all counters which are calculated at the sampling rate and to view or/and un-suspend the suspended users(the users who have exceeded at least one user limit).

- A new boolean attribute named `expire` was added to `COOKIE_JS` configuration entry (available at global level and at source level) and if the value of this attribute is `false` then un-intercepted (unformatted) cookies will no longer be sent back in an expiring Set-Cookie for Type 2 (Rewrite by Path) responses. Default value if missing, is `true`.

- To better unify MuseSearch Type 1 URLs with a Muse Proxy application in order to leverage the power of proxy source profile configuration, we added support to specify the authentication `groupID` of the Muse Proxy application beside application root and source id which are specified in MuseSearch source profile rewriting pattern by using the `",groupID:"` prefix. The MuseKnowledge Search application must have a `modulesutil.jar` version `1.2343` or higher.

- For source `FILTER`s support to specify the domain of the proxy FQDN in a `REPLACE` expression was added. This is done by specifying a value for `domain` attribute associated with `REPLACE` con figuration entry located in a `FILTER` entry from the source profile, for example:

  ```
  <REPLACE domain="0">$0</REPLACE>
  ```

  The purpose of this configuration is to replace the whole group (e.g. `$0`) with the domain (it removes the first label + '.' from the host name) of the proxy (in case of multi-tenancy the host that started the current navigation session is used). If Muse Proxy is accessed by IP then the same IP will result. But Muse Knowledge Proxy should not be accessed by IP. It must always have a name. Also if a short name (e.g. only `museknowledge`) is used then this will be the value for domain.

- Type 1 URL's originating from a MuseKnowledge Search application are signed with a HMAC signature, configurable in `MuseProxyAuthenticationToken.xml` file, via `SIGN`, `KEY` and `ALGORITHM` new elements. The MuseKnowledge Search application(s) that connects to this new version of Muse Knowledge Proxy must be updated to use modulesutil.jar version 1.2338 or higher. Other improvements on security for the rewritten URLs were also considered. Set the `KEY` value after the installation.

- For SAML authentication, the signing and digest method algorithms to be used in signing SAML messages and metadata can now be specified per tenant not just globally. In order to set a particular digest use the new `digestMethodAlgorithm` property of the corresponding `Exten`

`dedMetadata` bean. For example:

```
<property name="digestMethodAlgorithm"
value="http://www.w3.org/2001/04/xmlenc#sha256"/>
```

This property is automatically generated from Metadata Admin when filling in "Digest method algorithm:" field under "Advanced Settings" by using one of:

- `http://www.w3.org/2000/09/xmldsig#sha1`
- `http://www.w3.org/2001/04/xmlenc#sha256`
- `http://www.w3.org/2001/04/xmldsig-more#sha384`
- `http://www.w3.org/2001/04/xmlenc#sha512`

Note the different namespaces for `SHA1` and `SHA384`.

In order to set a specific signature algorithm use the already existent property `signingAlgorithm` of the same `ExtendedMetadata` bean (or the field Signing algorithm). Now this property is also used for signing the SAML messages and not just the metadata.

- Added support to specify the source IP for LDAP authentication for cases where this is important. Sites which perform LDAP authentication may limit incoming LDAP queries to known hosts. As part of this network programming extension, configurable timeouts were also introduced for the LDAP login module.

  The source IP is specified with the help of two new configurations `PROXY_USED`, used to specify from where to take the value of the binding IP (supported values are `SOURCE_LEVEL` and `APPLICATION_LEVEL`), and `PROXY_HOST`, which is used to specify the value for the bind address if the value set for `PROXY_USED` is set to `SOURCE_LEVEL`. It is recommended to use this setting only if the LDAP server limits by IP.

  Timeouts are specified in milliseconds through `CONNECT_TIMEOUT` and `READ_TIMEOUT` configuration entries from the LDAP login module configuration file.

- Added a new boolean configuration entry `<DISABLE_TARGET_KEEP_ALIVE>true|false</DISABLE_TARGET_KEEP_ALIVE>` in the source profile to disable the target keep alive in order to make sure that the TCP/IP connections are not reused for requests made for that source. This is needed for a very limited number of sources which behave unpredictable in case the same TCP/IP connection is reused multiple times. Disabling target keep alive as a global option was possible, but not optimal because to make only one source work the performance of all the rest suffered, so this option is now available per a source basis.
- The unauthorized LTI errors are including more details in the `ext.log` and are visible in the interface for the administrator (i.e. if also logged on into the MuseKnowledge Proxy Admin console).
- Because the newer versions of web browsers are implementing Intelligent Tracking Prevention

(3rd party cookies) and this option is enabled by default, this affects our integration with LMS systems as embedding resource through LTI standard. More changes were performed in order to run in an iframe within the LMS systems (Canvas, for example).

- In Manage Applications section from Administrator Console support to change the order of login modules from `Edit Authentication Groups` page with just a few clicks was introduced; until now changing the order was achieved only through raw edit option.
- To handle Unsolictied SAML Responses without a `RelayState` parameter, there is now support in `securityContext-metadata.xml` configuration file to specify the mappings between an entity target URL's and a target where the user will be redirected after a successful authentication if no relay state is found in request. These mappings are set under `entityTargetUrl` bean.
- The `logUserID` information was made available in the detailed page of the client session from Administrator Console for each context and also through JMX.
- `TRY_REUSE`, a new very complex source configuration item was introduced. Starting from minimizing the impact made by loading multiple cover images in MuseKnowledge Search through Muse Proxy to avoid creating a navigation session for each requested image inside the same client session, a new boolean configuration entry `<TRY_REUSE urlMustBePresent="true|false" paramsMustBeAbsent="true|false" type1MustBeUsed="true|false">true|false</TRY_REUSE>` was added in the source profile. It should be used only if instructed so by the MuseGlobal Support Team.
- Another complex option introduced is `<SKIP_LAST_NAVIGATE>true|false</SKIP_LAST_NAVIGATE>` - a boolean source con figuration entry located in the source profile to indicate that this source will not perform the last request against the native target's configured URLs (including the received one in url/qurl) before providing control to the browser. So the last URL (or the URL if it is just only one configured) will just be rewritten to its corresponding Type 2 or 3 and then handed over to the browser via redirect. The request to the native target will be done only after the browser request the Type2/3 form. As most of the sources use only one URL element it means that last == first, so no request is done initially (via extract and navigate scenario) with content saved in the navigation session. This flag should not be set on true if the last request is a `POST` or if `REFERER` is needed to be taken into account for that request for some kind of source authorization. It should be used only if instructed so by the MuseGlobal Support Team.
- A new source boolean configuration entry `<PRL_MODE linkOut="true|false" kee pUnmatched="true|false"/>` is possible to handle rare cases involving `//` URLs. The role of this configuration is that locally unmatched Protocol Relative URL's to be matched against all the application sources (if `LINK_OUT` is true) and if not found to be left as it is with `//`. If `LINK_OUT` is false and the Protocol Relative URL doesn't match it will be left as it was starting with `//` in case `keepUnmatched="true"`. It should be used only if instructed so by the MuseGlobal Support Team.
- The `<REFLECT_ORIGIN_NULL>true|false</REFLECT_ORIGIN_NULL>` boolean con figuration entry was added in the source profile to indicate that if the `Access-Control-Allow-Credentials` header is missing from response and the Origin header from request is "null", then we set `Access-Control-Allow-Credentials` header to "null". It should be used only if instructed so by the MuseGlobal Support Team.
- The protocol extracted from `X-Forwarded-Proto` header is used only when the value of con

figuration entry `<TRUST_X_FORWARDED_PROTO>` defined in MuseProxy.xml configuration file is set to `true`. By default, if missing, this flag is `true` and for single instance (non balanced en vironments) or load balanced environments using HAProxy PROXY Protocol it can be set to `false`. Also the global configuration `REMOVE_XFF`, `REMOVE_XFP` were introduced, more as a backward compatibility measure, both defaulting to `true`, in order to not keep the inbound `X-Forwarded-Proto` and `X-Forwarded-For` headers when making the request to the vendor sources. Normally `REMOVE_XFF` and `REMOVE_XFP` should not be set at all in the configuration file.

- An advanced configuration element can be specified per source level `<ON_URL_MATCH_FAIL>REDIRECT|ERROR|REDIRECT_IF_HOST_MATCH</ON_URL_MATCH_FAIL>` to control what to do when on a live navigation session an URL that is not matching the patterns is received. The global configuration defined in `NavigationSession.xml` con figuration should generally be enough. This option is to be used when higher protection is enforced via configuring `REDIRECT_IF_HOST_MATCH` in the global setting in `Naviga tionSession.xml`, but some sources are not working properly with this setting. Do not set it otherwise.

- The SAML authentication error page is now displaying details (except for the stack trace) and, in case of the `InResponseToField` protection error, the page is displaying guidelines on how to avoid this. Also, in case SAML `AuthNResponse StatusCode` is not `Success` some heuristics is involved to output hints in the error message and in the log file.

- Custom Single Sign on Login integration (Remote Auth) is possible starting with this release. It involves a Remote Authentication CGI script where proxy will redirect the request to validate credentials if a new session is necessary and then the script returns back to proxy via HMAC. If there is already an existent session (the end-user is already authenticated) in the server where the remote script resides then the script should not ask again for the credentials but redirect to the proxy with the HMAC signature. The script should construct the redirecting URL by pre-pending the scheme (`http/https`), the proxy host (eventually proxy port if not `80/443`) to the `_rwpReturn` parameter value, and then appending the HMAC parameters. For example,

  ```
  "https://proxy.example.com" + _rwpReturn + "&userName=" + userName +
  "&ts=" + timestamp + "&sig=" + digest
  ```

  Where `_rwpReturn` is the value of `_rwpReturn` parameter, `userName` is the userName from the remote web application session and `timestamp` and `digest` are calculated according to the HMAC authentication for Muse Proxy. The `_rwpAppId` parameter value can be used to select the necessary HMAC details (such as the secret) and eventually to select the proxy host, in case the same script would be used for multiple applications with multiple proxies. See `Authentica tionGroups.RemoteAuth.xml` and `ProxyLoginModuleRemoteAuth.xml` for more details.

- Experimental SAML configuration to use RelayState authentication, for all tenants, with the exception of some not supporting it or supporting only 80 bytes for the `RelayState`. In the bean `SAMLEntryPointRelayState` the `excludeRelayState` property is to be used for this exclusion, together with the one defined in `securityContext-metadata.xml`, for the SPs whose corresponding IdPs have limitation with regard to the `RelayState`. For these rare cases the HTTP Session is used for storing the initial proxy request. Note that, in this case, there is still no Concurrency warning and no `InResponseTo` field validation, so in case of concurrent usage the `/ssoRWP` page will be obtained for all but one concurrent requests and only one will, of

course, work.

- If MuseKnowledge Search integration via Type1 URLs is used, the recommended value for `ON_EXPIRY` in the file `NavigationSession.xml` is `LOGON_REDIRECT_ACTIVE_ANONYMOUS`.

7.2 Bug Fixes:

- Fixed a concurrency issue for the applications refresh function. There existed a small probability for skipping new applications or new changes in the web context and applications index file until the next change or until the next restart.
- The logic for providing the `MuseProxyID` cookie, if missing from the request, is now covering the direct requests for `/ssoRWP*` in order to ensure LTI and IdP initiated SSO are working correctly in a Load Balanced environment. The `ID` element from `MuseProxy.xml` must be used and configured with `cookie="true"` and `cookieSuffix="true"` attributes, for example:

```
<ID cookieSuffix="true" cookie="true">p1</ID>
```

- Fixed the following bug for the Administrator Console which was never opened in a browser so far (or cache, cookies and site data have been cleared meanwhile): login into the Administrator Console, and then going to Configuration/SAML Authentication pressing on Metadata, and then accessing the Manage Applications section brought the web session to an end, the administrator being logged out.
- The new style MKPF application template did not contain the `PRE_PROCESSING` configuration entry under `FILE_SETS` and comments for `authenticationFlow` in the web.xml con figuration file. These were now added.
- For MuseKnwoledge Search link integration, updating of the `modulesutil.jar` could have broke the next rewriting requests in that existing search session for `https://` URLs – this is fixed now.
- Multiple levels of URL encoding were required for Canvas when MuseKnowledge Proxy is embedded as external tool.
- The HMAC authentication is now using case insensitive comparison of hex hashes.
- Fixed a bug related to hidden sources in the alphabetical area – if two sources were defined one after another and both with hide attribute set to `true`, then the second one wasn't hidden.
- Solved the case when a URL doesn't have a path but is just followed by quote which is not matched by our JSON filter when Rewrite by Path is used.
- SAML metadata loading from remote secure URLs (`https://`) upon server restart / refresh con figuration could have blocked if SSL Handshaking blocked. The request timeout value specified in the constructor argument of the `FileBackedHTTPMetadataProvider` is now used not only for the connect timeout but also for the SSL Handshaking process itself:

```
<bean
class="org.opensaml.saml2.metadata.provider.FileBackedHTTPMetadataProv
```

```
ider">
    <constructor-arg type="java.lang.String"
value="https://ipdhost.example.com/idp-meta.xml"/>
    <constructor-arg>
        <value type="int">20000</value>
    </constructor-arg>
    <constructor-arg type="java.lang.String"
value="${MUSE_HOME}/proxy/tmp/ipdhostexamplecomidpmetaxml_idp.xml"/>
...
```

- Saving XML files from MuseKnowledge Proxy Administrator Console or via JMX is more reliable - due to encoding aspects, the backup file, to be automatically reverted to, in case of failure, for certain edit operations, was not right. Now this is fixed.
- The handling of `%b/%B` in the `access.log` format now covers the cases when there is no bytes transferred in the response (`-/0` was missing) and also the size for the WebSocket responses (this is written after the WebSocket closes).
- Fixed a race condition bug for TCP/IP connections re-usage.
- Fixing the rare case of Link Out sources linking out to another source after the authentication to the application context expired meanwhile.
- Fix the SAML/SSO authentication when the proxy applications are defined with IP rules and the same path, but the `URL_PATTERN` is not containing the host that would differentiate them.
- Fixing a race condition for cookies set concurrently in the same navigation session when in terpreting the responses from the target source.
- Fixing application admin templates for HMAC, Referer and External login modules.
- Under very high load there existed a small chance to obtain an incomplete or empty HTTP body response. This is now fixed.
- The cache mechanism was ignoring the size limits of a response. Now, the limits are taken into account when the reply has `Content-Length`.
- The cache mechanism was skipping one response header field when storing the response (the last one is given by the internal Java hash key order). This is now fixed.

7.3 Recommendations:

- Safari end-users should uncheck the option Safari > Preferences > Search > Smart search field: > "Preload Top Hit in the background".
- End-users should not block third party cookies or cookies from unvisited web sites in their browsers. Otherwise various integrations such as LTI in iframes will not be smooth.
- The values of the elements `REFRESH_INTERVAL` and `CLEANUP_INTERVAL` from `MuseProxy.xml` configuration file should not be lowered.
- In order to avoid browser caching delete the cache of the browser used to access the `/admin` interface before switching to the new version.

8.0

Changes in MuseKnowledge Proxy 5.0 Build 04

Release Date: 2018-06-21

8.1 New Features:

- A new proxy application template, `MKPF`, with a new look and feel is available starting with this version. `MuseProxyFoundation`, the old one is still maintained and available with all the features up to date. On a fresh install `Anonymous`, `MuseProxyFoundation` and `MKPF` will all be installed and available, while on an update `MKPF` will be installed but not linked to in the `Applications.xml` file to avoid compatibility issues with another application having the same code and to protect installations with Small or Medium licensing which already have 4 o 8 applications. So the administrators which are upgrading MuseKnowledge Proxy directly via the setup and want to discover the new template will have to edit Applications.xml and add this entry under the `APPLICATIONS` section, taking care to set an integer value to the `code` attribute.

```
<APPLICATION>
<IDENTIFIER code="${next available value}">MKPF</IDENTIFIER>
<CLASS>com.edulib.muse.proxy.handler.web.context.application.WebContextApplication</CLASS>
<DIRECTORY>${MUSE_HOME}/proxy/webcontexts/Applications/MKPF</DIRECTORY>
</APPLICATION>
```

  For the Small Organization installation upgrades we recommend replacing the `MuseProxyFoundation` or `Anonymous` entry.

- The Administrator Console is now based on a new theme and a new component is available to help the administrator of MuseKnowledge Proxy to manage and maintain the applications and sources. This section can be accessed from the Applications menu via the `Manage Applications` link as long as `SERVLET_ENGINE_ENABLED` is set on `true`. The servlet engine is not active by default because it needs more memory and disk resources and very small installations not needing SAML/SSO or visual administration should still work if memory is an issue. There are visual actions for the common elements and raw view/update actions, where directly the XML files behind are edited in a simple JavaScript editor ensuring XML well-formedness. An application can be edited, copied, exported, imported, backed up, restored, checked or deleted. A source can have its profile edited, backed up and restored, while the other attributes from the `Sources.xml` file related to a source, such as image, visibility, module, attributes and parameters can be also

managed. Source grouping including defining new visual areas and categories is also possible. Au thentication to an application can be managed, details for the login module set and their con figuration files can be edited in a raw XML form. The Manage Applications tools work with the last MKPF and old-style MuseProxyFoundation provided with the current setup, while for the existent applications not all the functions will be active.

- The new Administrator Console offers a smoother SAML Metadata administration under the same browser tab, and this interface was also updated to the new style.
- Starting with this version, MuseKnowledge Proxy supports relaying and filtering the WebSocket Protocol. There are more aspects to consider: 1) the rewrite of the initial address provided to the JS WebSocket constructor (the server end-point) in case this is not relative, 2) setting the filtering and frame serialization mode and 3) the effective filters. It is recommended that filtering is performed only if the frames are containing URLs that need rewriting, otherwise just relay the WebSocket transparently. Also it may not even be the case to do any relay if the process functions with the WebSocket end-point unrewritten.

  In order to rewrite the initial WebSocket call, if it was decided that we need to relay/filter and the ws(s) URI is not relatively constructed, a filter of the following type may work as long as the value of the WebSocket constructor is given inline:

```
<FILTER>
  <PATTERNS>the page where WS is created – be as exact as
possible</PATTERNS>
  <FIND>\("(ws(s)?://[^"]+)"\)</FIND>
  <REPLACE rewrite="1">("$1")</REPLACE>
</FILTER>
```

  To control WebSocket filtering and frame serialization mode the `WEBSOCKET` configuration element was introduced. Its main attribute is `applyFilters="false/true"`, while it supports other attributes for unusual cases.

```
<WEBSOCKET applyFilters="true"/>
```

  Some of its attributes are described bellow:

  - `applyFilters` attribute is a boolean one. If it is configured with the `true` value, then the source Filters will be invoked for WebSocket responses coming from the server. Applying filters on onward requests from client to server is also possible as long as a `FILTER` is configured with the attribute `forWSRequest="true"`. If `applyFilters="false"` or missing, then filters are not applied and WebSocket frames are passed transparently, not being assembled and disassembled to/from messages.
  - The `extendOnRequest` and `extendOnResponse` attributes will be used only if the value

of `applyFilters` attribute is `true` and the purpose of these is to extend the navigation session with each frame transferred, because an application may, from some point, work only based on socket traffic, and not standard HTTP request/responses and only a normal HTTP request/response is resetting the timeout counter.

- The `frameLimit` used if the value of `applyFilters` attribute is `true` and represent the maximum size of one frame expressed in bytes. If no value has been assigned, then the default value used is `32768`. The messages in both directions are split in frames based on this `frameLimit`. Rewriting increases the message length and sending the whole message in a single frame may not be accepted by the server.

The effective filters applied on the frame, if the `applyFilters` above is `true`, are chosen if their `PATTERNS` values match the initial `ws(s)://...` address. The text messages passing through the websocket can be of any type. If they are JSON, then a `FILTER` with `id="JSON"` can be applied. STOMP (Simple Text Orientated Messaging Protocol) messages are more complex and `content-length` should be removed, because rewriting modifies the length, for example using such a `FIND/REPLACE` filter rule:

```
<FIND>\\ncontent-length:[0-9]+\\n</FIND>
<REPLACE>\\n</REPLACE>
```

- OLSA support is available for proxy to platform authentication. Open Learning Services Ar chitecture (OLSA) is a comprehensive service oriented architecture initiative that is intended to simplify the effort required to integrate SkillSoft learning services with your Learner Management System(LMS) or portal of choice. A MuseKnowledge Proxy source can be integrated with `OLSA API` to create the user dynamically and perform the signon. This is based on SOAP requests which are made transparently when the configured source with this integration is accessed by a MuseKnowledge Proxy user. If all the configurations related to the organization are done, with this new feature a user of MuseKnowledge Proxy application can jump into the Skillsoft Skillport 8i platform as an authenticated user of this platform without the need for continuous content proxy rewriting.
- In order to reliably support third party libraries for various proxy to platform integrations, a new `CLASSPATH` structure to create ClassLoaders was added in `MuseProxy.xml` configuration file. The value of this configuration entry 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). For example the following entry is used for the OLSA based integration:

```
<CLASSPATH id="olsa">
  ${MUSE_HOME}/proxy/ext/OLSA/;
  ${MUSE_HOME}/proxy/ext/OLSA/classes/;
  ${MUSE_HOME}/proxy/ext/OLSA/classes/config/;
```

```
  ${MUSE_HOME}/proxy/ext/OLSA/lib;
</CLASSPATH>
```

The `MuseProxy.xml` configuration file may have several such structures and the ids of each entry can then be referred in the source profile via the `ref` attribute of the CLASSPATH element:

```
<CLASSPATH ref="olsa"/>
```

Also the classpath ids must not be duplicated. A refresh action was added in Administrator Console under `Advanced` menu – `Operations` page, where available ClassLoaders can be refreshed.

- If an application was configured with the LDAP Authentication module (`ProxyLoginModuleLDAP`) then from the HTML form use the same logon parameter names as in case of `ProxyLoginModuleUserPassword`. The parameters names to be used can be also mentioned in the LDAP configuration file using the new `USED_PARAMETERS` with positional `PARAM` elements – the first one for the user name, the second for the password.
- The LDAP Authentication module (`ProxyLoginModuleLDAP`) can now be configured with a script to take final decisions based on attributes and membership. Methods `isMember(String dn)` and `isAttribute(String name, String valueToMatch[, boolean isRegex])` can be invoked and authentication can be refused or a certain sources group can be selected. The adjacent `setWithUser(boolean flag)`, `setMemberUser(boolean flag)`, `setAnyAttributeValue(boolean flag)` and `setMemberAttribute(String attrName)` methods controls how the `isMember(..)` and `isAttribute(..)` operates. More details can be found in the comments from `ProxyLoginModuleLDAP.xml` and in the manuals.
- In case there are more LDAP servers to be confronted for a single application authentication then, for such a scenario, configuring multiple `USER_GROUP` entries in the `ProxyLoginModuleLDAP.xml` (one for each server) and linking to these setting `LEVEL` on `sufficient` from the `AuthenticationGroups.xml` file is possible.
- Referral configuration for LDAP authentication is possible via the new `REFERRAL` element with either `ignore`, `follow` or `stop` values. By default the referrals are ignored. However if there is an instance of LDAP server using referral then set the flag to follow. Note that the same credentials are applied and the host name used in the referral URI must be resolvable. All the normal entries are processed first, before following the continuation references. However a "referral" error response is processed immediately. When this setting is on `stop`, then when a continuation reference or "referral" error response happens the process stops.

Even with the referral setting, in case of Active Directory the Global Catalog (port `3268/3269`) should be used instead of individual instances (port `389/636`). This is because the referrals themselves often contain LDAP server hostnames which differ from the original Active Directory LDAP server (domain controller) hostname. These hostnames must be resolvable via DNS in order for successful resolution, but sometimes they are not. The referral service port must also be reachable via the network through firewalls, etc.

- New attribute flags for TLS related settings are available for the `LDAP_URL` in case of LDAP au thentication. These are `startTls="false/true"` and `sslTrustAll="false/true"`, ensuring a wider integration of Muse Knowledge Proxy with LDAP servers.
- Proxy Host and Port can be set during SAML, LDAP and SSO login depending on the au thentication properties, for example based on a group property coming from SAML we can choose a certain source IP for outbound. The sources will have to be configured using `PROXY_USED` set on `LOGIN_LEVEL`. The Login Level script section will have to set `proxyHost` and `proxyPort` accordingly. For outbound IP which is assigmed to the proxy machine itself the port must not be specified.
- Starting with this version the `SAML Configuration` can be refreshed without restarting the entire Servlet Engine. In the Administrator Console under `Configuration` menu - `SAML Au thentication` page identify the `Refresh SAML Configuration` button and click it to refresh the SAML Configuration after performing an edit to the `securityContext-metadata.xml`, adding a new keypair or a new SP/IDP metadata. Reloading the SAML related beans and context required a different organization of some beans in order to be reloaded and to avoid memory leaks and that is why in case of an upgrade the instructions displayed at the end of the setup (or in `${MUSE_HOME}/Upgrades.txt`) must be followed. Most of them are related to the file `${MUSE_HOME}/proxy/webcontexts/ssoRWP/WEB-INF/securityContext-metadata.xml`.
- The `SSO Configuration` can be also refreshed without restarting the entire Servlet Engine. In the Administrator Console under `Configuration` menu - `SSO Authentication` page identify the `Refresh SSO Configuration` button and click it to refresh the SSO Con figuration. As in the case of SAML, for an upgrade there are certain modifications imposed related to the Spring configuration elements. For example `applicationContext.xml` and `securityContext.xml` must be moved one level up. Also check that in `${MUSE_HOME}/proxy/webcontexts/ssoRWP2/WEB-INF/web.xml` the param-value corresponding to `contextConfigLocation` is now `/WEB-INF/applicationContext.xml,/WEB-INF/securityContext.xml`.
- The Apache JCS component used for the mixed memory-disk navigation system storage (when `NAVIGATION_SESSION_STORAGE` has the value `JCS`) was upgraded to version 2.2.
- In addition to the ECMA script processing via the Oracle Nashorn engine, the reference process in a source profile can now be done via `BeanShell` scripting engine. This is lighter with regard to the metaspace memory consumption because it is not using `LambdaForms` but classic reflection calls. This can be configured in a source profile by providing `beanshell` as the value for the new `type` attribute of the `DEF` configuration entry, for example

  ```
  <DEF src="MyScript.bsh" type="beanshell"/>
  ```

- From within the ECMA(JS)/Beanshell script used within a source profile for processing a console object is available to log messages in the `MuseProxy.log` file on various levels: `console.log()` writes on `NOTICE`, `console.warn` writes on `WARNING` , `console.error` writes on `ERROR` and `console.debug` writes on `DEBUG` level.
- Besides ECMA and Beanshell, Java classes located in the `CLASSPATH` configured for the source can be invoked by using the `"java:"` prefix in the value of `process` attribute, for example:

```
<PARAMETER process="java:example.GetOlsaUrl">
```

- The integration within a Learning Management System (LMS) was extended and starting with this version, a MuseKnowledge Proxy application can be integrated as `Rich Content Editor` in a Canvas LMS. Note that only the latest `MuseProxyFoundation` and `MKPF` applications can be integrated. Detailed guidelines for this integration are available in `Configuration` menu - `SSO Authentication` page from Administrator Console by accessing the `LTI Guidelines`.
- `excludeLocal` is a new pattern option with the scope to exclude from main source(Link Out) but rewrite with others if matching. This option can be used within `include` and `exclude` pattern options from `REWRITTING_PATTERNS` configuration entry from the source profile.
- When accessing an expired application a different HTTP status can be presented instead of Not Found. This is configured using the new element `EXPIRY_ERROR` inside the `Applications.xml` file

  ```
  <EXPIRY_ERROR status="410">true</EXPIRY_ERROR>
  ```

  For an upgrade check if this element was automatically added in the existent `Applications.xml` file by the setup.
- If the default `HttpModule` source module implementation is used, in case an empty body HTTP error status is received from the server with the first request, it is now forwarded to the browser - this was previously done only when a body was present, too.
- A new attribute `method` was added for `URL` configuration entry from the source profile to specify which HTTP method to be used for the request done - each URL part of an extract and navigate scenario can have its own `method` attribute (if empty `GET` or `POST` is assumed). One of the following values can be used: `GET`, `HEAD`, `POST`, `PUT`, `DELETE`, `CONNECT`, `OPTIONS`, `TRACE` or `PATCH`. The following configuration entry represents an example of usage for this attribute:

  ```
  <URL method="PATCH">http://httpbin.org/patch</URL>
  ```

- The `HEADER` element is a new configuration item in the source profile aiming to set HTTP headers for each preliminary request. This option may appear several times under one `URL` configuration entry. This element has two attributes: `name` which represents the name of the header and `value` representing its value which will be set and this can contain variables captured by an extraction group or configured in the profile. Because the `URL` configuration entry may occur several times, only the `HEADER` elements located under the first `URL` entry will be send at the first request and the `HEADER` configurations located under the next `URL` will be send at next request. The following entries represent an example of usage for this configuration entry:

  ```
  <URL method="POST">http://domain.com</URL>
  <HEADER name="Content-Type" value="application/json"/>
  <HEADER name="Accept" value="application/json"/>
  <POST_PARAMETERS>...</POST_PARAMETERS>
  <EXTRACTOR ref="token">...</EXTRACTOR>
  <URL>http://domain.com/home.html</URL>
  <HEADER name="Content-Type" value="text/html"/>
  ```

```
<HEADER name="Accept" value="text/html"/>
<HEADER name="Some-Token" value="${token_1}"/>
```

- An option to refresh java policy files was added in Administrator Console and can be found under `Advanced` menu - `Operations` page. Java Policy Files which will be refreshed by this action are `${MUSE_HOME}/java.policy` and `${MUSE_HOME}/jaas.policy`. An error message will be displayed in case a file is malformed pointing to the erroneous line.
- The backend Jetty and all of the web applications running inside are configured with an extra layer of security.
- Because some browsers, such Microsoft Edge, don't send the cookies in a request for the favicon image, the response is a redirect to the login page and because the login process is initiated it adds unnecessary pressure on the proxy in case of SAML/SSO authentication. To avoid this favicons for Type 3 are transformed in absolute links and left unrewritten.
- Error templates in all the contexts are now neutral, without any branding and without any external resources to CSS, JS or images. They resemble the simple browser style used to report errors.
- For source profiling a new boolean flag, `DYN_HOST_MAPPING` is available. It indicates that host mapping to a shorter generated value will be used on the fly if the https version of the host label exceeds `63` characters (even if the host is requested on plain `http://` as later it may change on `https://`). Similar to `HOST_MAPPING`, for this option to take effect it is required to use `Utilities` / `Evaluate Shortcut URL` / `Update Mappings` or wait at most the `MuseProxy.xml`'s `REFRESH_INTERVAL` period (default `5` minutes). `HOST_MAPPING` have priority over `DYN_HOST_MAPPING` when it is decided if a host will be mapped.
- A grace time for the links encoded via `rwpState` during the SAML/SSO authentication was added in `${MUSE_HOME}/proxy/webcontexts/Services/profiles/TinyURL.xml`. After being consumed the links are not expired immediately after their first usage, but rather after the specified number of milliseconds within the element `RWP_STATE_LINGER`. This ensures that the URL with `rwpState` in it, that was saved before the login flow starts and which encodes the initial request to Muse Proxy, can now survive more after being consumed for the first time. In turn, this ensures that, if somehow the back button is pressed on some browsers, or the Load Balancer times out and the user press enter/refresh on the URL in the bar, that link will still work.

8.2 Bug Fixes:

- The `Cookie` header could be duplicated in the next URL requests from the extract and navigate scenario in case of using `HTTPModuleApache` as a source module. This is now fixed.
- While performing the initial source request, an invalid `Set-Cookie` header field (for example no pair just `Set-Cookie: HttpOnly;Secure`) is no longer triggering the failure of the entire source.
- Restarting the Servlet Engine used for SSO and SAML can lead to Thread and Classloader leak due to the usage of `java.util.Timer` for each metadata. A configuration change to use a single

Timer, which is canceled when the server is destroyed, resolves this. For an upgrade, the file `securityContext-metadata.xml` must be updated to replace

```
<constructor-arg>
  <bean class="java.util.Timer"/>
</constructor-arg>
```

with

```
<constructor-arg ref="mtdTimer"/>
```

- Improved the relaying of binary responses represented as chunked transfer encoding streaming.
- Correctly escape the cookies inserted as the content of `_rwpSessionCookies` - also backslashes and slashes, besides quote and apostrophe.
- Fixed a concurrency bug when deleting `Connections` from the Administrator Console in case of heavy usage.
- Fixed cache issues when an outbound source IP is used for a source and when no `BINDADDRESS` is specified.
- HTTP requests with `HEAD` method containing the `Content-Length` header are now correctly working.
- The `REMOVE_COOKIES` configuration from `MuseProxy.xml` worked only if there was one cookie; starting with this version this is fixed and multiple cookies are now correctly removed. This item is needed because some load balancers adds security cookies but never deletes them.
- Changed the algorithm which parses html pages to skip `@import` rule from div elements because it is no more used by the CSS Style Attributes standard and usage of CPU could have increased on big pages.
- `DEF src` and `scripting` attribute values were not read if `DEF`'s content was blank - this is fixed.

9.0

Changes in MuseKnowledge Proxy 4.5 Build 03

Release Date: 2017-10-05

9.1 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 `AuthenticationGroups.LTI.xml` file which just needs to be renamed into `AuthenticationGroups.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 `ProxyLoginModuleToken` 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 '`authenticationFlow`' 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.

9.2 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.

10.0

Changes in MuseKnowledge Proxy 4.4 Build 02

Release Date: 2017-05-31

10.1 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 `${name}` syntax and its value will be resolved to the content of this node, exactly as if it were defined inside the source profile.
- The Single Sign-on Authentication (other than SAML) core was upgraded and part of the new features Central Authentication Service (CAS) is also supported. Upgrades instructions related to how to handle the changes in `securityContext.xml` are provided via the setup.
- Keep up with SHA1 deprecation.
- To avoid DNS limitation of 63 bytes per label in case of proxing `https://` hosts via Rewrite by Host technique using `https://` proxy URLs, there is now the possibility to map the very long FQDN to shorter names by using `HOST_MAPPING` elements in the source profile.
- This release contains an improvement for Multi tenant environments using the same host name for all tenants but individual IPs for each one. A chaining request to proxy itself on the different IP is no longer taking place for each request, rather the outbound IP is set directly to the one of the chaining proxy. To achieve this only the `PROXY_HOST` must be configured to the allocated IP in the source profile or at the application level and the `PROXY_PORT` must remain empty.
- The generation of the Client Session cookie values was improved.
- The generation of the Connection ID values was also improved.
- Implement an improvement when `action=source` is called, and data is read from the `Sources.xml` application file.
- Refine the errors `"Unexpected exception while accessing target source."` to contain more details about what was wrong with the access.
- Added a limit for the size of multipart `POST` requests which will be kept in memory and requests which will exceed this limit will be temporary saved on the file system. The value of the limit is configured in `USE_MULTIPART_TMP_AFTER` tag from `${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.

10.2 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=`).

11.0

Changes in MuseKnowledge Proxy 4.3 Build 02

Release Date: 2016-11-24

11.1 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 au thentication 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 au thorization 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.

11.2 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.

12.0

Changes in MuseKnowledge Proxy 4.2 Build 02

Release Date: 2016-06-10

12.1 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: unrecognized_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.

12.2 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.

13.0

Changes in Muse Proxy 4.1 Build 01

Release Date: 2015-06-30

13.1 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 or ganization 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, au thentication 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 con figurations 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`.

13.2 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>(&amp;(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.

14.0

Changes in Muse Proxy 4.0 Build 02

Release Date: 2014-12-22

14.1 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.

15.0

Changes in Muse Proxy 4.0 Build 01

Release Date: 2014-12-19

15.1 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=source&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.

15.2 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.

16.0

Changes in Muse Proxy 3.1 Build 02

Release Date: 2014-04-17

16.1 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.

16.2 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.

17.0

Changes in Muse Proxy 3.1 Build 01

Release Date: 2013-09-27

17.1 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.

17.2 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.

18.0

Changes in Muse Proxy 3.0 Build 04

Release Date: 2013-08-14

18.1 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.

19.0

Changes in Muse Proxy 3.0 Build 03

Release Date: 2013-08-08

19.1 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.

19.2 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.

19.3 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.

20.0

Changes in Muse Proxy 3.0 Build 01

Release Date: 2013-06-21

20.1 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.

20.2 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.

20.3 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.

21.0

Changes in Muse Proxy 2.6 Build 20

Release Date: 2013-03-11

21.1 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.

21.2 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.

22.0

Changes in Muse Proxy 2.6 Build 10

Release Date: 2012-08-24

22.1 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.

22.2 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.

23.0

Changes in Muse Proxy 2.5 Build 09

Release Date: 2011-08-18

23.1 New Features:

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

23.2 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.

24.0

Changes in Muse Proxy Server 2.5 Build 06

Release Date: 2011-06-16

24.1 Bug Fixes:

25.0

Changes in Muse Proxy Server 2.5 Build 05

Release Date: 2011-05-26

25.1 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.

26.0

Changes in Muse Proxy 2.5 Build 04

Release Date: 2011-03-31

26.1 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.

27.0

Changes in Muse Proxy Server 2.5 Build 03

Release Date: 2011-02-22

27.1 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.

27.2 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.

28.0

Changes in Muse Proxy 2.5 Build 00

Release Date: 2010-10-07

28.1 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.

28.2 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.

29.0

Changes in Muse Proxy Server 2.4 Build 09

Release Date: 2010-09-01

29.1 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.

29.2 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.

30.0

Changes in Muse Proxy Server 2.4 Build 06

Release Date: 2010-06-10

30.1 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.

30.2 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.