MUSE

MuseKnowledge™ Source Package Building

MuseGlobal, Inc.
One Embarcadero
Suite 500
San Francisco, CA 94111
415 896-6873
www.museglobal.com

MuseGlobal S.A
Calea Bucuresti
Bl. 27B, Sc. 1, Ap. 10
Craiova, România
40 251-413496
www.museglobal.ro

EduLib, S.R.L.
Calea Bucuresti
Bl. 27B, Sc. 1, Ap. 2
Craiova, România
40 351-420970
www.edulib.com

Version: 1.2
Date: 15th November 2016
Author: EduLib, S.R.L.

MuseKnowledge™ Tools

Building Connectors becomes a production line activity, and maintenance takes its toll.

- **Hand coded connectors take weeks;**
- **Tools reduce times to days;**
- **Tools enable:**
  - Syntax mappers;
  - Code generators;
  - Data conversion generators;
  - Automated code analysis and testing;
  - Overnight builds and delivery;
  - Maintenance.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

Prerequisites for MuseKnowledge™ Connector Development

- **Java Development Kit Standard Edition version 6 or later**
  - This is required for compiling all Java source code
- **Apache Ant**
  - This is a software tool for automating software build processes;
- **Eclipse IDE**
  - This is a Java IDE. It is used for editing, writing and fixing the Muse Connectors;
- **Network Sniffer**
  - In order to build Muse Connectors, the programmer needs software that can spy the network requests and responses of the protocol involved;
- **XML Editor**
  - Editor which supports XML editing, XSL transformation and XSD validation.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

Types of Sources

- **HTML Sources**
  - Are the Sources that send HTTP requests and receive HTML response;
  - There is a software tool for automating build processes;
- **Z39.50 sources**
  - Are the Sources that communicate via the Z39.50 protocol;
- **XML sources**
  - Are the Sources that send HTTP requests and receive XML responses. A subset of these are the SOAP sources for which the HTTP requests are POST requests containing XML data formatted according to the Simple Object Access Protocol;
- **API sources**
  - Are the Sources that are made for a certain API like Endeca, SQL, JXDM, etc. This type of Sources cannot be defined as a group since they are completely different from one another.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

HTML Sources

The main aspects and steps that are followed each time when creating a new HTML Source:

- Profile the new Source using the MuseKnowledge™ Source Package Assistant Tool;
- Update the profile generated by the MuseKnowledge™ Source Package Assistant Tool with the current Source details using a regular text editor;
- Search through the native site for help files or documentation regarding the most advanced query syntax supported by the site, to be able to know the search form to use when creating the Muse translator and Muse Connector;
- Create the Muse Authenticator manually if the Source requires one; Always consider the case when an IP authentication is used for this source. Ideally the source should work with IP authentication as well, but with authenticator removed.
- Create the Muse Connector by using the MuseKnowledge™ Connectors Generator.
- Create the ISR and related query translation files.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

MuseKnowledge™ Builder

MuseKnowledge™ Builder - is a suite of tools used by the Applications and Connectors department in the Source Packages development. These tools are:

- **MuseKnowledge™ Source Package Assistant**
  - Automates the creation and import of the Source Packages related files for the following work tasks types: new, fix or evaluation;
- **MuseKnowledge™ Connectors Generator**
  - Helps developers to easily create and fix modules. The current supported modules are HTTP connectors and extended parsers;
- **MuseKnowledge™ Search Query Translator Generator**
  - Creates DSD (Data Source Description) files and also generates the Search Query Translators files (the generated files are .XSL files);
- **MuseKnowledge™ Source Package Testing**
  - Automates the testing process by automatically searching on the Muse Source Package and provides functionality to analyze the Muse extracted records by adding post-processing operations on the retrieved records.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

MuseKnowledge™ Connectors Generator

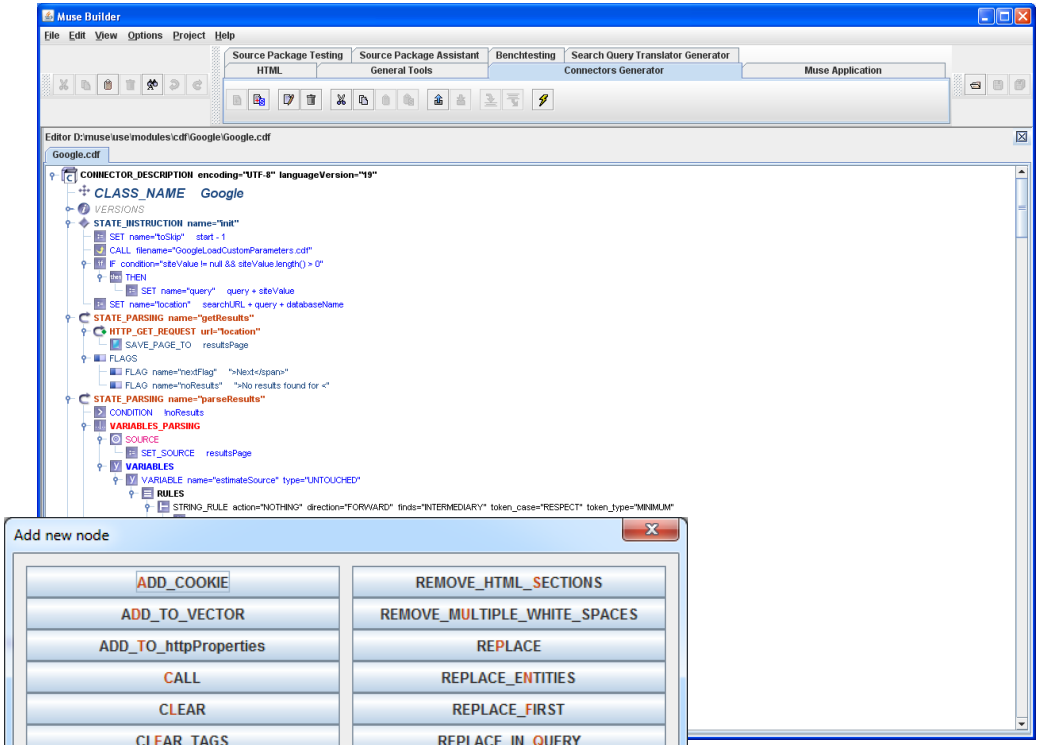
When creating Muse Connectors with MuseKnowledge™ Connectors Generator we have the following scenario:

- There are more than one step needed in order to reach the result page;
- Estimate parsing must be done;
- Record boundaries must be identified;
- Record fields must be parsed;
- Citation parsing on the citation source must be done;
- Date formatting must be done;
- Next page navigation must be ensured.
- Extended page parsing must be done for the Source Packages where is was explicitly requested to do so.
- Any error that can appear on each of the levels above muse be threated and I18N message must be considered for display.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

MuseKnowledge™ Connector Generator

- Easy to use graphical IDE;
- State machine: init state, estimate parsing state, records parsing state, etc.;
- Dozens complex instructions: CONDITION, SET, SETTER, SET_URL, REPLACE, REPLACE_FIRST, REPLACE_ENTITIES, etc.;
- Support for complex parsing;
- Direction of parsing: parsing rules can function in both directions, both forward and backward;
- Auto-completion for thousands of fields in the Data section;
- Positional extraction;
- Citation Parsing;
- Step by Step debugger;
- Multi-level fields.



COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

MuseKnowledge™ Connectors Generator

Besides the standard steps presented, depending on the source, the following cases may also appear, cases that are supported by MuseKnowledge™ Connectors Generator:

- Simulate/duplicate in the Muse Connector's code the native JavaScript processing;
- Generate Muse Connectors with categories;
- Query processing - when the information from the query generated using the Muse SQTG must be further processed before making the search request;
- Database selection and processing - when the database as it is specified in the profile must be broken into pieces before being used in the Muse Connector;
- Repetitive fields block parsing;
- Supports regular expressions parsing rules as well as positional index parsing rules;
- Automatic hidden fields parsing.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

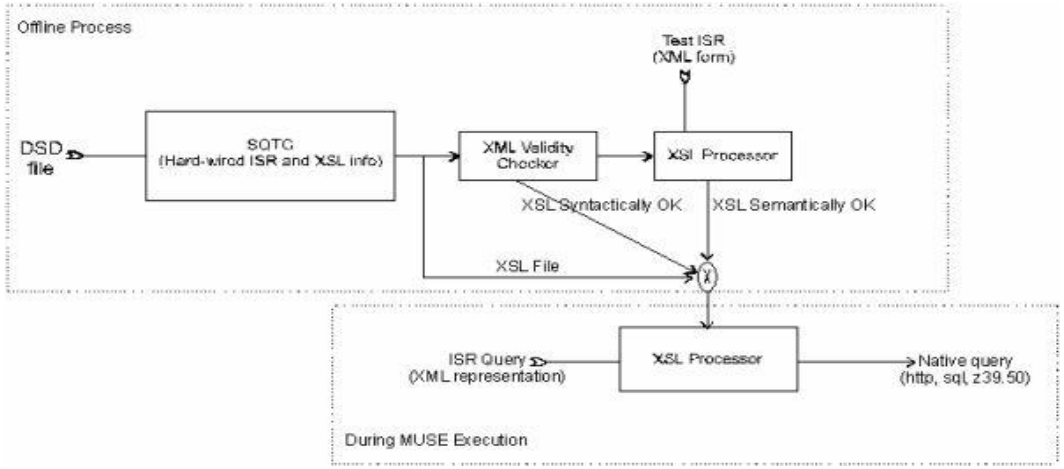
MuseKnowledge™ SQTG

MuseKnowledge™ SQTG is used to create the following files:

- ISR (Internal Search Representation) translator (.XSL file) - which is used to translate the Muse's ISR query into the Source native query;
- Capabilities file (.CPB) - storing the query capabilities supported by that Source;
- Pre-mapping file (.PMF) - storing rules for pre-mapping the query before applying the stylesheet on it;

The DSD (Data Source Description) types supported by SQTG are:

- Normal;
- Split;
- Distributed by operators;
- Composed;
- XML queries.

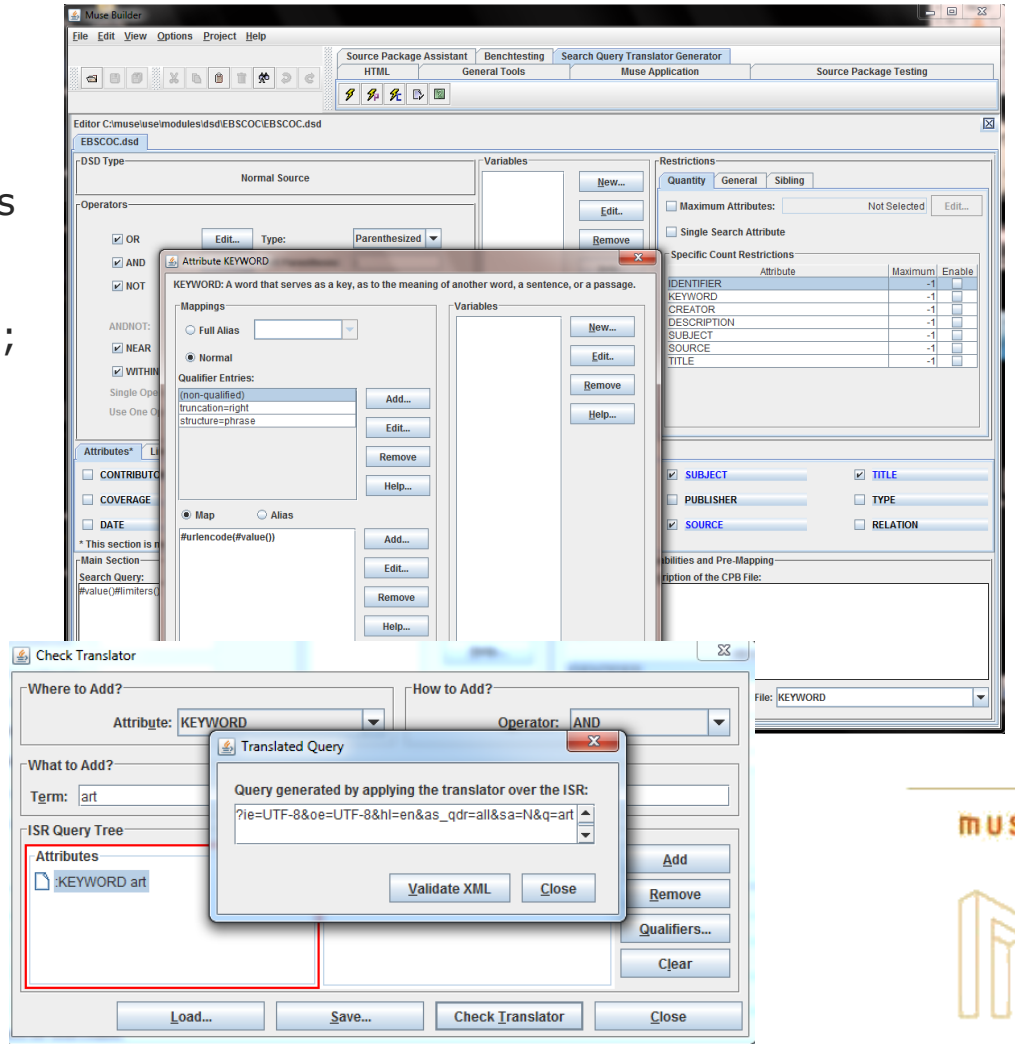


More

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

MuseKnowledge™ SQTG

- Easy to use graphical IDE;
- Convert into totally distinct grammars such as non parenthesized, postfix, infix, splitting and combining terms and operators in their own fields, either simple or with indexes (numerical or literal);
- Complex mappings;
- Support for limiters, Boolean operators, qualifiers (truncation, phrase, etc.;);
- Help;
- Check translator.



COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

MuseKnowledge™ Source Package Testing

The main features of MuseKnowledge™ Source Package Testing tool are:

- Automates test functions for the Muse Source Package (comparative native tests are still manual);
- Helps investigating problems that occur only for a part of extracted records;
- Generates statistics on occurrences of citation fields within records;
- Generates report output (Excel document containing two sheets: Tests and Statistics).

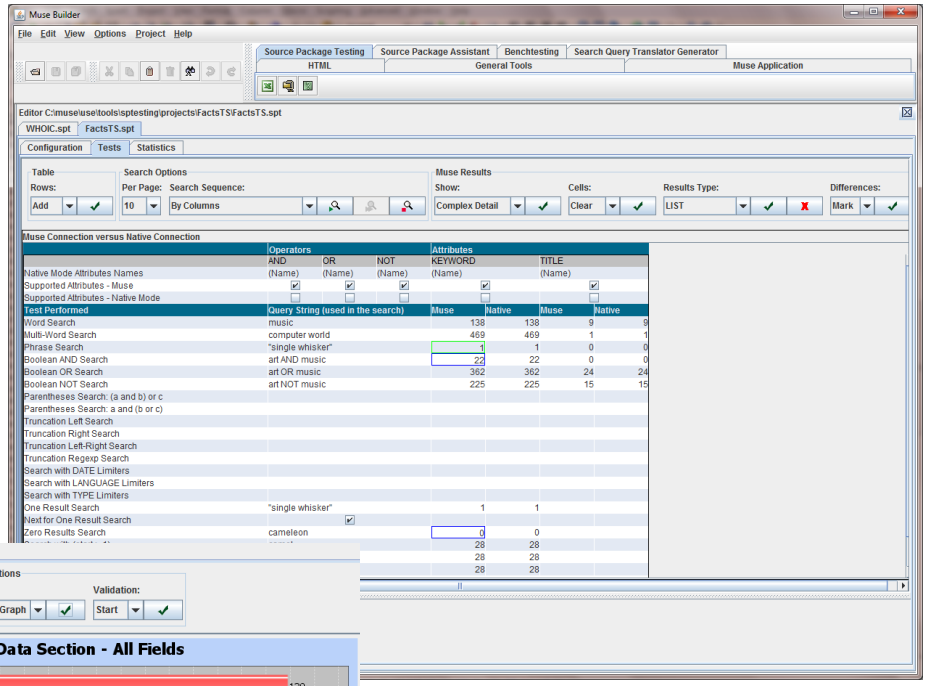
MuseKnowledge™ Source Package Testing supports:

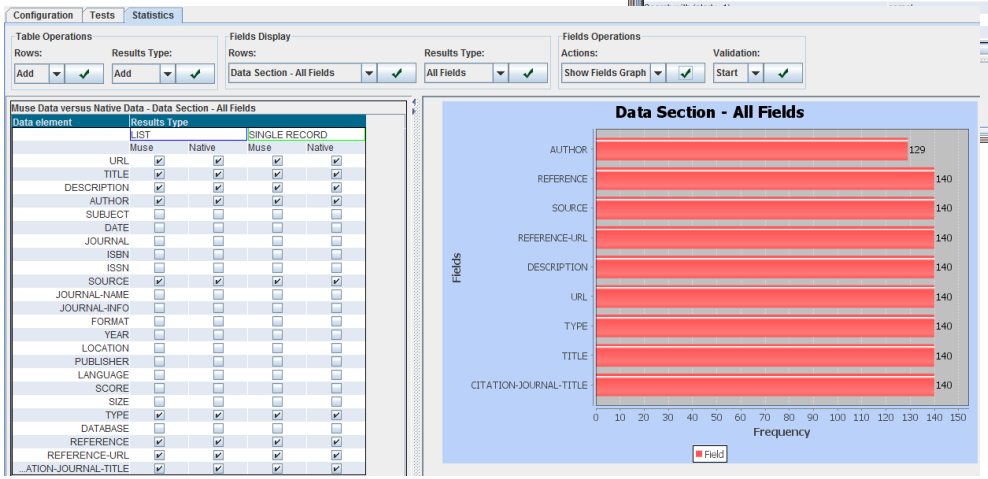
- Comparison between Muse Source Package output and native site on any generic query attribute like keyword, title, description;
- Single-word search term;
- Multi-word search term;
- Phrase search term, etc.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

MuseKnowledge™ Source Package Testing

- Easy to use graphical IDE;
- Configuration, Tests, Statistics sections;
- Complex mappings;
- Support for all query combinations;
- Help;
- Visual check of extracted records;
- Graphical statistics;





COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

Z39.50 Source vs. HTML Source

Tips to consider when developing Z39.50 sources:

- Use the MuseKnowledge™ Source Package Assistant when profiling a Z39.50 source;
- There is a dedicated connector for Z39.50 protocol that it is mainly used for all of the Z39.50 sources;
- Do not use the MuseKnowledge™ SQTG to generate the query for Z39.50 sources, the translator for this type of sources is completely different and not generable from SQTG;
- There are some additional files that are need for Z39.50 sources only, files that are not created also for other types of sources;
- The Z39.50 sources need very little maintenance because the Z39.50 protocol defines beside the communication protocol also the response format and also the information that is stored for each field. So no variations can appear after implementation;
- Requires a Z39.50 client application in order to be able to do requests and understand responses from the native resource.

MUSE

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

XML Source vs. HTML Source

Tips to consider when developing XML sources:

- Use the MuseKnowledge™ Source Package Assistant when profiling a XML source;
- Do not use the MuseKnowledge™ Connector Generator to generate the connector for an XML source, this cannot be generated;
- Use the MuseKnowledge™ SQTG tool in order to generate the query for an XML source;
- Additional map files needs to be created for XML sources only, file that map the fields parsed by the source connector.

Types of XML sources grouped by request:

- HTTP request – The parameters from the request are sent in the key=value format and the response is an XML. For this type of query standard SQTG DSD can be used to generate the query string;
- SOAP request – The parameters from the request are sent as an XML in the post string. For this type of query XML SQTG DSD can be used to generate the query string.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

API Sources

Tips to consider when developing API sources:

- The "API sources" have very little in common with the other source types and they are very different one from another;
- They are practically separate programs;
- The API Muse Connectors must extend ICEConnector class and not ICEHttpConnector class, since the API Sources are not doing, in most of the cases plain HTTP requests.

Types of API sources:

- Java API oriented – Sources that are retrieving records by using a java dedicated API to connect and returns a java object or a java object list as return;

  Eg: Endeca, SAP, etc;

- Protocol oriented – Sources that are retrieving records by using a dedicated proprietary communication protocol, mostly based on HTTP.

  Eg: TN5250, SIP, etc.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

Development Requirements

To create HTML Sources a developer must have at least the following knowledge:

- **Algorithmic thinking** - for understanding the connector logic, deciding which parsing rules are the most suitable ones, understanding the complex JavaScript code used in some pages to compute the parameters sent in the requests, modeling the complex connectors with categories;
- **Java advanced knowledge** – in order to be able to create authenticators, understand the generated connector code;
- **XSLT knowledge** - in order to be able to understand the ISR translators generated and create limiters mapping in SQTG;
- **HTTP, HTML , Javascript, CSS advanced knowledge** - to be able to understand the request/responses to/from server and create parsing rules;
- **Regular Expressions advanced knowledge** – to be able to create citation parsing and date formatting.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

Finalizing Source Package Development

After the Source Package development is completed by the developer, the Source Package Assistant Tool is used to complete the development by following the below steps:

- The Developer publishes the Source Package in his Outbox directory together with the tests performed;
- The Team Leader takes the Source Package published by developer and import it in his local application for testing purposes. While importing the Source Package the Source Package Assistant Tool is also performing a large variety of consistency tests;
- The Team Leader ensures that all the work done by the developer is in conformity with the requirement form the partner;
- If everything works correctly with the Source Package, the code is committed into the CVS repository;
- An email notifying the customer regarding the work done for this Source Package is also sent.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

Source Packages Automatic Build

MuseKnowledge™ Control Center Tool is used to build the Source Packages final .jar file from the code committed in CVS by Team Leader. The steps followed are:

- Get the latest updates committed in the CVS repository;
- Build the .java sources into binary .classes files;
- Run various consistency tests on the code and configuration files added in the repository;
- Build binary .jar files, one for each new/updated Source Package(not updated Source Packages will not be re-built);
- Upload the new .jar files created to ftp.museglobal.com in order to be later uploaded to Muse Source Factory by another process.

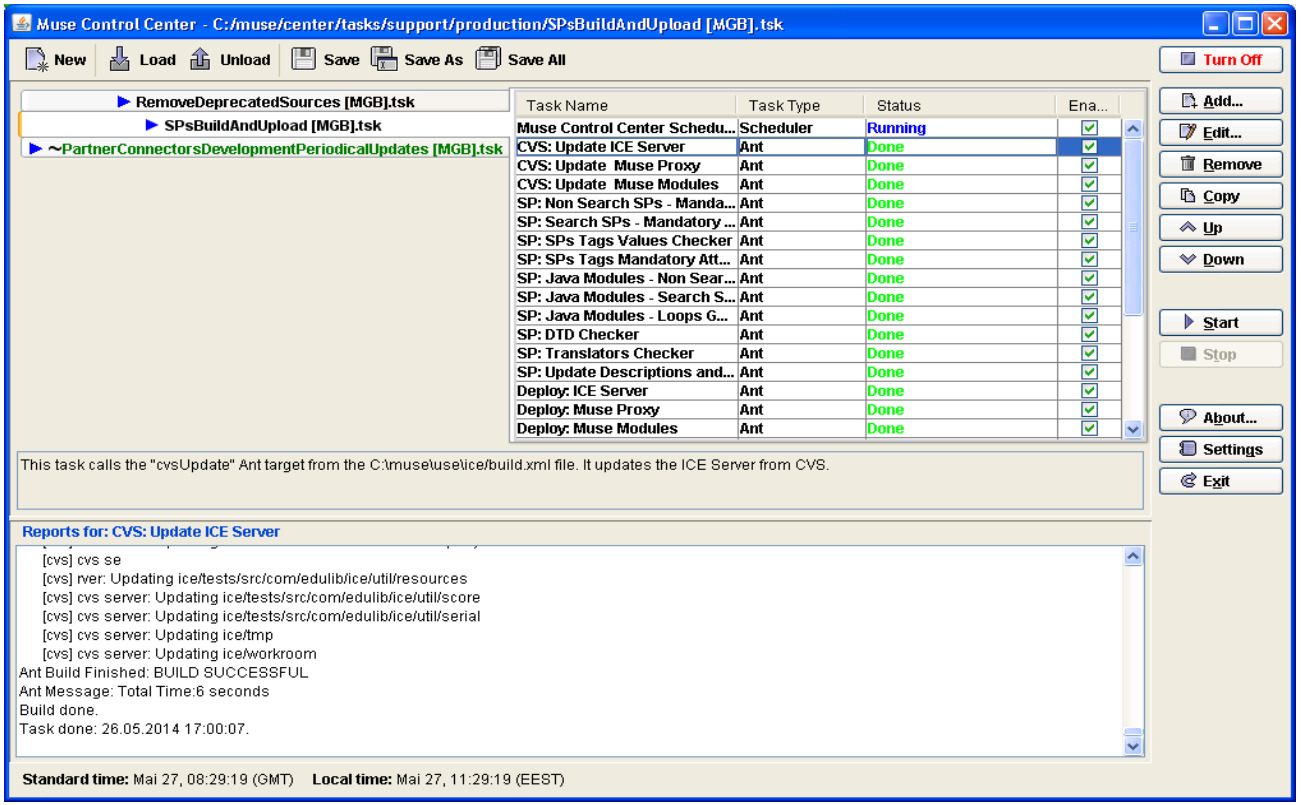
Note:

- This build is automatically triggered at 15:00 GMT each day from Monday to Friday.

COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

Source Packages Automatic Build

MuseKnowledge™ Control Center task file for the Source Packages Automatic Build – individual tasks chained to achieve sequentially the building of Source Packages on a daily basis and to upload them on an FTP location.



COMPANY CONFIDENTIAL © 2016 MUSEGLOBAL, INC. ALL RIGHTS RESERVED

MUSE

MuseKnowledge™ Source Package Building