Test Ditamap

Contents

Introducing PHEMI Central

PHEMI Central is a big data warehouse that offers big data capability with fully integrated privacy, security, and governance and advanced data management functionality.

PHEMI Central allows organizations that need to protect and govern the use of their information to take advantage of big data technology to access, catalogue, and analyze their digital assets at speed and scale.

Data in PHEMI Central

To be written.

This topic will follow an item of data as it traverses PHEMI Central. It will include diagrams. It will start at the data source, through collect, through curate, and through consumption by users, machine destinations, applications, and so on.

Data in PHEMI Central

Data in PHEMI Central follows a lifecycle of collect, curate, and consume.

Throughout the data lifecycle, data is managed according to the organization's governance policies.

Submitting Data to PHEMI Central

Data can be submitted to PHEMI Central using the PHEMI RESTful API, by manually ingesting it, by using FTP or SSH batch ingestion, or by using extract, transform, and load (ETL) tools.

Administrator Quick Start

This section shows you the workflow for a first configuration of the PHEMI Central Management and Governance Console.

Introducing the Management and Governance Console

The Management and Governance Console is a web-based interface for configuring and monitoring PHEMI Central.

Note: Use either Apple Safari, Mozilla Firefox, or Google Chrome to access the PHEMI Central Management and Governance Console. Microsoft Internet Explorer is not supported.

Management and Governance Console Reference

This section steps you through all the functions and procedures in the Management and Governance Console

Filter Expressions

Filter data in reports, logs, and lists of objects using simple regular expressions.

Regular expressions used to filter information in PHEMI Central must be compatible with both the Java and the Python programming languages.

The simple regular expressions described in this section are supported by the system. More complicated regular expressions may succeed, but are not guaranteed to succeed.

Character Classes

Generally, the allowed character classes include lowercase letters, uppercase letters, numbers (decimal digits), and the underscore character ("_"). Taken together, these characters comprise the "word" character class.

To match a specific string of letters, numbers, punctuation, or a mix, simply enter the string. For example entering the string `2014` matches any entry containing the string "2014". A blank space is entered simply as a blank space.

To make more flexible expressions, use *metacharacters* and *escape sequences*.

Metacharacters

Metacharacters are characters that are not matched themselves, but indicate how a filter expression should be interpreted.

Character(s)	Description
[ ]	Square brackets. Matches any single character within the brackets. For example, `[abc]` matches any of the characters "a", "b", or "c". Backslash escapes are not allowed within square brackets. To filter on a sequence of more than one character inside brackets, delimit the expressions using a single quote; for example, ['11"12"13'] A closing square bracket may be included in the bracket expression if it is the first character of the set (or the first character after the carat), as in `[]abc]`.
-	Hyphen. When within square brackets, specifies a range of values. For example, `[a-z]` matches any lowercase letter from "a" through "z". The expression `[a-d]` is equivalent to `[abcd]`. The hyphen is treated as a literal character if it is the first or last character inside square brackets, as in `[abc-]` or if it is the first character after the carat inside square brackets.
[^]	Matches the negation or complement of the characters within the brackets. For example, `[^abc]` matches any character that is neither an "a" nor a "b" nor a "c", while `[^a-z]` matches any character that is not a lowercase alphabetic character. When outside square brackets, the carat is matched as an ordinary character.
^	Matches the starting position of the string or line.
$	Matches the ending position of a string or line, or the position immediately preceding the newline character.
\	The escape character. When followed by an ordinary character, the escape character signals a special sequence; for example, `\n` indicates a newline character. When followed by a metacharacter, the metacharacter loses its special meaning and is matched in the same way as an ordinary character.

Character(s)	Description
.	Dot, or period. Matches any single character, except newline (\n). For example, `.at` matches "bat", "cat", "hat", "mat", and so on. Within a bracket expression, the dot matches a literal dot. Therefore, while `.at` matches "bat", "cat", and so on, `[.at]` matches "." or "a" or "t".

Escape Sequences

There are certain characters that take on a new meaning when they are preceded by the escape character, the backslash.

Escape Sequence	Description
\n	Newline.
\r	Carriage return.
\t	Horizontal tab.
\x0B	Vertical tab.
\f	Form feed.
\d	A digit. Equivalent to [0-9].
\D	A non-digit. Equivalent to [^0-9].
\s	A whitespace character. Equivalent to [\t\n\r\x0B\f].
\S	A non-whitespace character. Equivalent to [^\s].
\w	A member of the word character class. Equivalent to [a-zA-Z_0-9].
\W	A non-word character. Equivalent to [^\w].

Glossary of Terms and Concepts

access policy

Lowercase (i.e., not "Access Policies").

An access policy is a set of logical rules that determines how users can consume data stored in PHEMI Central. The access policy specifies what user authorizations are required to interact with data tagged with specified sensitivity, or visibility. Access policies can be applied to data collections and datasets.

authorizations

Lowercase (i.e., not "Authorizations").

User authorizations are configurable attributes you can assign to PHEMI Central users. Authorizations are defined in PHEMI Central by the PHEMI Administrator, who sets them in accordance with the organization's governance policies.

big data

Do not capitalize. Do not hyphenate when used as an adjectival phrase (i.e., "big data warehouse" not "big-data warehouse").

cell (field)

A cell, or field, is the smallest unit of data storage in PHEMI Central. A cell is a single data item, which can range from a single byte up to gigabytes, plus the metadata associated with the data item. Any piece of raw data, regardless of size, is stored in a single cell. Elements of derived data (transformed from the raw data) are also each stored individually in cells. Any cell can be protected by applying data visibilities.For derived data, each derived item can be individually assigned a visibility (which may be different than that configured for the data collection) by the DPF performing the processing.

code library

A code library is a package of executable code that is included in a DPF archive. Whether the code is source or compiled depends on the coding language. Code libraries must be portable and self-contained; that is, all dependencies required for the DPF to function must be bundled inside the library, in the appropriate way, for whatever language is being used.

dataset

One word (i.e., not "data set").

A dataset is a view, or map, of an underlying set of data. Data items in a dataset can be selected from across multiple data collections. The dataset is a view, or map, to the underlying data. The actual content of the dataset (that is, the dataset's data) is generated when the dataset is executed or when it is queried against.

data category

Not "data collection category."

Data categories are a way to classify data into broader groupings. Examples of data categories are "Research Reports," "X-Rays," and "Prescriptions."

Data Processing Function, DPF

Capitalized.

A Data Processing Function, or DPF, is an executable piece of code that supplies the instructions for processing raw data (for example, a log message or medical report) to extract from heterogeneous data collections meaningful, context-specific information (such as a temperature reading or blood glucose measurement) that can be queried or exported for analysis. The code is executed by the PHEMI Central DPF Engine, which uses it to direct curation of the data. The input to a DPF is the raw binary data ingested into the system. The output of a DPF is a set of structured elements, each of which includes a type property (for example, INT or STRING) and

can specify data visibilities (for example, SECRET or IDENTIFIABLE) on a per-field basis. The data elements output by a DPF are called derived data. The collection of derived data produced by a DPF is automatically indexed in PHEMI Central.

data collection

Two words (i.e., not "datasource").

In PHEMI Central, a data collection is the set of management and governance rules and policies that will be applied to a collection of data. A data collection configuration should be defined for each set of data that is to be stored and managed according to the same retention, legal, and governance rules.

data visibilities

See visibilities.

derived data

Derived data is data that has been parsed, extracted, or otherwise enriched or processed by running a DPF on stored raw data. The set of derived data items can be searched, queried, further processed, or exported from the system.

digital asset

A digital asset is any piece of data stored with metadata in the system. This may be raw data that has had metadata applied on collection, or it may be derived data that has been parsed, indexed, catalogued, and/or enriched with additional metadata.

DPF archive

Note that "archive" is lower case.

The set of code that makes up a DPF is called a DPF archive. A DPF archive is delivered as a ZIP file archive. It consists of two parts: a manifest file and a code library. To associate a DPF with a data collection, the DPF archive is ``registered`` with the data collection by uploading the archive during data collection configuration.

ETL

Extract, transform, and load. In databases, a set of tools or processes that extracts data from sources, transforms the format or structure for storage, query, and analysis, and loads it into the receiving or consuming system.

ingestion

Ingestion is the process by which data is brought into in PHEMI Central. The sending system (the data source) submits the data to PHEMI Central, which listens for the data using a web service. Data can also be ingested manually, by using the PHEMI Central Management and Governance Console. The specific characteristics of data ingestion can be specified per data collection as part of the data collection configuration.

JSON

JSON stands for JavaScript Object Notation. JSON is a lightweight data-interchange format that is easy for humans to read and write and easy for machines to parse and generate. JSON is used in the body of several REST requests in the PHEMI RESTful API. PHEMI Central also includes a system DPF that can create derived data from JSON objects, providing the objects conform to PHEMI's JSON specification.

key-value pairs

A key-value pair is a set of two linked data items: a key which uniquely identifies some item of data, and the data itself. PHEMI Central uses key-value store to efficiently store, process, and retrieve data.

M2M

M2M is a way of referring to machine-to-machine interfaces, used in machine-to-machine communication.

manifest file

A manifest file is a JSON file that specifies the output of a DPF. With the code library, the manifest file makes up the DPF archive that is uploaded to register the DPF with a data collection. The manifest file should include the properties of the DPF along with the details of each derived data item to be generated.

metadata

One word (i.e., not "meta data")

Metadata is information about a piece of data. In PHEMI Central, metadata is information about how a given piece of data is to be managed. When a piece of raw data is ingested into PHEMI Central, information from the connection (for example, the timestamp) together with policy information configured for the data collection (for example, the data visibility) and some derived information (for example, a "time to live," as derived from the timestamp and the data retention policy) is used to create metadata properties that are stored with the data. Further, PHEMI Central also automatically indexes and catalogues all stored data, whether raw or derived; the indexes and catalogues can also be considered a kind of metadata.

PII

Personally Identifiable Information, or PII, is a legal concept used in US privacy law and information security to mean information that can be used on its own or with other information to identify, contact, or locate a single person or to identify an individual in context. When thinking about PII, it is important to distinguish legal requirements to remove attributes uniquely identify an individual from a general technical ability to identify individuals. Because of the versatility and power of modern re-identification algorithms, together with the amount of information freely available from all sources, the absence of PII data does not guarantee that de-identified data cannot be used, perhaps in combination with other data, to identify individuals.

privacy-level visibilities

Privacy-level visibilities are data visibilities that characterize the privacy level of a data item. PHEMI Central includes predefined privacy-level visibilities designed to apply to data domains where privacy is important.

- **IDENTIFIED**. The data contains Personally Identifying Information that potentially identifies an individual. Examples of information of this type include name, Social Insurance Number, and date of birth.
- **DE-IDENTIFIED**. The data contains IDENTIFIED information that has been masked or encrypted.
- **NON-IDENTIFIED**. The data is not identifying in and of itself. Examples of this type of information include weight or favorite food.

Although privacy-level visibilities are preconfigured, their descriptions can be modified by configuration.

privacy, security, and governance

Always use the same order. Do not capitalize (unless in a heading or title).

Privacy by Design

Title case (i.e., capitalize "Privacy" and "Design"; not "by"). If shortened, the acronym is "PbD."

raw data

In PHEMI Central, raw data items are files, objects, records, images, and so on that are submitted for ingestion into the system. Raw data is stored exactly as received, along with the metadata generated for it on ingestion.

REST, RESTful API

"REST" is all caps. The "ful" in "RESTful" is lower case.

Representational Statement Transfer (REST) is an architectural style that uses HTTP requests and associated methods (POST, PUT, GET, and DELETE) to create, update, read, and delete data. A RESTful API is an application programming interface (API) based on REST.

visibilities

All raw data and derived data stored in PHEMI Central can be tagged with labels that provide information about the data's sensitivity. This sensitivity is described in terms of the visibility the data should have to different system users. The visibility tags you define for your data should reflect the sensitivity of the data as identified by your organization.

ZIP file

"ZIP" is all caps.