Generic standards specification

Statement of Work

Version 0.2

This document is confidential and contains proprietary information

Generic standards specification

Version

Version	Author(s)	Date Released	Notes
0.1	Dominic Marshall	21 September 2021	Commenced
0.2	Dominic Marshall	19 October 2021	Updated after initial feedback
0.3	Dominic Marshall	8 November 2021	Updated after further initial feedback
0.4	Dominic Marshall	3 December	Update to format functions after considering issues if uc formatting was used

Background

The Darwin Core tabs and fields were included in the ecatalogue module in the early 2000s as a response to the need for the biodiversity community to format data and output it for sharing in a standard format. They were added as a means of simplifying data exports and reports by mapping data from associated modules into atomic readonly fields in the catalogue. This choice was made to circumvent reporting limitations with regards to pulling data from complex data structures, such as nested and doubly nested tables, and reformatting it. An additional set of Darwin Core fields was added at a later stage (some OBIS and MaNIS fields), but these have not been revised since. Over time a large proportion of these original fields have been deprecated or formatting has changed.

To use the Darwin core fields, each Natural History institution must map its own schema to the Darwin Core fields. The mappings are added to EMu by Axiell and not by the institutions themselves. Over time this results in a disparity between what is in the EMu Darwin Core fields and what is required by the data standards because these mappings are not updated.

NYBG took an alternative approach and developed a standalone Darwin Core module. The fields in Darwin Core module function in exactly the same fashion as those in the ecatalogue module. Likewise, they suffer from the same limitations as the catalogue fields, in that the mappings are not updated to reflect the most up-to-date versions of the standard.

The static nature of the existing solutions has made many original reports obsolete and new reports are very complex due to the need to reformat data to map to the new standard. In some cases the original readonly fields can no longer be used because they do not reflect the updated standard at all. These reports are usually run through crystal so that a properly formatted CSV file can be exported. Crystal Report's has an extremely slow record conversion process and so large data sets can require significant processing time, sometimes in the order of days.

In 2020 FMNH added standard tabs and fields to the multimedia module to accommodate the Audubon Core standard. The process used to generate the data for these fields is the same as used in the catalogue and so is subject to the same limitations.

It is anticipated that over time further modifications will be made to existing standards and new standards will appear. As institutions increasingly focus on the ideals of FAIR data a method to produce new, up to date datasets is required. The method must allow for the definition of individual data elements which may be grouped in various ways to generate exports in standard formats.

Purpose

The purpose of this document is to detail the changes required to create a generic mechanism that may be used to produce data exports in the manner outlined in the Standards: Structural Changes Doc [estandards] document produced by FMNH. A summary of the inadequacies of the existing Darwin Core reporting mechanism within EMu highlighted in this document can be found in the Background section above..

The aim of this specification is to create a means by which individual data elements may be configured and tied together so as to define a standard export format. The definition of data elements must allow for complex data structures to be taken and simplified into individual data elements. Each definition must be able to be updated and upon being updated, take immediate effect. Any number of different export formats should be allowed to be defined.

An outcome of the delivery of this statement of work is a step towards export conformity across institutions by allowing all institutions to use the same export definitions albeit tailored to their individual schemas. The changes involve the creation of a new module to store the data definitions, a mechanism to initiate exports of data formats and a mechanism to store generated exports allowing for later referral.

It is desirable that the data definitions be simple enough that any EMu user would be capable of defining them rather than requiring a resource from IT or Axiell to perform the task. Storing the definitions in a module allows for them to be easily exported by one institution and then imported by another. Any import should flag records in the import where a definition contains a column not present in that institution's schema so these may be updated. An export should not run if an invalid data definition is present.

Intended use

This document is intended as a discussion document between Axiell and FMNH. It is a work in progress and until it reaches an agreed state it should not be circulated to other parties. Once agreement has been reached by all parties, the document may be released to other interested institutions.

General Requirements

The following general requirements apply to the statement of work described in this document:

- All work will be carried out on the EMu 6.0 source code branch and released at part of EMu 8.
- The standard form height for an EMu module will be increased by 20% for the Standards Module. This will allow all of the relevant definition fields to be displayed on one tab.
- Exports should be able to be generated in XML, CSV or JSON. The addition of JSON as a standard export format is scheduled for release with the EMu 8.2 minor release later in 2022.

General Questions

The following general questions apply to the functionality of the statement of work described in this document:

1. Are the existing Darwin Core fields used for query purposes or are queries performed against the component fields?
2. What happens with the existing Darwin Core module and fields given the proposal is to now only store the data in the export?

Statement of Work

The work required to implement the generic standard functionality to allow data definitions and exports to be produced from those definitions can be broken down into the following sections:

- New Standards Module
- Example term definitions
- Expand existing exports process to allow for defining a standards export

Each of these sections is examined in some detail.

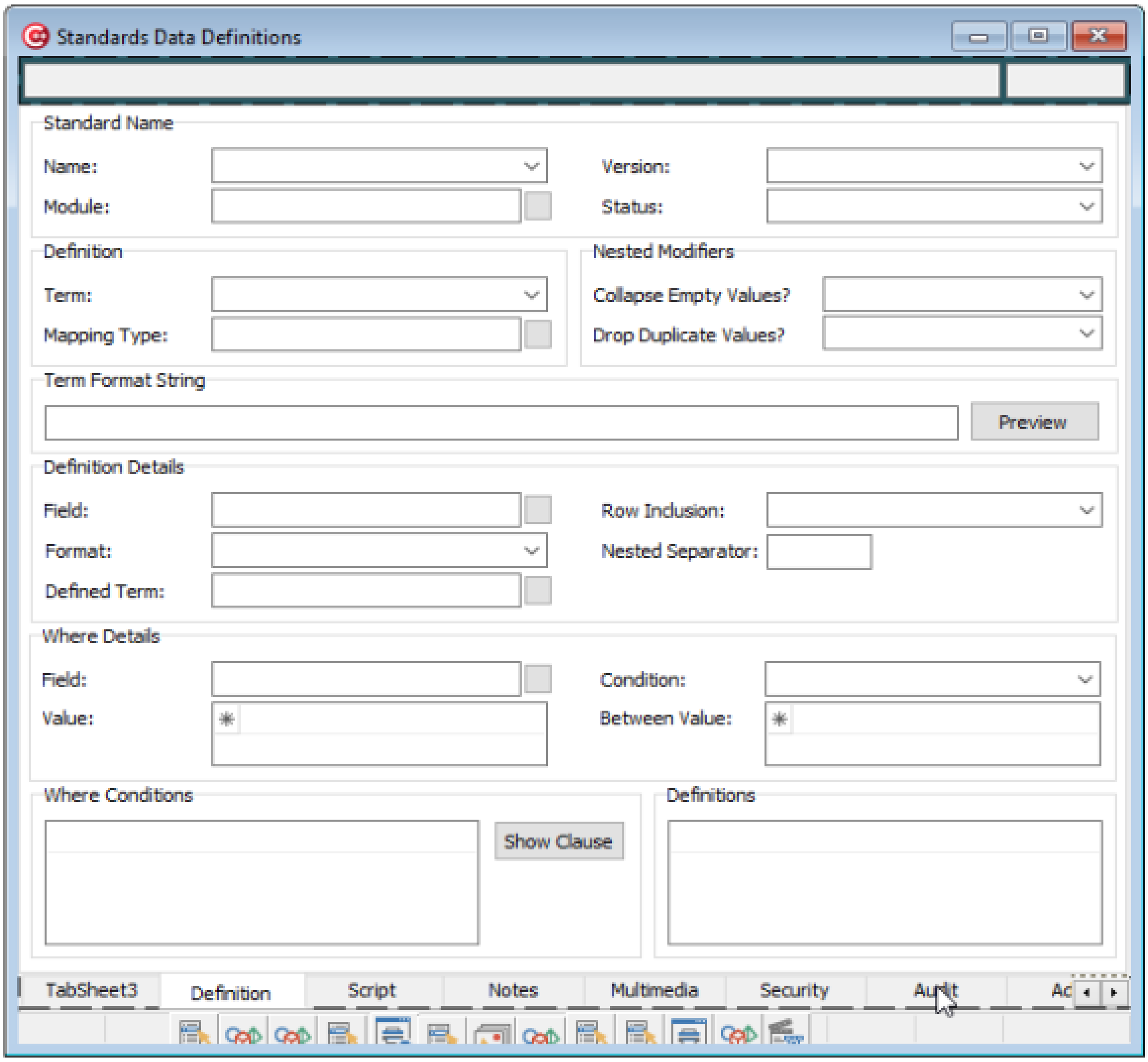
New Standards Module

A new module is required to hold term definitions for standards The module will be made up of two tabs to hold the term definition and the standard EMu tabs available in each module (Notes, Multimedia, Security, Audit, Admin).

Definition tab

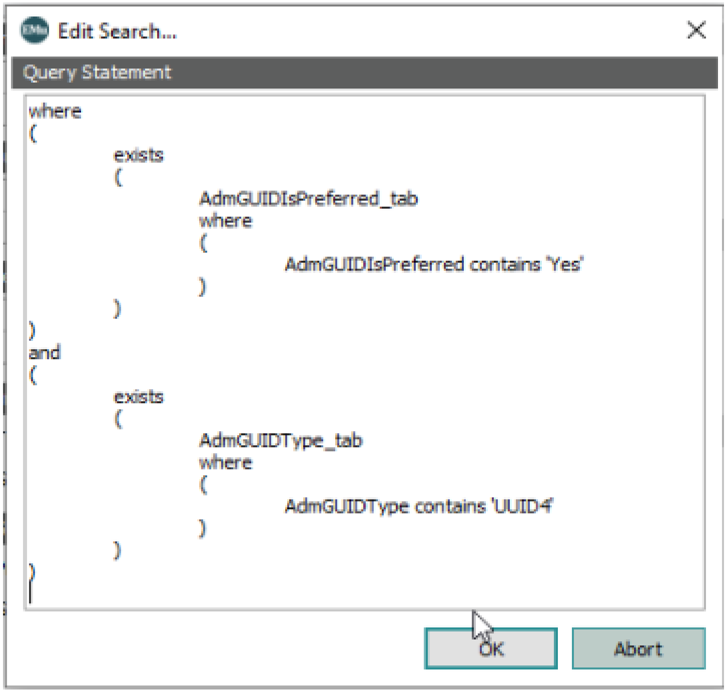
The Definition tab is required to hold data that makes up a term definition for a standard.

The proposed screenshot of the new tab is below for reference:



Nested Functionality

The Definition Details and Where Details group boxes provide fields which have nested functionality controlled by the Definitions and Where Conditions grids at the bottom of the tab. Selecting a row in the Definitions grid will refresh the fields in the Definition Details group box while selecting a row in the Where Conditions grid will refresh the fields in the Where Details group box. The Show Clause button in the Where Conditions group box will display the where clause generated by the specified fields in the Where Details. The clause will be shown in a read only window similar to the Edit Search window show as part of the existing EMU Show Search functionality.



```
where
(
        exists
        (
                AdmGUIDIsPreferred_tab
                where
                (
                        AdmGUIDIsPreferred contains 'Yes'
                )
        )
)
and
(
        exists
        (
                AdmGUIDType_tab
                where
                (
                        AdmGUIDType contains 'UUID4'
                )
        )
)
```

Example output for where conditions has been included with the Clause Prompt (WhereClause Column) in the example term definitions below.

Field mappings

The table below maps each of the above fields to its corresponding backend column:

Prompt	Column	Type	Notes
Name	StandardName	List Text (30)	The name of the standard the definition applies to. New lookup list with name "Standard Name ".
Version	StandardVersion	List Text (30)	The version of the standard the definition is compliant with. New lookup hierarchy with name "Standard Version " with parent “Standard Name”.
Module	Module	List Text (30)	The module the standard export is initiated from.. Lookup list generated on modules available through the “Table Access” registry entry.
Status	Status	~~RadioButtons~~ List Text (30)	A status indicator to indicate whether or not a term from a defined standard should be included in an export.

Term	StandardTerm	List Text (50)	The name of the term within the standard. New lookup hierarchy with name "Standard Term " with parent “Standard Name”. The value entered into this field must be unique.
Mapping Type	MappingType	List Text (30)	The type of mapping to apply to define the data. New lookup list with name "Mapping Type ".
Collapse Empty Values	CollapseEmptyValues	Text (10)	Used to indicate that if field values are empty their corresponding separator should be dropped. Radio Button values set to Lookup list “System Yes” and “System No”.
Remove Duplicate Values	RemoveDuplicateValues	Text (10)	Used to indicate where separate fields used to produce an output element contain the same value, whether or not the duplicate value should be removed. Radio Button values set to Lookup list “System Yes” and “System No”.
Term Format String	TermFormatString	Text (1000)	A simple formatting string to generate the term output.
Field	Field	Text (100)	The EMu Field to use for extracting data for this definition. When selecting the add field button at the end of the field the standard EMU field selector will be shown (see Adding Fields below).
~~Separator~~	~~FieldSeparator~~	~~Text (255)~~	~~Where more than one field is used in a definition what text should be used to join this field with the previous. If no value is specified the separator will default to the space character.~~
Format	Format	Text (50)	Predefined list of format functions based on the column type.
Defined Term	DefinedTerm	Text (50)	For use where the value of an already defined term in the standard is required. When selecting the add term button at the end of the field a list of already defined terms for the

			standard will be shown.
Row Inclusion	RowInclusion	List Text (20)	An indicator of which row(s) to include when the field is a nested table. New lookup list with name "Row Inclusion".
Separator (Nested)	NestedSeparator	Text (255)	Where nested tables are used to generate the data for this definition, the separator to use between each of the generated rows. If no value is specified the separator will default to the pipe character.
Field (Where)	WhereField	Text (100)	The EMu Field to use for defining the row condition for matching data to choose a nested table row. When selecting the add field button at the end of the field the standard EMU field selector will be shown (see Adding Fields below).
Condition	WhereCondition	List Text (100)	The description of the condition to be used to match the nested table row. New lookup list with name "Where Condition".
Value	Value	List Text (100)	The value of the condition to be used to match the nested table row.
Between Value	BetweenValue	List Text (100)	The second value of the condition to be used to match the nested table row when the condition is Between.

New Lookup Lists

Prompt	Column	Values	Permission
Name	StandardName	Darwin Core	Default: ReadOnly Admin: ReadWrite
Version	StandardVersion		Default: ReadOnly Admin: ReadWrite
Term	StandardTerm		Default: ReadOnly Admin: ReadWrite
Mapping Type	MappingType	Complex Nested Or Simple	Default: ReadOnly Admin: ReadWrite

		Script	
Format	Format	Refer to Format Functions	Default: ReadOnly Admin: ReadWrite
Row Inclusion	RowInclusion	All Rows First Row Last Row Where All Where First Where Last	Default: ReadOnly Admin: ReadWrite
Condition	WhereCondition	Between Contains Does Not Contain Does Not Equal Equals Greater Than Less Than Is Null Is Not Null	Default: ReadOnly Admin: ReadWrite

The mapping type of **Or** will sequentially evaluate each condition until a condition evaluates to be true. The Or type can be combined with another type e.g. **Nested Or** to generate output covered by both types

Format Functions

A suite of standard formatting functions will be supplied for each data type. This formatting will be applied prior to any Term formatting.

Data Type	Function Name	Description
Text	Uppercase	Convert all text characters to uppercase e.g. UPPERCASE
	Lowercase	Convert all text characters to uppercase e.g. lowercase
	Titlecase	Convert the first character of each word and uppercase and the remaining characters to lowercase e.g. Title Case
Date	ISO-8601	ISO standard date format: YYYY-MM-DD
	DMY	Date formatted in day, month, year order separated by Nested Separator (/) e.g. DD/MM/YYYY
	MDY	Date formatted in month, day, year order separated by Nested Separator (-) e.g. MM-DD-YYYY
	YMD	Date formatted in year, month, day order separated by Nested Separator ()[empty defaults to no space] e.g. YYYYMMDD

	DMonY	Date formatted in day, month (abbreviated), year order separated by Nested Separator ()[empty defaults to a space] e.g. 01 Oct 2021
	MonDY	Date formatted in month (abbreviated), day, year order separated by Nested Separator ()[empty defaults to a space] e.g. Sep 10, 2021
	YMonD	Date formatted in year, month (abbreviated), day order separated by Nested Separator ()[empty defaults to a space] e.g. 2021 Aug 21
	DMonthY	Date formatted in day, month (full), year order separated by Nested Separator ()[empty defaults to a space] e.g. 01 October 2021
	MonthDY	Date formatted in month (full), day, year order separated by Nested Separator ()[empty defaults to a space] e.g. September 10, 2021
	YMonthD	Date formatted in year, month (full), day order separated by Nested Separator ()[empty defaults to a space] e.g. 2021 August 21
Time	ISO-8601	ISO standard time format: hh:mm:ss.sss
	24hr	24 hour clock format hhmm
	HMS	Time formatted for 24 hr clock in hour, minutes, seconds order separated by Nested Separator e.g. hh:mm:ss
	DHMS	Time formatted for 12 hr clock (duodecimal) in hour, minutes, seconds order separated by Nested Separator e.g. hh:mm:ss AM/PM
Latitude	DMSA	Latitude formatted in degrees, minutes, seconds order e.g. 30°12’33.22”
	CDMSA	Latitude formatted in compass degrees, minutes, seconds order e.g. 30°12’33.22”N
	DDA	Latitude formatted in decimal degrees e.g. 30.125
	CDDA	Latitude formatted in compass decimal degrees e.g. 30.125N
Longitude	DMSO	Longitude formatted in degrees, minutes, seconds order e.g. -50°09’25.26”
	CDMSO	Longitude formatted in compass degrees, minutes, seconds order e.g. 50°09’25.26”W
	DDO	Longitude formatted in decimal degrees e.g. -50.093
	CDDO	Longitude formatted in compass decimal degrees e.g. 50.093W

Term Format String

A format string defined using perl sprintf format specifiers https://perldoc.perl.org/functions/sprintf. This provides great flexibility for generating output text without the need to define any processing script.

Preview Button

A new Registry entry will be added which will allow for specifying a record to use to test the generated output for a term definition. The entry will have the following format:

System|Setting|Standards Preview Record|Standard Name|irn

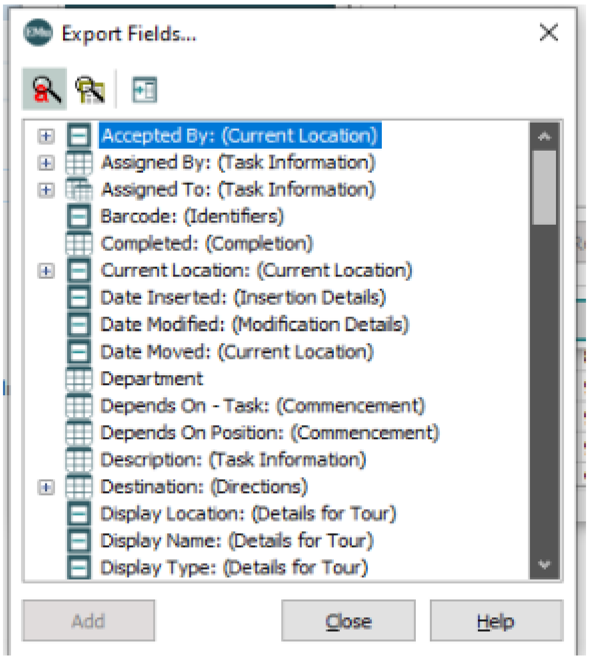
e.g. System|Setting|Standards Preview Record|Darwin Core|1234

This entry defines that when a term is being defined for Darwin Core, to test the generated text for the term definition, irn 1234 shall be used to extract the data elements.

The Preview button will be enabled if this entry is defined for the standard and if either the Term Format String or Script field are filled.

Adding Fields

When using the add field button the standard EMU field selector shall be displayed.


Export Fields...
Accepted By: (Current Location)
Assigned By: (Task Information)
Assigned To: (Task Information)
Barcode: (Identifiers)
Completed: (Completion)
Current Location: (Current Location)
Date Inserted: (Insertion Details)
Date Modified: (Modification Details)
Date Moved: (Current Location)
Department
Depends On - Task: (Commencement)
Depends On Position: (Commencement)
Description: (Task Information)
Destination: (Directions)
Display Location: (Details for Tour)
Display Name: (Details for Tour)
Display Type: (Details for Tour)
Add
Close
Help

This will allow for the selection of the field to be used for the term definition. Where more than one field is required to make up the definition, each field is added by selecting a new row in the definitions grid at the bottom.

Adding Defined Terms

When using the add defined term button a selection list will be displayed which shows all of previously defined terms for the standard which are not depreciated.

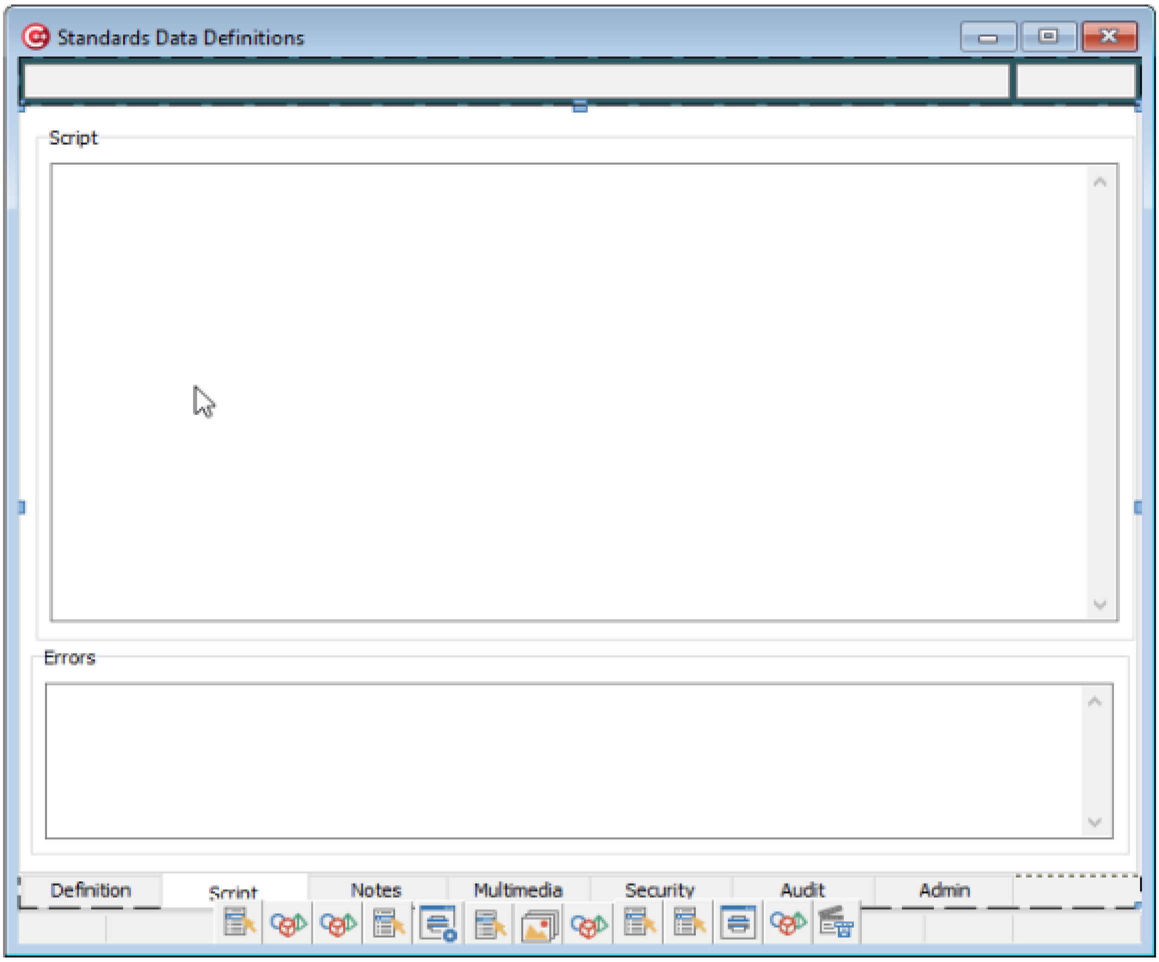
Additional Functionality

When generating a where query it is possible that more than one row could match the condition. Where this occurs the data from each matching row should be included for the term separated by the Nested Separator. If the Nested Separator is not specified it will default to the pipe character (|).

Script tab

The script tab is required to hold script text for formatting data for a term where a Complex relationship between fields needs to be defined.

The proposed screenshot of the new tab is below for reference:



Field mappings

The table below maps each of the above fields to its corresponding backend column:

Prompt	Column	Type	Notes
Script	Script	Text (10000)	The scripting code to format the data for the term.
Errors	Errors	Text (10000)	A listing of errors identified in the script field.

Additional Functionality

It is envisioned that the scripting will all be done using a preferred language. The reason for this efficiency. While multiple languages could be supported it is extremely expensive on processing time to execute a script in a language other than the language of the main processing script. The main programming language for scripts run by the EMu server is Perl. To get the most efficiency gains from allowing scripting all script code should be written in Perl. The script will be passed a reference to an array of data. Each array element will contain data for a corresponding field in the term definition. The order of array elements will match that of the field definitions.

It is understood that Perl may not be the preference of all customers but it is also understood that no language will be universally accepted either, hence a decision had to be made. To cater for other language preferences the ability to run a post process on the generated output has been included (see Expanding existing export process below).

Example term definitions

Example 1

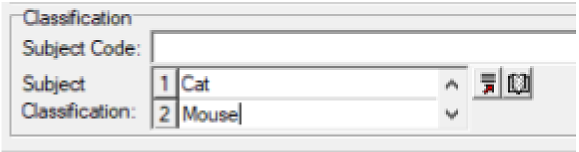
The field values for the following mapping are defined below:.

edarwincore.columnName	ecatalogue.columnName	*term:value*	*FMNH Mapping*
	DarClassification Code	*NONE*	*CULTURAL USE: DarClassificationCode - No DwC mapping* *edarwincore term:NONE* *= Pipe-delimited list ALL DesSubjectClassification_tab(n)*

Prompt	Column	Value
Name	StandardName	FMNH Standard Field

Version	StandardVersion	2021-03-29
Module	Module	ecatalogue
Term	StandardTerm	fmnh:classificationCode
Mapping Type	MappingType	Nested
Collapse Empty Values	CollapseEmptyValues	Yes
Remove Duplicate Values	RemoveDuplicateValues	Yes
Term Format String	TermFormatString	
Field	Field	DetSubjectClassification_tab
Defined Term	DefinedTerm	
~~Separator~~	~~FieldSeparator~~	
Format	Format	Uppercase
Row Inclusion	RowInclusion	All Rows
Separator (Nested)	NestedSeparator	| (Pipe)
Field (Where)	WhereField	
Condition	WhereCondition	
Value	Value	
Between Value	BetweenValue	

Using the above specification for



Classification

Subject Code:

Subject Classification: 1 Cat
2 Mouse

would result in the following output:

fmnh:classificationCode=CAT|MOUSE

Example 2

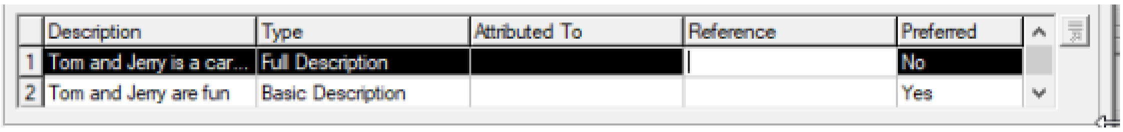
The field values for the following mapping are defined below:.

edarwincore.columnName	ecatalogue.columnName	*term:value*	*FMNH Mapping*
	DarDescription	*NONE*	*CULTURAL USE: DarDescription:Object Description - dc:description*

			edarwincore term:NONE = First DesKDescription0 where DesKType_tab = “Basic Description” and DesKPreferred_tab = “Yes”

Prompt	**Column**	**Value**
Name	StandardName	FMNH Standard Field
Version	StandardVersion	2021-03-29
Module	Module	ecatalogue
Term	StandardTerm	fmnh:description
Mapping Type	MappingType	Nested
Collapse Empty Values	CollapseEmptyValues	Yes
Remove Duplicate Values	RemoveDuplicateValues	Yes
Term Format String	TermFormatString	
Field	Field	DesKDescription0
Defined Term	DefinedTerm	
~~Separator~~	~~FieldSeparator~~	
Format	Format	Titlecase
Row Inclusion	RowInclusion	Where First
Separator (Nested)	NestedSeparator	
Field (Where)	WhereField (1)	DesKType_tab
Condition	WhereCondition (1)	Equals
Value	Value (1)	Basic Description
Between Value	BetweenValue (1)	
Field (Where)	WhereField (2)	DesKPreferred_tab
Condition	WhereCondition (2)	Equals
Value	Value (2)	Yes
Between Value	BetweenValue (2)	
Clause	WhereClause	where exists(DesKType_tab where DesKType = ‘Basic Description’) and exists(DesKPreferred_tab where DesKPreferred = ‘Yes’)

Using the above specification for



	Description	Type	Attributed To	Reference	Preferred
1	Tom and Jerry is a car...	Full Description			No
2	Tom and Jerry are fun	Basic Description			Yes

would result in the following output:

dc:description=Tom And Jerry Are Fun

Example 3

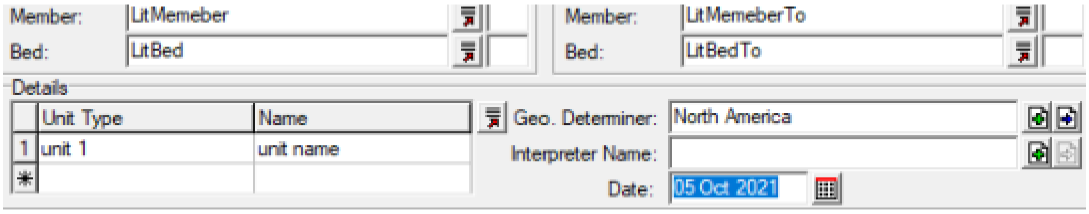
The field values for the following mapping are defined below:.

edarwincore.columnName	**ecatalogue.columnName**	***term:value***	***FMNH Mapping***
DarBed	DarBed	*bed*	*DarBed:Bed - bed* *edarwincore term:DarBed* *= PalStrStratigraphyRef_tab**.LitBedFrom_tab + “ - ” + PalStrStratigraphyRef_tab**.LitBedTo_tab + “|” + PalStrStratigraphyRef_tab**.PalStrDate0 PalStrStratigraphyRef_tab**.LitDate0 WHERE PalStrKind_tab = “Lithostratigraphy”*

Prompt	Column	Value
Name	StandardName	Darwin Core
Version	StandardVersion	2021-03-29
Module	Module	ecatalogue
Term	StandardTerm	dwc:bed
Mapping Type	MappingType	Nested
Collapse Empty Values	CollapseEmptyValues	
Remove Duplicate Values	RemoveDuplicateValues	
Term Format String	TermFormatString	%s - %s|%s
Field	Field (1)	PalStrStratigraphyRef_tab,(enhmstratigraphy).LitBedFrom_tab
Defined Term	DefinedTerm (1)	

~~Separator~~	~~FieldSeparator (1)~~	
Format	Format	Uppercase
Row Inclusion	RowInclusion	Where ~~All~~First
Separator (Nested)	NestedSeparator	
Field	Field (2)	PalStrStratigraphyRef_tab,(enhmstratigraphy).LitBedTo_tab
~~Separator~~	~~FieldSeparator (2)~~	~~- (dash)~~
Format	Format	Uppercase
Field	Field (3)	PalStrStratigraphyRef_tab,(enhmstratigraphy).LitDate0
~~Separator~~	~~FieldSeparator (3)~~	~~| (pipe)~~
Format	Format	ISO-8601
Field (Where)	WhereField	PalStrKind_tab
Condition	WhereCondition	Equals
Value	Value	Lithostratigraphy
Between Value	BetweenValue	
Clause	WhereClause	where exists(PalStrKind_tab where PalStrKind = ‘Lithostratigraphy’)

Using the above specification for



Member: LitMemeber

Bed: LitBed

Member: LitMemeberTo

Bed: LitBedTo

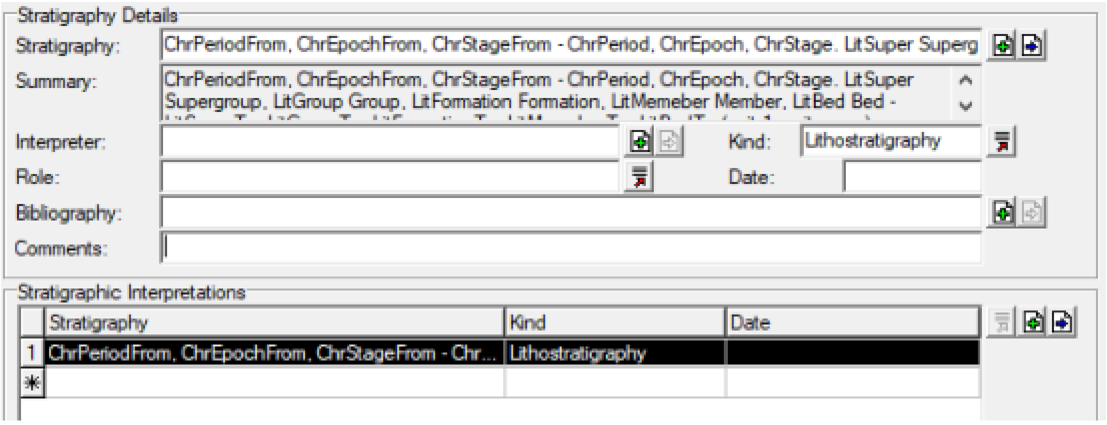
Details

	Unit Type	Name
1	unit 1	unit name
✳		

Geo. Determiner: North America

Interpreter Name:

Date: 05 Oct 2021



Stratigraphy Details

Stratigraphy:	ChrPeriodFrom, ChrEpochFrom, ChrStageFrom - ChrPeriod, ChrEpoch, ChrStage. LitSuper Superg
Summary:	ChrPeriodFrom, ChrEpochFrom, ChrStageFrom - ChrPeriod, ChrEpoch, ChrStage. LitSuper Supergroup, LitGroup Group, LitFormation Formation, LitMemeber Member, LitBed Bed -
Interpreter:	
Kind:	Lithostratigraphy
Role:	
Date:	
Bibliography:	
Comments:	

Stratigraphic Interpretations

	Stratigraphy	Kind	Date
1	ChrPeriodFrom, ChrEpochFrom, ChrStageFrom - Chr...	Lithostratigraphy	
✱			

would result in the following output:

dwc:bed=LITBED - LITBEDTO|2021-10-05

Example 4

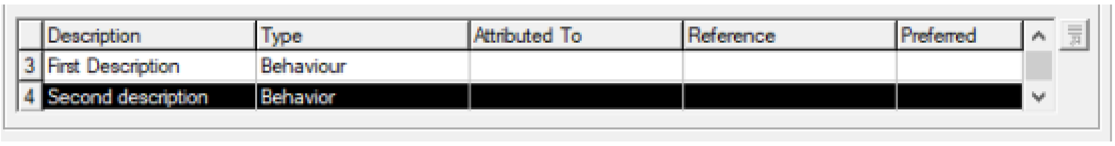
The field values for the following mapping are defined below:.

edarwincore.columnName	ecatalogue.columnName	*term:value*	*FMNH Mapping*
DarBehavior		*behavior*	*DarBehavior: Behaviour - dwc:behavior edarwincore term:DarBehavior = DesKDescription0(n) where DesKType_tab(n) = "Behaviour" or "Behavior"*

Prompt	Column	Value
Name	StandardName	Darwin Core
Version	StandardVersion	2021-03-29
Module	Module	ecatalogue
Term	StandardTerm	dwc:behaviour
Mapping Type	MappingType	Nested
Collapse Empty Values	CollapseEmptyValues	Yes
Remove Duplicate Values	RemoveDuplicateValues	Yes
Term Format String	TermFormatString	
Field	Field	DesKDescription0
Defined Term	DefinedTerm	
~~Separator~~	~~FieldSeparator~~	

Format	Format	
Row Inclusion	RowInclusion	Where All
Separator (Nested)	NestedSeparator	| (pipe)
Field (Where)	WhereField	DesKType_tab
Condition	WhereCondition	Equals
Value	Value	Behaviour Behavior
Between Value	BetweenValue	
Clause	WhereClause	where exists(DesKType_tab where DesKType = ‘Behaviour) or exists(DesKType_tab where DesKType = ‘Behavior’)

Using the above specification for



	Description	Type	Attributed To	Reference	Preferred
3	First Description	Behaviour			
4	Second description	Behavior			

would result in the following output:

dwc:behaviour=First description|Second description

Example 5

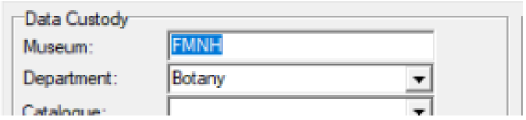
The field values for the following mapping are defined below:.

edarwincore.columnName	ecatalogue.columnName	*term:value*	*FMNH Mapping*
DarCatalogNumber	DarCatalogNumber	*catalogNumber*	*DarCatalogNumber: Catalog Number - dwc:catalogNumber* *edarwincore term:DarCatalogNumber* *If CatDepartment = Botany* *= ObjEcode* *Else* *= GeoSpecimenNo*

Prompt	Column	Value
Name	StandardName	Darwin Core

Version	StandardVersion	2021-03-29
Module	Module	ecatalogue
Term	StandardTerm	dwc:catalogNumber
Mapping Type	MappingType	Or
Collapse Empty Values	CollapseEmptyValues	
Remove Duplicate Values	RemoveDuplicateValues	
Term Format String	TermFormatString	
Field	Field (1)	ObjEcode
Defined Term	Defined Term (1)	
~~Separator~~	~~FieldSeparator (1)~~	
Format	Format	
Row Inclusion	RowInclusion	
Separator (Nested)	NestedSeparator	
Field	Field (2)	GeoSpecimenNo
Field (Where)	WhereField (1)	CatDepartment
Condition	WhereCondition (1)	Equals
Value	Value (1)	Botany
Between Value	BetweenValue (1)	
Clause	WhereClause (1)	where ColDepartment = ‘Botany’
Field (Where)	WhereField (2)	CatDepartment
Condition	WhereCondition (2)	Does Not Equal
Value	Value (2)	Botany
Between Value	BetweenValue (2)	
Clause	WhereClause (2)	where ColDepartment not = ‘Botany’

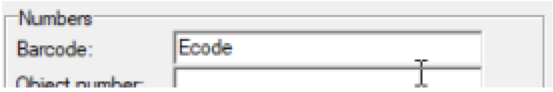
Using the above specification for


Data Custody

Museum: FMNH

Department: Botany

Catalogue:


Numbers

Barcode: Ecode

Object number:

would result in the following output:

dwc:catalogNumber=Ecode

Example 6

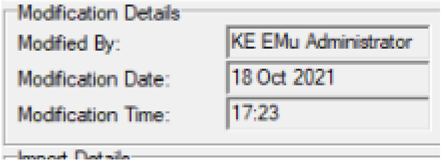
The field values for the following mapping are defined below:.

edarwincore.columnName	ecatalogue.columnName	*term:value*	*FMNH Mapping*
DarDCTermsModified	DarDateLastModified	*modified*	*DarDateLastModified:Date Last Modified - dc:modified* *edarwincore term:DarDCTermsModified?* *Format as ISO 8601 date & time (https://www.iso.org/standard/40874.html) : YYYY-MM-DD + "T" + hh:mm:ss.sss + "-06:00"* *= AdmDateModified*

Prompt	Column	Value
Name	StandardName	Darwin Core
Version	StandardVersion	2021-03-29
Module	Module	ecatalogue
Term	StandardTerm	dc:modified
Mapping Type	MappingType	Complex
Collapse Empty Values	CollapseEmptyValues	
Remove Duplicate Values	RemoveDuplicateValues	
Term Format String	TermFormatString	%sT%s-06:00
Field	Field (1)	AdmDateModified
Defined Term	Defined Term (1)	
Format	Format	ISO-8601

Row Inclusion	RowInclusion	
Separator (Nested)	NestedSeparator	
Field	Field (2)	AdmTimeModified
Format	Format	ISO-8601
Field (Where)	WhereField	
Condition	WhereCondition	
Value	Value	
Between Value	BetweenValue	

Using the above specification for



Modification Details

Modified By: KE EMu Administrator

Modification Date: 18 Oct 2021

Modification Time: 17:23

Import Details

would result in the following output:

dc:modified=2021-10-18T17:23:00.000-06:00

Example 7

The field values for the following mapping are defined below:.

edarwincore.columnName	ecatalogue.columnName	*term:value*	*FMNH Mapping*
DarDCTermsBibliographicCitation	DarCitation	*bibliographicCitation*	*DarCitation: Citation - dwc:bibliographicCitation edarwincore term:DarDCTermsBibliographicCitation? Dynamically build this based on FMNH Norms documentation. [occurrenceID].[catalogNumber].[dataset name]. [data publisher]. [link to dataset] (last modified [date YYYY-Month-DD]).* *DarGlobalUniqueIdentifier + DarCatalogNumber + DarRightsHolder* *If CatDepartment = "Zoology"* *If CatCatalog = "Amphibians and Reptiles"* *"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_herps"*

			If CatCatalog = “Birds” *“https://fmipt.fieldmuseum.org/ipt/resource?r=fm_birds”* *If CatCatalog = “”* *“https://fmipt.fieldmuseum.org/ipt/resource?r=fm_birds_eggs”* *If CatCatalog = “Fishes”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_fishes* *If CatCatalog = “Insects”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_insects* *If CatCatalog = “Invertebrate Zoology”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_invertebrates* *If CatCatalog = “Mammals”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_mammals* *Else* *“”* *If CatDepartment = “Botany”* *If CatCatalog = “Seed Plants”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_seedplants* *If CatCatalog = “Bryophytes”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_bryophyte* *If CatCatalog = “Fungi”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_fungi* *If CatCatalog = “Lichens”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_lichens* *If CatCatalog = “Pteridophytes”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_pteridophytes* *Else* *“”* *If CatDepartment = “Geology”* *If CatCatalog = “Fossil Invertebrates”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh-fossil-inverts* *If CatCatalog = “Paleobotany”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh-paleobotany* *If CatCatalog = “Fossil Birds” OR “Fossil Fishes” OR “Fossil Birds” OR “Fossil Mammals” OR “Fossil Synapsids” OR “Fossil Vertebrates”* *https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh-fossil-verts* *Else*

			“” *If CatDepartment = “Anthropology”* *“”* *+ “(last modified “ + FORMAT:YYYY-Month-DD DarDateLastModified +”)”* *Example: 012da259-4a53-4232-9ec5-8be1f25a40b1 FMNH243174 Field Museum of Natural History (Zoology) Invertebrate Collection. Field Museum. http://fmipt.fieldmuseum.org:8080/ipt/resource.do?r=fmnh_invertebrates (last modified 2014-July-28)*

Prompt	Column	Value
Name	StandardName	Darwin Core
Version	StandardVersion	2021-03-29
Module	Module	ecatalogue
Term	StandardTerm	dwc:bibliographicCitation
Mapping Type	MappingType	~~Complex~~Script
Collapse Empty Values	CollapseEmptyValues	
Remove Duplicate Values	RemoveDuplicateValues	
Term Format String	TermFormatString	
Field	Field (1)	
Defined Term	DefinedTerm (1)	dwc:occurenceID
~~Separator~~	~~FieldSeparator (1)~~	
Row Inclusion	RowInclusion	
Separator (Nested)	NestedSeparator	-
Field	Field (2)	
Defined Term	DefinedTerm (2)	dwc:catalogNumber
Field	Field (3)	
Defined Term	DefinedTerm (3)	dwc:rightsHolder
Field	Field (4)	CatDepartment

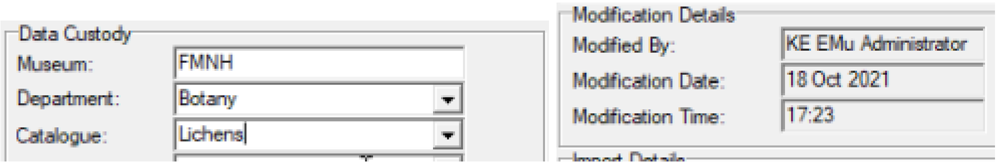
Defined Term	DefinedTerm (4)	
Format	Format	Titlecase
Field	Field (5)	CatCatalog
Defined Term	DefinedTerm (5)	
Format	Format	Uppercase
Field	Field (6)	AdmDateModified
Defined Term	DefinedTerm (6)	~~dwc:modified~~
Format	Format	YMonthD
Field (Where)	WhereField	
Condition	WhereCondition	
Value	Value	
Between Value	BetweenValue	

For this ~~Complex~~Script definition the Script field would be used to generate the correct output: Code similar to the following could be used:

```
my $id = $fields->[0];
my $number = $fields->[1];
my $rights = $fields->[2];
my $department = $fields->[3];
my $catalog = $fields->[4];
my $time = $fields->[5];

my $ref = getRef($department, $catalog);
return "$id $number $rights $ref (lastmodified $time)";

sub
getRef
{
        my $department = shift;
        my $catalog = shift;

        if ($department =~ /Zoology/i)
        {
                return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_herps" if ($catalog
=~ /Amphibians and Reptiles/i);
                return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fm_birds_eggs" if
($catalog =~ /Birds Eggs/i);
                return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fm_birds" if ($catalog
=~ /Birds/i);
                return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_fishes" if
($catalog =~ /Fishes/i);
                return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_insects" if
($catalog =~ /Insects/i);
                return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_invertebrates" if
($catalog =~ /Invertebrate Zoology/i);
                return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_mammals" if
($catalog =~ /Mammals/i);
                return "";
        }
        if ($department =~ /Botany/i)
        {
                return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_seedplants" if
($catalog =~ /Seed Plants/i);
```

```
            return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_brypphyte" if
($catalog =~ /Bryophytes/i);
            return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_fungi" if ($catalog
=~ /Fungi/i);
            return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_lichens" if
($catalog =~ /Lichens/i);
            return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_pteridophytes" if
($catalog =~ /Pteridophytes/i);
            return "";
      }
      if ($department =~ /Geology/i)
      {
            return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh-fossil-invertebrate
s" if ($catalog =~ /Fossil Invertebrates/i);
            return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh-paleobotany" if
($catalog =~ /Paleobotany/i);
            return
"https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh-fossil-verts" if
($catalog =~ /Fossil Birds/i || $catalog =~ /Fossil Fishes/i ||
$catalog =~ /Fossil Birds/i || $catalog =~ /Fossil Mammals/i ||
$catalog =~ /Fossil Synapsids/i || $catalog =~ /Fossil Vertebrates/i);
            return "";
      }
      return "";
}
```

Using the above specification for



Data Custody
Museum: FMNH
Department: Botany
Catalogue: Lichens

Modification Details
Modified By: KE EMu Administrator
Modification Date: 18 Oct 2021
Modification Time: 17:23

would result in the following output:

dwc:bibliographicCitation=012da259-4a53-4232-9ec5-8be1f25a40b1 FMNH123456 Field Museum of Natural History (Botany) Lichens Collection. Field Museum. http://fmipt.fieldmuseum.org:8080/ipt/resource?r=fmnh_lichens (last modified 2021-October-18)

Example 8

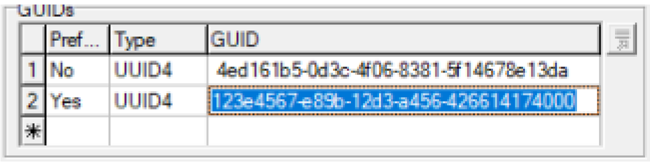
The field values for the following mapping are defined below:

edarwincore.columnName	**ecatalogue.columnName**	***term:value***	***FMNH Mapping***
DarHigherGeographyID		*higherGeographyID*	*DarHigherGeographyID: Higher Geography ID - dwc:higherGeographyID edarwincore term:DarHigherGeographyID = ColCollectionEventRef->ColSiteRef->PolParentRef.AdmGUIDValue_tab WHERE AdmGUIDType_tab = UUID4 AND AdmGUIDIsPreferred_tab = No*

Prompt	**Column**	**Value**
Name	StandardName	Darwin Core
Version	StandardVersion	2021-03-29
Module	Module	ecatalogue
Term	StandardTerm	dwc:higherGeographyID
Mapping Type	MappingType	Nested
Collapse Empty Values	CollapseEmptyValues	Yes
Remove Duplicate Values	RemoveDuplicateValues	Yes
Term Format String	TermFormatString	
Field	Field	ColCollectionEventRef(ecollectionevents).ColSiteRef(esites).PolParentRef(esites).AdmGUIDValue_tab
Defined Term	DefinedTerm	
~~Separator~~	~~FieldSeparator~~	
Format	Format	
Row Inclusion	RowInclusion	Where ~~All~~First
Separator (Nested)	NestedSeparator	| (pipe)
Field (Where)	WhereField (1)	AdmGUIDType_tab
Condition	WhereCondition (1)	Equals

Value	Value (1)	UUID4
Between Value	BetweenValue (1)	
Clause	WhereClause (1)	where exists(AdmGUIDType_tab where AdmGuidType = ‘UUID4’)
Field (Where)	WhereField (2)	AdmGUIDIsPreferred_tab
Condition	WhereCondition (2)	Equals
Value	Value (2)	No
Between Value	BetweenValue (2)	
Clause	WhereClause (2)	where exists(AdmGUIDIsPreferred_tab where AdmGuidIsPreferred = ‘No’)

Using the above specification for



GUIDs

	Pref...	Type	GUID
1	No	UUID4	4ed161b5-0d3c-4f06-8381-5f14678e13da
2	Yes	UUID4	123e4567-e89b-12d3-a456-426614174000
✱			

would result in the following output:

dwc:higherGeographyID=4ed161b5-0d3c-4f06-8381-5f14678e13da

Example 9

The field values for the following Audubon Core mapping are defined below:.

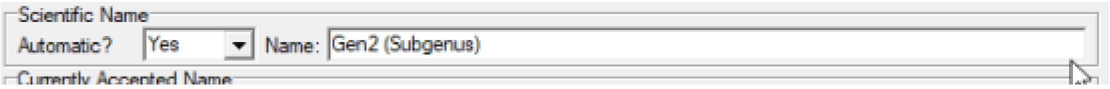
edarwincore.columnName	**emultimedia.columnName**	***term:value***	***FMNH Mapping***
	AudTaxonCoverage	ac:taxonCoverage	Maps to ac:taxonCoverage - http://rs.tdwg.org/ac/terms/taxonCoverage If ObjObjectsRef_tab is NULL and TaxTaxonomyRef_tab is NOT NULL (e.g., irn 1413162): Then AudTaxonCoverage = TaxTaxonomyRef_tab.SummaryData *(Pipe delimited if Multimedia is attached to*

			multiple Taxonomy records) ELSE AudTaxonCoverage = ObjObjectsRef_tab.DarScientificName *(Pipe delimited if Multimedia is attached to multiple Catalog records, and remove duplicates)*

Prompt	Column	Value
Name	StandardName	Audubon Core
Version	StandardVersion	2020-01-27
Module	Module	emultimedia
Term	StandardTerm	ac:taxonCoverage
Mapping Type	MappingType	Nested Or
Collapse Empty Values	CollapseEmptyValues	Yes
Remove Duplicate Values	RemoveDuplicateValues	Yes
Term Format String	TermFormatString	
Field	Field	ecatalogue:MulMultiMediaRef_tab.IdeTaxonRef_tab.(etaxonomy).ClaScientificName
Defined Term	DefinedTerm	
~~Separator~~	~~FieldSeparator~~	
Format	Format	Uppercase
Row Inclusion	RowInclusion	All Rows
Separator (Nested)	NestedSeparator	| (pipe)
Field (Where)	WhereField	(ecatalogue)MulMultiMediaRef_tab
Condition	WhereCondition	Equals
Value	Value	irn
Between Value	BetweenValue	
Clause	WhereClause	where exists(ecatalogue.MulMultiMediaRef_tab where MulMultiMediaRef = emultimedia.irn)

Field	Field	etaxonomy:MulMultiMediaRef_tab.SummaryData
Defined Term	DefinedTerm	
~~Separator~~	~~FieldSeparator~~	
Format	Format	Uppercase
Row Inclusion	RowInclusion	All Rows
Separator (Nested)	NestedSeparator	| (pipe)
Field (Where)	WhereField	(etaxonomy)MulMultiMediaRef_tab
Condition	WhereCondition	Equals
Value	Value	irn
Between Value	BetweenValue	
Clause	WhereClause	where exists(etaxonomy.MulMultiMediaRef_tab where MulMultiMediaRef = emultimedia.irn)

Using the above specification (which matched one record) for


Scientific Name
Automatic? Yes
Name: Gen2 (Subgenus)
Currently Accepted Name

would result in the following output:

ac:taxonCoverage=GEN 2 (SUBGENUS)

Scripts:

Where a script is specified within a term definition, the script will perform all formatting and any format specified in Term Format String will be ignored. Some thought was given to retaining the Term Format String and allowing a script to be specified per field definition but this was deemed to be too difficult to display in a meaningful format and difficult for users to understand. As such a single script is the preferred approach, which, when executed, will return a formatted string.

Questions:

1. What should happen with the FMNH specific definitions?

They will form their own standard definition.

2. Please clarify mapping for dwc:bed = PalStrDate0 or LitDate0?

   This should be LitDate0.

3. What should happen when a nested table is linked?

   I can't remember the purpose of this question nor the response. Does anyone else remember this?

   Where a nested table exists in a definition, it follows Sharon's path 1). The fields will be concatenated using the Nested Separator. Where an extension is used Sharon's path 2) applies. The fields are exported in a separate file with a key/foreign key pair to associate the nested table back to the main record.

4. What should happen when there is more than one row match? Plan is to join rows with a pipe which may be a field join.

   Where more than one row match join the rows with the Nested Separator (use | as the default if nothing is specified).

Expand existing export process to allow for defining a standards export

Description

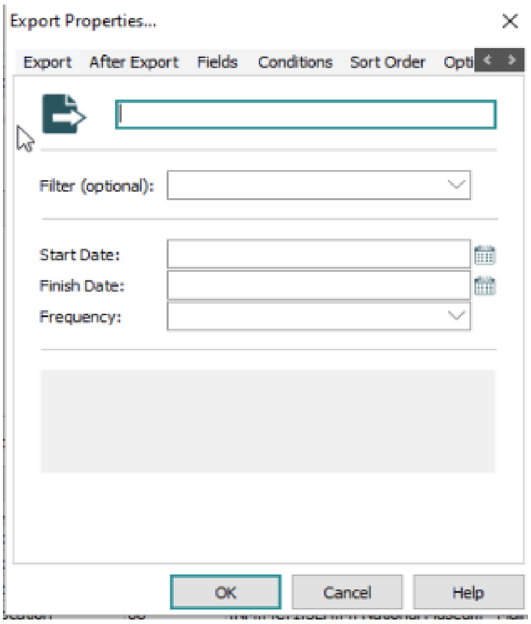
The existing export process comprises two components. The first is handled by the EMu client and defines the export and when it is run. The second is handled by the EMu server which generates the export at the appropriate time.

Rather than outlining the entire process here the following link is to the EMu help which covers the process required to define an export. http://help.emu.axiell.com/latest/en/Topics/Common/How%20to%20create%20a%20Scheduled%20Export.htm?Highlight=Export%20

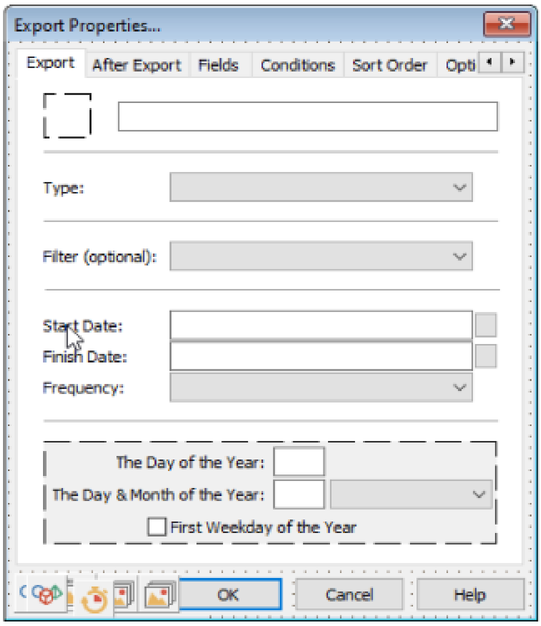
This process provides a significant amount of the functionality required to define and export for a standard. The difference between the two is really that export merely takes data for defined fields and outputs it whereas a standards export takes data and formats it according to a definition for the standard.

As such, adopting the existing export definition starting point as the starting point for defining a standards export makes good sense. An additional field is required to define the type of export which will then allow different information collection tabs to be displayed.

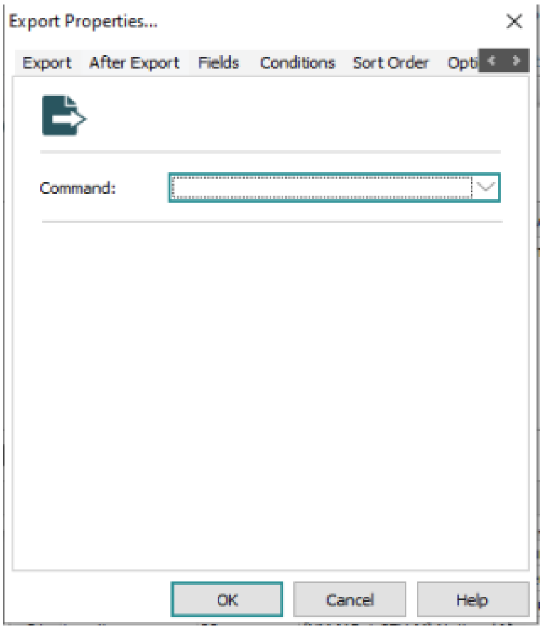
Existing tab



New tab

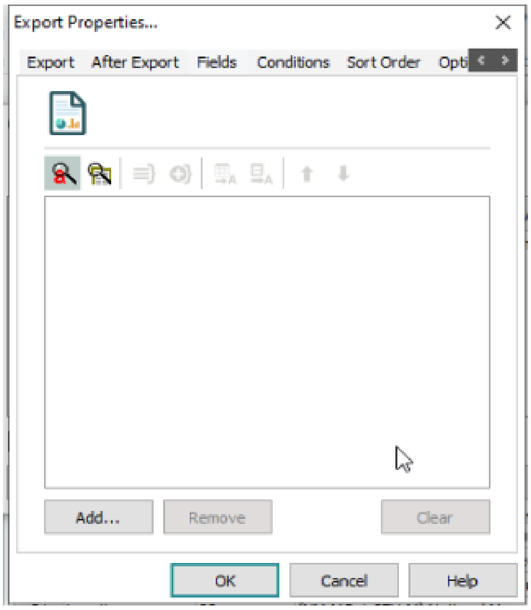


The ability to post process an export is required to handle any special formatting or data tidy up/exclusion that may be required prior to a standard export being released. The existing After Export tab will be retained for this.

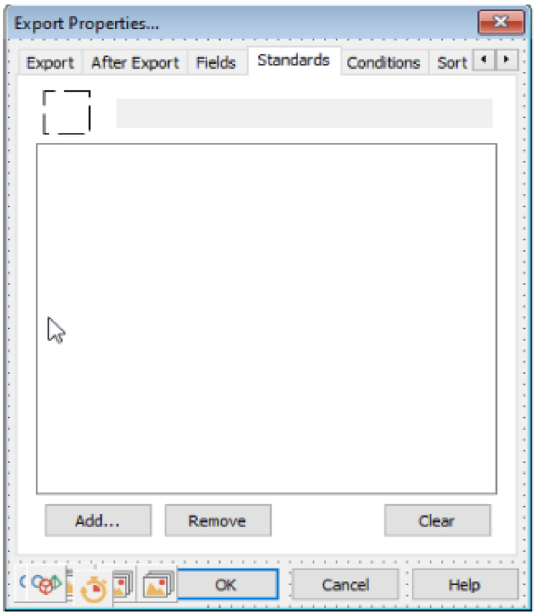


It is necessary to tie the standards export to one or more standards definitions in the standards module. This is in essence choosing the fields you want to appear in the export. When a standards export is being defined, the Fields tab will be replaced with a Standards tab.

Existing tab

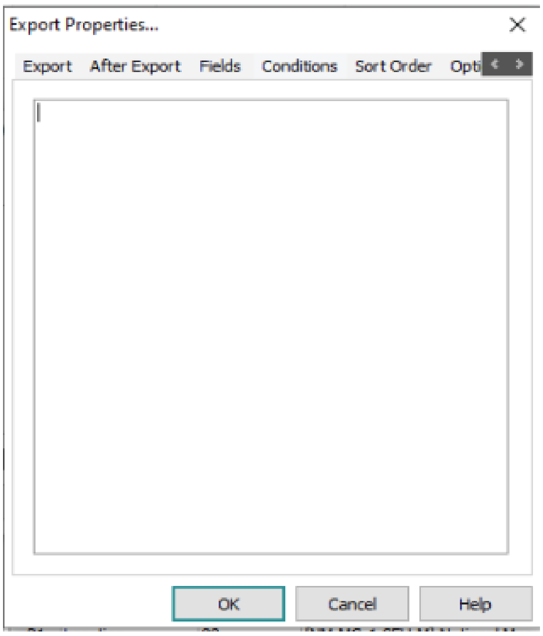


New tab

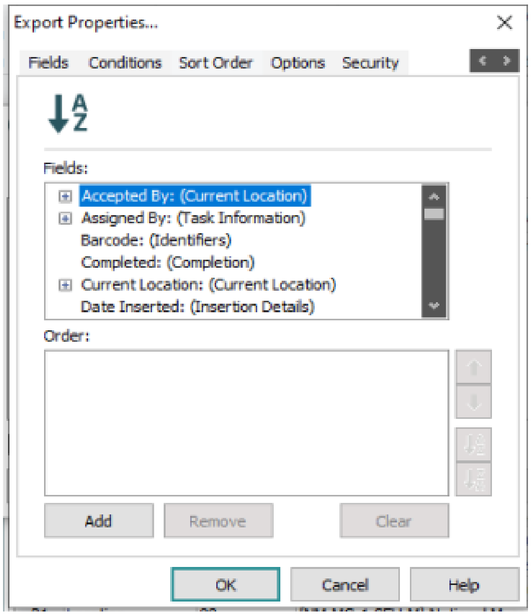


The tab will allow for the addition of individual standard definitions and extensions to be added to generate the complete standards export

The ability to restrict the export to a set of records based on a condition is required. The existing Conditions tab provides this functionality and so will be used.



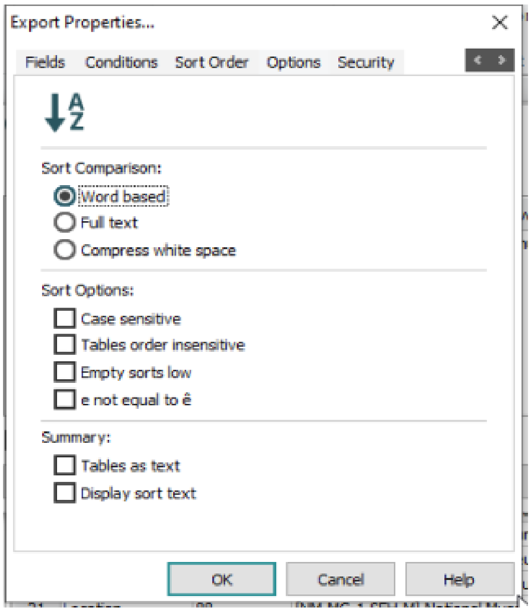
The ability to output the records within an export in a particular order is also required. The existing Sort Order tab provides this functionality and will be used. Each standards export may contain many complex relationships between fields and so it may be determined that the existing sort functionality does not adequately cover what is required here.



Fields:

- ⊞ Accepted By: (Current Location)
- ⊞ Assigned By: (Task Information)
- Barcode: (Identifiers)
- Completed: (Completion)
- ⊞ Current Location: (Current Location)
- Date Inserted: (Insertion Details)

Order:



Sort Comparison:

● Word based
○ Full text
○ Compress white space

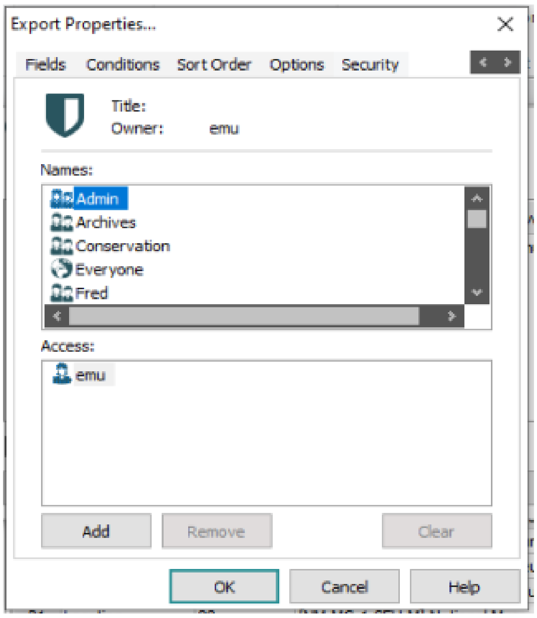
Sort Options:

□ Case sensitive
□ Tables order insensitive
□ Empty sorts low
□ e not equal to ê

Summary:

□ Tables as text
□ Display sort text

The ability to define who can update or run a standards export is also required. The existing Security tab provides this functionality and will be used.



Title:
Owner: emu

Names:
Admin
Archives
Conservation
Everyone
Fred

Access:
emu

Additional Functionality

The existing functionality which stores an export as a record in the Exports module must be retained for all standards exports.

Questions:

1. Will these exports ever be set on a scheduled basis or is it anticipated that they will always be extracted manually?
   It is expected that the exports will be run using both mechanisms so both must be supported.
2. When defining an export is it always just a Standard or a Standard and an Extension? Is it ever part of one e.g. take 20 fields from Darwin Core? If so, should that be defined as a separate standard or should those fields be able to be individually picked?
   An export will always be the full definition. If a subset is needed a new definition can be created.
3. If the answer to 1 above is Yes, Is it necessary to provide a means where a condition can be specified to limit the records added to an export e.g. records updated in the last month that have the Publish On Internet field set to Yes?
   An export should be able to be run with or without a condition. If a condition is the usual way for running the export it should be specified as part of the export. A manual export can always be run with or without a condition. A condition is defined to be a query restriction.
4. Is it necessary to be able to define a sort order for the export? Should the sort be allowed to be defined on any field in the export including those where data is constructed from complex field conditions?
   An export should be able to be run with or without a sort. If a sort is the usual way for running the export it should be specified as part of the export. A manual sort can always be run on a set of records prior to the export.