

## Field Validator

## Field Validator

The Field Validator plug-in is used to validate descriptive metadata according to the metadata profile setup. It uses regular expression syntax to define the pattern to be validated.

## Plug-in Parameters

Plug-in parameters are set while installing a new instance of the plug-in and include the following:

- Regular Expression – Mandatory free text parameter to type the regular expression pattern to be validated

Plug-in Information			
Plug-In Id	19256981	Plug-In Name	ChecksumDigitValidatorPlugin
Plug-In Type Name	FieldValidatorPlugin	Implementation Type	JAVA
Implementation Name	com.exlibris.dps.repository.pl...	Interface	-
Owner	CRS00	Material Type	DIGITAL
Family Type	TASK	Description	RegexpFieldValidator
Resource Type	-	Module	-
Status	ACTIVE	Plug-In Version	3.0
Public API	-		

  

Contact Information			
Contact Type	admin	Email address	shaib@exlibris.co.il
Last Name	Ltd	First Name	Exlibris
Address 1	Agodat Asport2 Building 9	City	Jerusalem
Telephone 1	(050)746-1234		

  

Plug-in Parameters	
* Regular Expression	.*

### Field Validator Plug-in Details

## Usage

A list of the implemented Field Validator plug-ins will be available as validation routines in the Metadata Form and Metadata Profile – DC Element Editor pages.

The screenshot shows the 'Metadata Form Details Page' for a field named 'Publisher'. At the top, it displays metadata: Type: Generic, Created on: 31/01/2011, Updated on: 23/05/2021, and Created by: admin1. The main area is divided into 'Design Information' and 'Fields Information'. Under 'Fields Information', there's a table to 'Select a field to view/edit:' with 8 rows of fields like Title, Alternative T1, Publisher, Creation Date, Description, Is Valid, Subject, and Type. To the right, 'Properties for Publisher' includes fields for Field Name (Publisher), DC Tag (Publisher (DC)), Mandatory (No), Single Line (No), Default Value, Tooltip (Creator), Label (Creator), and Code Table Name (Locally Defined). Below this is an 'Options Fields' section with a table of values and labels, and an 'Add New Option' button. At the bottom, there are 'Preview', 'Cancel', and 'Save Form' buttons.

## Metadata Form Details Page

## Implementations

Rosetta includes the following implementations of the Field Validator plug-in:

- EmailValidator – Regular expression to validate email template
- ISBNValidator – Regular expression to validate ISBN template
- ISMNValidator – Regular expression to validate ISMN template
- ISSNValidator – Regular expression to validate ISSN template
- NumericValidator – Regular expression to validate numeric values
- PhoneValidator – Regular expression to validate phone template
- Regular\_Expression\_Validator\_Plugin – A generic regular expression plug-in
- URLValidator – Regular expression to validate URL template
- W3CDTFValidator – Regular expression to validate W3CDTF (W3C Date and Time Format) template

The screenshot shows the 'Plug-In Management' page. It has tabs for 'Bundled' and 'Custom'. A filter is set to 'Field Validator Plugins'. Below the filter is a search bar and a 'Go' button. The main area shows a table of installed plugins with columns for Active, Plug-In Type Name, Plug-In Name, Description, Contact Person, and Plug-In Version. There are 100 records displayed, and the current view is '1 - 2 of 2 Plug-In'.

Active	Plug-In Type Name	Plug-In Name	Description	Contact Person	Plug-In Version		
<input checked="" type="checkbox"/>	FieldValidatorPlugin	ChecksumDigitValidatorPlugin	RegexFieldValidator	Exlibris Ltd	3.0	View	Edit Delete
<input checked="" type="checkbox"/>	FieldValidatorPlugin	Regular_Expression_Validator_Plugin	RegexFieldValidator	Exlibris Ltd	2.0	View	Edit Delete

## Field Validator Plug-in List

The Field Validator plug-in can be used to validate descriptive metadata according to the relevant metadata profile setup. It uses regular expression syntax to define the pattern to be validated.

There is currently one out-of-the-box implementation, `RegexFieldValidatorPlugin`, for the Field Validator plug-in.

`RegexFieldValidatorPlugin` receives `regex` as a parameter and uses it to validate a field during runtime.

This implementation enables customers to create different plug-in instances using custom regular expressions, without having to write plug-in implementations themselves.