
Technical Note

Field Validation Using Regular Expressions

Overview

In Rep++, values typed by an end user were usually validated using an input mask. However, you can now use a regular expression.

This article describes how to use regular expressions to validate a Rep++ field.

The products and software mentioned in this document are trademarks, registered trademarks or trade names of their respective holders.

Field Validation Using Regular Expressions

One way to validate the value of field entered by an end user in an application is to use a regular expression. In Rep++, regular expressions can apply to a field regardless of where it is used, or to a field used in a specific Rowset. In the former case, the regular expression is set in the Field editor, in the latter it is set in the Rowset editor.

Regular expressions can be used on any Rep++ field type except Formatted String, where it is replaced by the Input Mask.

You will use the contact management application to illustrate how to validate a Rep++ field using a simplified regular expression.

NOTE: Make sure your regular expression is valid. The Rep++ regular expression engine is based on the Perl 5 syntax.

Adding an e-mail field to the contact management application

You will create a new field, EMAIL, that will be validated with a simple regular expression. The field will be included in the CLIENT Rowset. You will then create an application to test it. The regular expression is `^\w+@[a-zA-Z]+\.[a-zA-Z]{2,3}$` and is described as follows.

Method	Description
<code>^\w+</code>	Matches one or more occurrences of word characters (letters and digits) at the beginning of a line.
<code>@</code>	Matches a single instance of itself.
<code>[a-zA-Z_]+</code>	Matches one or more occurrences of any letter or underscore.
<code>\.</code>	Matches a single instance of ".".
<code>[a-zA-Z]{2,3}\$</code>	Matches two or three letters at the end.

To add an EMAIL field that is validated with a regular expression

1. Open the DEMO system in Rep++ studio.
2. Create a new field named EMAIL (to simplify the process, do not attach the field to a database column).
3. Set its attributes as follows:
 - o Type: to Alpha String
 - o Control Type: Edit box
 - o Minimum Size: 6
 - o Maximum Size: 80
4. Add the newly created EMAIL field to the CLIENT Rowset.
5. Do one:

- To add the validation at the field level: go to the Field editor, select the EMAIL field and set the regular expression to `^\w+@[a-zA-Z_]+\.[a-zA-Z]{2,3}$`.
- To add the validation at the Rowset level: go to the Rowset editor, select the CLIENT Rowset, select the EMAIL field and set its regular expression to `^\w+@[a-zA-Z_]+\.[a-zA-Z]{2,3}$`.

The screenshot shows a software interface for defining a field. The 'Field' is named 'EMAIL'. The 'Type' is 'AL Alpha string' and the 'Control Type' is '10 Edit box'. The 'Regular Expression' field is populated with the pattern `^\w+@[a-zA-Z_]+\.[a-zA-Z]{2,3}$`. Other fields include 'System' (DEMOV2), 'Section' (\$ROOT.), 'Column', 'Description', 'Alternate Name', 'SQL Command for List', 'Size' (Minimum: 6, Maximum: 80), and 'Value' (Minimum and Maximum).

Figure 1. The EMAIL field in the Field editor.

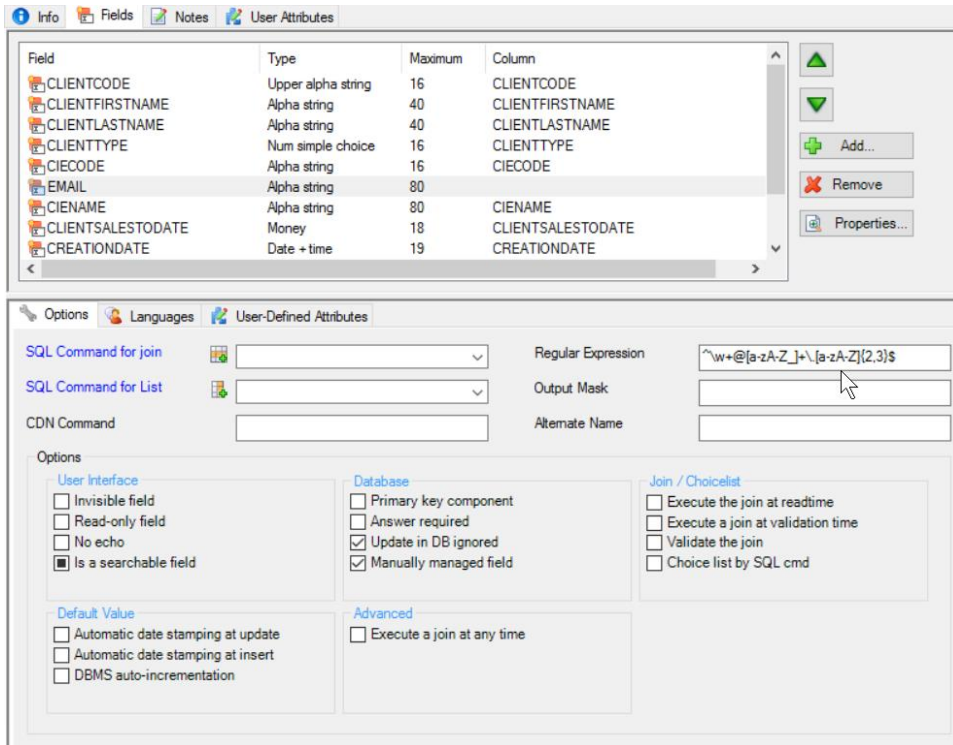


Figure 2. The EMAIL field options in the CLIENT Rowset editor.

To quickly test the behaviour of the EMAIL field, use the Rep++ wizard to generate a Windows® or Web application and try entering different values for the EMAIL field. The error occurs when you save.

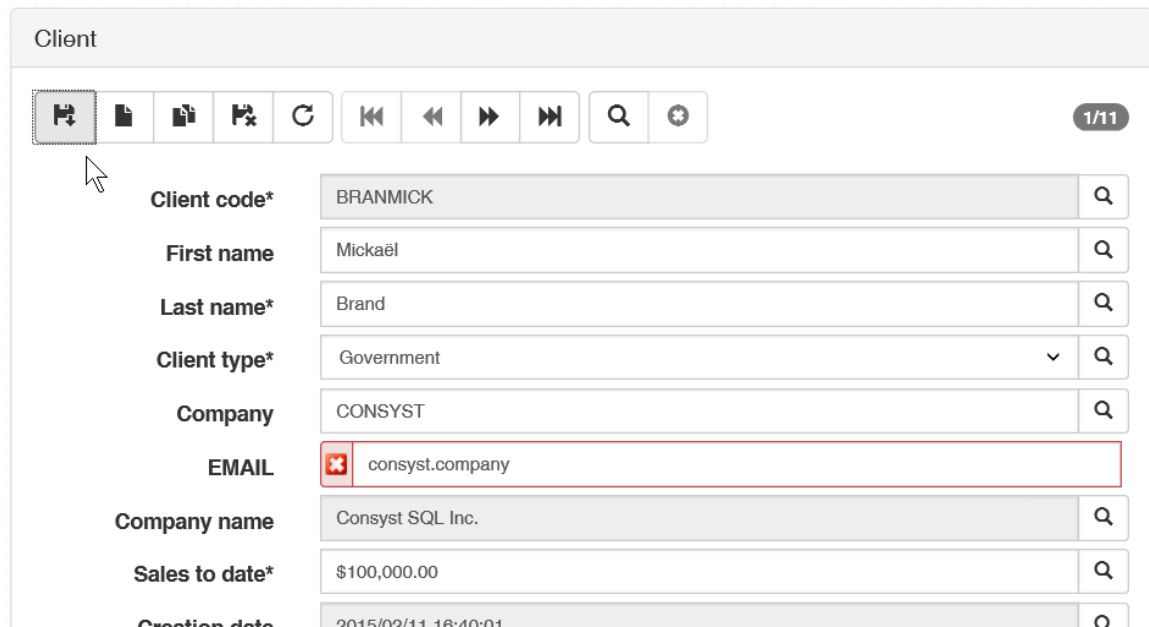


Figure 3. Regular expression validation at work – invalid e-mail.

The screenshot shows a web form titled "Client" with a toolbar at the top containing icons for home, save, print, delete, refresh, and navigation. A "1/11" indicator is in the top right. The form fields are as follows:

Client code*	BRANMICK	Q
First name	Mickaël	Q
Last name*	Brand	Q
Client type*	Government	Q
Company	CONSYST	Q
EMAIL	hello@consyst.com	
Company name	Consyst SQL Inc.	Q
Sales to date*	\$100,000.00	Q
Creation date	2015/09/11 18:40:01	Q

Figure 4. Regular expression validation at work – valid e-mail.

References

The simplified regular expression for email address validation was taken from the online regular expression library <http://regexlib.com/>. There are plenty of resources and tutorials on the Web devoted to regular expressions.

[http://msdn2.microsoft.com/en-us/library/2k3te2cs\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/2k3te2cs(VS.80).aspx)

<http://regexlib.com/>

<http://www.regular-expressions.info/>

<http://www.codeproject.com/dotnet/RegexTutorial.asp>