
Technical Note

Field Validation Using Regular Expressions

Author(s): R&D Department

Publication date: December 8, 2006

Revision date: May 17, 2010



Field Validation Using Regular Expressions

Overview

In REP++, field values were usually validated using an input mask. However, you can now use a regular expression instead of an input mask.

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

Adding an EMAIL field to the contact management application

You will use the contact management application to illustrate how to validate a REP++ field using a simplified 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. Add the newly created EMAIL field to the CLIENT group.
4. To validate the EMAIL field, you can specify a regular expression either at the field level (i.e. the validation will be applied in all groups that use this field) or at the group level (i.e. the validation will be applied only in the group where it has been specified).
 - To add the validation at the field level, go to the EMAIL field editor and set the Regular Expression text to `^\w+@[a-zA-Z_]+\.[a-zA-Z]{2,3}$`.
 - To add the validation at the group level, go to the CLIENT group editor, select the EMAIL field and set the Regular Expression text to `^\w+@[a-zA-Z_]+\.[a-zA-Z]{2,3}$`.

The regular expression 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. |

Technical Note

The screenshot shows the 'Field editor' window for the 'EMAIL' field. The 'System*' field is set to 'DEMO', 'Section*' to '\$ROOT', and 'Field*' to 'EMAIL'. The 'Column' dropdown is set to 'EMAIL'. The 'Type*' is 'AL Alpha string', and the 'Control Type*' is '10 Edit box'. The 'Regular Expression' field contains the pattern '~w+@[a-zA-Z_]+\.[a-zA-Z]{2,3}\$'. The 'Minimum' size is 0 and the 'Maximum' size is 256. The 'Decimals' field is empty. The 'Output Mask' and 'CDN Command' fields are also empty.

Figure 1. The EMAIL field in the Field editor.

The screenshot shows the 'CLIENT Rowset editor' window. The 'Fields' tab is active, displaying a list of fields with their types and maximum lengths. The 'EMAIL' field is highlighted. The 'Options' tab is also visible, showing the 'Regular Expression' field with the pattern '~w+@[a-zA-Z_]+\.[a-zA-Z]{2,3}\$'. The 'Options' section contains several checkboxes:

| Option | Checked |
|--------------------------------------|-------------------------------------|
| Invisible field | <input type="checkbox"/> |
| Field not accessible | <input type="checkbox"/> |
| Default answer fixed | <input type="checkbox"/> |
| No echo | <input type="checkbox"/> |
| Answer required | <input type="checkbox"/> |
| Primary key component | <input type="checkbox"/> |
| Update in DB ignored | <input checked="" type="checkbox"/> |
| Not a DB field | <input checked="" type="checkbox"/> |
| Choice list by SQL cmd (1 time) | <input type="checkbox"/> |
| Choice list by SQL cmd (x times) | <input type="checkbox"/> |
| Execute a join at validation time | <input type="checkbox"/> |
| Execute a join at any time | <input type="checkbox"/> |
| Validate the join | <input type="checkbox"/> |
| Automatic date stamping at update | <input type="checkbox"/> |
| Automatic date stamping at insertion | <input type="checkbox"/> |
| Default answer by SQL cmd | <input type="checkbox"/> |
| Field out of rowset | <input type="checkbox"/> |
| User-defined flag | <input type="checkbox"/> |
| DBMS auto-incrementation | <input type="checkbox"/> |

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

5. 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.

Technical Note

The screenshot shows the 'Client Editor' interface. On the left is a table of clients. The main form contains fields for Client Code, First Name, Last Name, EMAIL, Client Type, Sales To Date, Company Code, Creation Date, and Modification Date. The EMAIL field contains 'Me@sssss' and has a red error message to its right. An orange arrow points to this error message.

| Client Code | First Name | Last Name |
|-------------|------------|-----------|
| BRANMICK | Mickaël | Brand |
| CHIRMARI | | Chirac |
| CLAUDAN | Clauss | Dan |
| FERRALDO | Aldo | Ferrarini |
| FLEUSYLV | Sylvain | Fleury |
| JEANMARI | Mario | Jean |
| LULUCY | Luoy | Lu |
| STRAAAR | Aaron | Strauss |
| THIEANTH | Anthony | Thierry |
| TREMPIER | Pierre | Tremblay |

| Address Code | Address #1 | Address #2 | City | Postal Code | Province/State |
|--------------|-------------------|------------|----------|-------------|----------------|
| 8 | 2525 March Fields | | Winnipeg | G4H4G4 | Alberta |
| 15 | 120 rue Latulipe | | Ottawa | H3A3C8 | Ontario |

Figure 3. Regular expression validation at work – invalid email. A warning appears to the right of the invalid field.

The screenshot shows the 'Client Editor' interface with the same client list as Figure 3. In this view, the EMAIL field contains 'Me@sssss.com' and there is no error message. The rest of the form and tables are identical to Figure 3.

Figure 4. Regular expression validation at work – valid email.

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>