Removing “commands” from subject line commands regardless of spelling

iQ.Suite Tips & Tricks for IBM Domino

You are using a rule with a special command, e.g. “”.

(see Tips and Tricks December 2009)

'Command Rule'

iQ.Suite Text Rule

<table>
<thead>
<tr>
<th>Basics</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Command Rule</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Active, Not active</td>
</tr>
<tr>
<td><strong>Log level</strong></td>
<td>Loglevel 0 (value from NOTES.IN)</td>
</tr>
<tr>
<td><strong>Execution mode</strong></td>
<td>Dependent on job or global parameter</td>
</tr>
<tr>
<td><strong>Text fields</strong></td>
<td>Subject</td>
</tr>
<tr>
<td><strong>Text pattern</strong></td>
<td>&quot;&lt;command&gt;&quot;</td>
</tr>
</tbody>
</table>

But users won’t always write it the way you ask them to. For example, users may write “”, “”, “” or under some circumstances even “”.

Send   Send and File... Save as Draft   Delivery Options...   Display   More...

To: Group Admin/iQSUITE.
Cc: 
Bcc: 
Subject: This is a test email to remove the <command> command.

this is a test

When evaluating the rule, it doesn’t matter how the key word is written.
To remove it, you need an Action Mail Job, which only runs when the corresponding command rule applies.

'Replace <command>'

iQ.Suite Action Mail Job

**Basics**

- **Job name**: Replace <command>
- **Status**: Active
- **Priority**: 10000
- **Runs on**: All
- **Attachment dependency**: All
- **Controls for rules**
- **Positive selection rule dependency**: All true
- **Negated selection rule dependency**: All false

**Rule summary**
(Command Rule)

**Rule execution mode**: Set in global parameter

**Upon error in rule evaluation**: Do not run job

**Runs in a replicated environment**: on all servers

**Operations**

**Execution mode**

```
_command := "<command>";
_Field := "Subject";
_Value := Subject;

_lcase := @LowerCase(_value);

_left := @Left(_lcase_command);
_length_left := @Length(_left);

_right := @Right(_lcase_command);
_length_right := @Length(_right);

_LeftNow := @Left(_value; length_left);
_RightNow := @Right(_value; length_right);

_new := @Trim(_LeftNow + _RightNow);
_out := @If(_new = " _Value = '');

@SetField(_Field_out);
```

**Notes formula**

```
@Success
```

**Check correctness of formula syntax**
The job has the following structure

\[ _\text{command} := ";"; \]
\[ _\text{Field} := "Subject"; \]
\[ _\text{Value} := \text{Subject}; \]

\[ _\text{low} := @\text{LowerCase}(\text{Value}); \]

The green texts are the key word you used and the field in which you are querying it.

\[ _\text{left} := @\text{Left}(\text{low}; _\text{command}); \]
\[ _\text{length_left} := @\text{Length}(\text{left}); \]
\[ _\text{right} := @\text{Right}(\text{low}; _\text{command}); \]
\[ _\text{length_right} := @\text{Length}(\text{right}); \]

The field content is “standardized” here

\[ _\text{LeftNew} := @\text{Left}(\text{Value}; _\text{length_left }); \]
\[ _\text{RightNew} := @\text{Right}(\text{Value}; _\text{length_right }); \]

The position of the key word within the field is determined here

\[ _\text{new} := @\text{Trim}(\text{LeftNew}+ _\text{RightNew}); \]
\[ _\text{out} := @\text{If}(\text{new} = \"\" ; _\text{Value}; _\text{new}); \]

The text on the left and right of the key word is deleted

The new field value is formed from the left and right parts.

\[ @\text{SetField}(\text{Field}; _\text{out}); \]

This value is returned to the field.

@Success

Required for the Notes formula to work.

And now the e-mail will have the right appearance.

---

This is a test email to remove the command.

Group Admin to: Group Admin

this is a test