

String replace by regular expression in Delphi

TRegEx is a useful class for regular expression. You can define a instance with a regular expression, and match the case by TMatch. If the case exists, the search can be continued by TMatch.NextMatch. Following is a example to search XML tag by TRegEx and TMatch

```
uses System.RegularExpressions

.
.
.

function getXMLTagValue(source, tag:string):string;
var
  mRegEx: TRegEx;
  match: TMatch;
begin
  mRegEx := TRegEx.Create( '<' + tag + '>(.*?)</' + tag + '>', [roIgnoreCase,roMultiline]);
  match := mRegEx.Match( source);
  if match.Success then
  begin
    Result := match.Groups.Item[1].Value;
  end else begin
    Result := '';
  end;
end;
```

Below example continuously search the case as long as the result exists

```
procedure FindSomething(const searchMe : string);
var
  regexpr : TRegEx;
  match   : TMatch;
  group   : TGroup;
  i       : integer;
begin
  // create our regex instance, and we want to do a case insensitive search, in multiline mode

  regexpr := TRegEx.Create('^.*\b(\w+)\b\sworld',[roIgnoreCase,roMultiline]);
  match := regexpr.Match(searchMe);
  if not match.Success then
  begin
    WriteLn('No Match Found');
    exit;
  end;

  while match.Success do
  begin
    WriteLn('Match : [' + match.Value + ']');
    //group 0 is the entire match, so count will always be at least 1 for a match
    if match.Groups.Count > 1 then
    begin
      for i := 1 to match.Groups.Count -1 do
        WriteLn('    Group[' + IntToStr(i) + '] : [' + match.Groups.Item[i].Value + ']');
      end;
      match := match.NextMatch;
    end;
  end;
```