
Subject: Re: Very Simple Report Generator (Use QTF format)

Posted by [sergeynikitin](#) on Tue, 21 Aug 2012 19:23:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

jibe wrote on Sat, 18 August 2012 03:04 Yes, I saw that. But if the example shows how to use `##BT` and `##ET` and let me guess how it works, there is nothing about `##BP` and `##EP`. I think that a lot of people would appreciate a short explanation about that !

`##BT` and `##ET` - body repeatable block (begin and end) in table structure (`{ { .. :: .. :: .. .... .. } }`) - automatically find boundary of columns.

`##BP` and `##EP` - body repeatable block (begin and end) in plain text (may be in contract or article, for ex.)

Additionally, We can create custom body repeatable block for nested table data. (for example `##LANGBT` and `##LANGET` for languages, person can speaking)

This operation is very simple:

```
RepGenSubProc repLangs(rep, "##LANGBT","##LANGET");
repLangs.RepGenSubProcBody = THISBACK(PrintResume_CalcLangBody);
```

and definition of `PrintResume_CalcLangBody`

```
void AnketaListDlg::PrintResume_CalcLangBody() {
#undef VAR
#define VAR(x) rep.SubstVar("##" #x,AsString(sql[ANL_##x]));
    Sql sql;
    sql * SqlSelect(SqlAll()).From(ANKETALANGUAGE).Where(ANL_ANKETAID ==
gANK_ID).OrderBy(ANL_LANGUAGE);
    while (sql.Fetch()) {
        rep.RepGenCalculateBodyBeforeSubst();
        VAR(LANGUAGE);
        VAR(FINDLEVEL);
        VAR(LEVEL);
        rep.RepGenCalculateBodyAfterSubst();
    }
}
```

`##BT` and `##ET` can also create variable-columns reports.
