
Subject: Re: template + convert problem

Posted by [sergeynikitin](#) on Thu, 30 Apr 2015 05:55:24 GMT

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

I've made this template!!!

Universal Lookup Convertor syntax like this:

```
(Single<ConvLookup<COMPANY,COM_ID,COM_NAME> >() )
```

Let's test:

dict.h

```
template<typename T> T& LookupSetup ( T& grid, Sqlld& tablename, Sqlld& colid, Sqlld&
colname, bool showid = false, bool valueiscode = true, SqlBool where = true );
template<> DropGrid& LookupSetup ( DropGrid& grid, Sqlld& tablename, Sqlld& colid, Sqlld&
colname, bool showid, bool valueiscode, SqlBool where );
template<> WithDropChoice<EditString>& LookupSetup ( WithDropChoice<EditString>& grid,
Sqlld& tablename, Sqlld& colid, Sqlld& colname, bool showid, bool valueiscode, SqlBool where );
template<> DropList& LookupSetup ( DropList& grid, Sqlld& tablename, Sqlld& colid, Sqlld&
colname, bool showid, bool valueiscode, SqlBool where );
```

```
template<typename T> T& LookupSetup1 ( T& grid, Sqlld& tablename, Sqlld& colid, Sqlld&
colname1, Sqlld& colname2, bool showid = false, bool valueiscode = true, SqlBool where = true );
template<> DropGrid& LookupSetup1 ( DropGrid& grid, Sqlld& tablename, Sqlld& colid, Sqlld&
colname1, Sqlld& colname2, bool showid, bool valueiscode, SqlBool where );
template<> WithDropChoice<EditString>& LookupSetup1 ( WithDropChoice<EditString>& grid,
Sqlld& tablename, Sqlld& colid, Sqlld& colname1, Sqlld& colname2, bool showid, bool
valueiscode, SqlBool where );
```

```
template< Sqlld &LookupTable, Sqlld &LookupCode, Sqlld &LookupValue, bool show_id=false >
struct ConvLookup : Convert
{
    Value Format ( const Value &q ) const
    {
        int sid = q;
        if ( q.IsNull() ){
            return Null;
        } else if ( q == 0 ) {
            return "";
        }
    }

    static VectorMap<int, Value> valuecache;
    static Time lastcleartime;
    if ( GetSysTime() - lastcleartime > 600 )
    {
```

```

    valuecache.Clear();
    lastcleartime = GetSysTime();
}
if ( valuecache.Get ( sid, "X" ) != "X" )
{
    if (show_id) {
        return AsString ( sid ) + ": " + AsString(valuecache.Get ( sid, Value() ));
    } else {
        return valuecache.Get ( sid, Value() );
    }
}
else
{
    Sql sql;
    sql * Select ( LookupValue ).From ( LookupTable ).Where ( LookupCode == sid );
    if ( sql.Fetch() )
    {
        valuecache.Add ( sid, AsString(sql[LookupValue]) );
        if ( valuecache.Get ( sid, "X" ) != "X" )
        {
            if (show_id) {
                return AsString ( sid ) + ": " + AsString(valuecache.Get ( sid, Value() ) );
            } else {
                return valuecache.Get ( sid, Value() );
            }
        }
        return sql[LookupValue] ;
    }
    else
    {
        return AsString ( sid ) + ": ??? " + t_(~LookupTable) + " ???" ;
    }
}
return "ERROR RETURN PLACE" ;
}
};

```

dict.cpp

```

template<typename T> T& LookupSetup ( T& grid, SqlId& tablename, SqlId& colid, SqlId&
colname, bool showid, bool valueiscode, SqlBool where ) {}

```

```

template<> DropGrid& LookupSetup ( DropGrid& grid, SqlId& tablename, SqlId& colid, SqlId&
colname, bool showid, bool valueiscode, SqlBool where )
{

```

```

Sql sql;

if ( grid.GetList().GetColumnCount() == 0 )
{
    grid.ClearButton();
    grid.Resizeable ( false );
    grid.AddColumn ( colid, "", showid ? 10 : 0, true );
    grid.AddColumn ( colname, showid ? " : " : "", 70 );

    if ( showid )
        grid.SetValueColumn ( 0 );
    else
        grid.SetValueColumn ( 1 );

    if ( valueiscode )
    {
        grid.SetKeyColumn ( 0 );
    }
    else
    {
        grid.SetKeyColumn ( 1 );
    }

    if ( showid )
    {
        grid.AddValueColumn ( 0 );
    }
    grid.AddValueColumn ( 1 );

    grid.SetFindColumn ( 1 );

    grid.NoHeader();

    grid.ColorRows();
}

grid.Clear();

sql * Select ( colid,
              colname ).From ( tablename ).Where ( where ).OrderBy ( colid );

while ( sql.Fetch() )
{
    grid.Add ( sql[colid], sql[colname] );
}

return grid;
}

```

```

template<> WithDropChoice<EditString>& LookupSetup ( WithDropChoice<EditString>& grid,
Sqlld& tablename, Sqlld& colid, Sqlld& colname, bool showid, bool valueiscode, SqlBool where )
{
    Sql sql;
    grid.ClearList();
    sql * Select ( colname ).From ( tablename ).Where ( where ).OrderBy ( colname );

    while ( sql.Fetch() )
    {
        grid.AddList ( sql[colname] );
    }

    return grid;
}

```

```

template<> DropList& LookupSetup ( DropList& grid, Sqlld& tablename, Sqlld& colid, Sqlld&
colname, bool showid, bool valueiscode, SqlBool where )
{
    Sql sql;

    if ( grid.GetList().GetColumnCount() == 0 )
    {
        grid.SetDropLines ( 12 );
    }

    grid.ClearList();

    sql * Select ( colname ).From ( tablename ).Where ( where ).OrderBy ( colname );

    while ( sql.Fetch() )
    {
        grid.Add ( sql[colname] );
    }

    return grid;
}

```

```

template<typename T> T& LookupSetup1 ( T& grid, Sqlld& tablename, Sqlld& colid, Sqlld&
colname1, Sqlld& colname2, bool showid, bool valueiscode, SqlBool where ) {}

```

```

template<> DropGrid& LookupSetup1 ( DropGrid& grid, Sqlld& tablename, Sqlld& colid, Sqlld&
colname1, Sqlld& colname2, bool showid, bool valueiscode, SqlBool where )
{
    Sql sql;

    if ( grid.GetList().GetColumnCount() == 0 )

```

```

{
    grid.ClearButton();
    grid.Resizeable ( false );
    grid.AddColumn ( colid, "", showid ? 10 : 0, true );
    grid.AddColumn ( colname1, valueiscode ? " : " : "", 70 );
    grid.AddColumn ( colname2, "", 70 );
    if ( showid )
        grid.SetValueColumn ( 0 );
    else
        grid.SetValueColumn ( 1 );

    if ( valueiscode )
    {
        grid.SetKeyColumn ( 0 );
    }
    else
    {
        grid.SetKeyColumn ( 1 );
    }

    if ( showid )
    {
        grid.AddValueColumn ( 0 );
    }
    grid.AddValueColumn ( 1 );

    grid.SetFindColumn ( 1 );

    grid.NoHeader();

    grid.ColorRows();
}

grid.Clear();

sql * Select ( colid,
              colname1, colname2 ).From ( tablename ).Where ( where ).OrderBy ( colid );

while ( sql.Fetch() )
{
    grid.Add ( sql[colid], sql[colname1], sql[colname2] );
}

return grid;
}

```

```

template<> WithDropChoice<EditString>& LookupSetup1 ( WithDropChoice<EditString>& grid,
SqlId& tablename, SqlId& colid, SqlId& colname1, SqlId& colname2, bool showid, bool

```

```
valueiscode, SqlBool where )
{
    Sql sql;
    grid.ClearList();
    sql * Select ( colname1, colname2 ).From ( tablename ).Where ( where ).OrderBy ( colname1
);

    while ( sql.Fetch() )
    {
        grid.AddList ( AsString ( sql[colname1] ) + " " + AsString ( sql[colname2] ) );
    }

    return grid;
}
```

And in AddColumn if GridCtrl you may write as this:

```
listvisitor.AddColumn ( VISR_COMPANY_ID, t_("Company"), 265
).SetConvert(Single<ConvLookup<COMPANY,COM_ID,COM_NAME> >()).HeaderAlignCenter();
```
