

U++ - Feature #100

Add introspection tools for .sch, improve SqlCtrls

09/22/2011 08:51 AM - Miroslav Fidler

Status:	Approved	Start date:	09/22/2011
Priority:	Immediate	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description			
<p>- it should be possible to get a set of columns of given table name from .sch - it should be possible to automatically assign SqlCtrls based on layout ids - add EditKey and DeleteKey and perhaps some sort of InsertKey to SqlCtrls</p>			

History

#1 - 11/16/2011 08:45 AM - Miroslav Fidler

```
// Introspection
```

```
#define TABLE(x) struct SINS_##x##_ { SINS_##x##_(); } SINS_##x##_; SINS_##x##_::SINS_##x##_() { SchDbInfo(#x)
#define COLUMN(type, ctype, name, width, prec) .Column(#name)
#define REFERENCES(table) .References(#table)
#define REFERENCES_CASCADE(table) .References(#table)
#define REFERENCES_(table, column) .References(#table, #column)
#define REFERENCES_CASCADE_(table, column) .References(#table, #column)
#define COLUMN_ARRAY(type, ctype, name, width, prec, items) .ColumnArray(#name, items);
#define END_TABLE ; }
```

```
#include SCHEMADIALECT
```

```
struct SchTableInfo {
    Vector<String> column;
    Vector<String> ref_table;
    Vector<String> ref_column;

    SchTableInfo& Column(const char *name);
    SchTableInfo& References(const char *table);
    SchTableInfo& References(const char *table, const char *column);
};
```

```
static ArrayMap<String, SchTableInfo> sSchTableInfo;
static SchTableInfo sSchTableInfoZero;
```

```
SchTableInfo& SchTableInfo::Column(const char *name)
{
    column.Add(name);
    ref_table.Add();
    ref_column.Add();
    return *this;
}
```

```
}
```

```
SchTableInfo& SchTableInfo::References(const char *table)
```

```
{  
    ref_table.Top() = table;  
    return *this;  
}
```

```
SchTableInfo& SchTableInfo::References(const char *table, const char *column)
```

```
{  
    References(table);  
    ref_column.Top() = column;  
    return *this;  
}
```

```
SchTableInfo& SchDbInfo(const char *table)
```

```
{  
    return sSchTableInfo.GetAdd(table);  
}
```

```
const SchTableInfo& GetSchTableInfo(SqlId table)
```

```
{  
    return sSchTableInfo.Get(~table, sSchTableInfoZero);  
}
```

```
SqlBool Join(String tab1, String tab2)
```

```
{  
    const SchTableInfo& t1 = GetSchTableInfo(tab1);  
    const SchTableInfo& t2 = GetSchTableInfo(tab2);  
    for(int i = 0; i < t1.ref_table.GetCount(); i++)  
        if(t1.ref_table[i] == tab2)  
            return SqlId(t1.column[i]).Of(SqlId(tab1)) == SqlId(t2.column[0]).Of(SqlId(tab2));  
    for(int i = 0; i < t2.ref_table.GetCount(); i++)  
        if(t2.ref_table[i] == tab1)  
            return SqlId(t2.column[i]).Of(SqlId(tab2)) == SqlId(t1.column[0]).Of(SqlId(tab1));  
    return SqlBool::False();  
}
```

```
SqlBool FindJoin(const char *tables)
```

```
{  
    Vector<String> s = Split(tables, ',');  
    if(s.GetCount() >= 2) {  
        String tab1 = s.Top();  
        for(int i = 0; i < s.GetCount() - 1; i++) {  
            SqlBool b = Join(tab1, s[i]);  
            if(!b.IsFalse())  
                return b;  
        }  
    }  
    NEVER();  
    return SqlBool::False();  
}
```

#2 - 12/16/2011 08:59 AM - Miroslav Fidler

- *Priority changed from Normal to Immediate*

#3 - 12/16/2011 08:58 PM - Miroslav Fidler

- *Status changed from New to Approved*