

ArchiCAD,



SQL И ODBC

В предлагаемой вашему вниманию статье представлены способы взаимодействия между внешними программами и моделью-проектом ArchiCAD посредством драйверов ODBC¹. Статья ориентирована прежде всего на технических специалистов, имеющих хотя бы минимальные познания в области ОС Windows, программирования и баз данных: просто повторяя действия автора, вы сможете получить все результаты самостоятельно. Надеемся, эти материалы будут интересны и тем, кто хотел бы больше узнать о технологии ODBC-SQL-ArchiCAD.

Введение

Программа ArchiCAD предназначена для архитектурно-строительного проектирования. В основу принципа ее работы положена концепция Виртуального здания — модели, состоящей из трехмерных архитектурно-строительных элементов. Благодаря тому что пользователь ArchiCAD работает с образами реальных объектов (стен, окон, дверей, балок, элементов мебели и строительных конструкций), он может максимально подробно составить модель проектируемого или уже существующего здания. При этом Виртуальное здание представляет собой обычную базу данных, в которую можно делать SQL-запросы² через ODBC-драйвер и получать любую необходимую информацию по проекту, структурированную в виде

таблиц. Эта технология позволяет организовать динамическую (!) связь модели здания с любыми внешними программами. Области применения этой технологии практически ничем не ограничены: данные о модели можно передавать в программы по прочностным расчетам, сметные программы, программы для расчета теплотеперь, анализа воздушных потоков и т.д. Ниже мы рассмотрим механизм применения ArchiCAD при эксплуатации зданий и формировании различной отчетной документации в среде Microsoft Excel (списки инвентарных номеров офисной мебели, расположения персонала по зданию, эксплуатируемых помещений, анализ по срокам эксплуатации оборудования и оргтехники, расчет амортизационных отчислений и т.д.).

SQL в ArchiCAD

Возможность исполнять SQL-запросы в ArchiCAD реализована в программе начиная с версии 7.0. Самый простой способ сделать запрос в модель ArchiCAD — ввести его непосредственно из среды программы в диалог *Запрос SQL*. Рассмотрим этот путь подробнее на примере ArchiCAD версии 9.

Прежде всего нам необходимо активировать команду *SQL* меню *Расчеты* (рис. 1): по умолчанию данный пункт отключен. Для этого перейдите в окно *Параметры* → *Окружающая среда* → *Схемы расположения команд* → *Меню* и установите находящийся слева список меню согласно упорядочению *Все меню в алфавитном порядке*. Выберите меню *Расчеты* во всплывающем меню *По-*

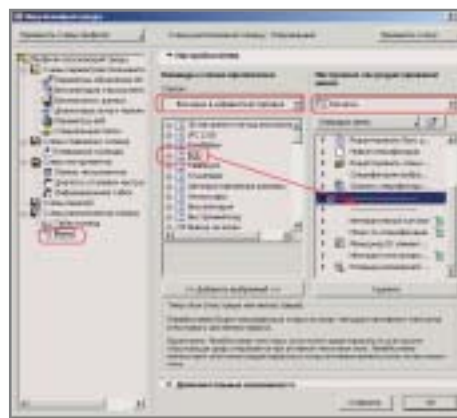


Рис. 1. Как включить отображение пункта SQL в меню *Расчеты*?

¹ODBC (Open DataBase Connectivity) — открытый интерфейс доступа к базам данных, разработанный в 1992 году группой SQL Access.

²SQL (Structured Query Language) — язык структурированных запросов. Стандартный язык запросов в базу данных любого типа.

Стандарт принят в 1986 году Американским национальным институтом по стандартам (ANSI) и Международной организацией по стандартам (ISO).

строение или редактирование меню. Найдите в левом списке команду SQL и перетащите ее в правый список. Нажмите OK.

По завершении этих действий в меню *Расчеты* появится новый пункт SQL. Выберите команду *Запрос...* и вставьте следующую строчку (в поле *Сформулируйте здесь ваш запрос SQL*): `SELECT * FROM WALLS`.

Если дословно перевести запрос на русский язык, он означает следующее: ВЫБРАТЬ ВСЁ ИЗ СТЕН. С помощью этой простейшей команды мы получим список всех стен, используемых в проекте ArchiCAD, и всех их параметров (высота, толщина, этаж, площадь внутренних и внешних поверхностей и т.д.). Условная запросы (например, "Выбрать все стены с первого этажа" или "Выбрать все несущие стены с первого этажа"), мы получаем любую информацию из проекта. Более подробно о структуре SQL-запросов в ArchiCAD можно прочитать в справке по программе.

Проблема в том, что результат SQL-запроса из ArchiCAD выводится в браузер в формате XML-документа³. Этот результат интересен с теоретической точки зрения, но его сложно использовать практически. Информацию (например, о конструктивных элементах) гораздо удобнее получать во внешних программах. Как же это сделать? Прежде всего необходимо открыть доступ к проекту из внешних программ. При этом ODBC-драйверы выступают в качестве средств доступа — своего рода транспорта, с помощью которого мы будем отправлять наши запросы в проект и получать ответы.

Установка и настройка ODBC-драйвера

ArchiCAD Plan ODBC Driver свободно распространяется разработчиком ArchiCAD — компанией Graphisoft. Новейшую версию вы всегда сможете найти на сайте www.graphisoft.com в разделе *Support/Developer* (прямая ссылка: <http://www.graphisoft.com/support/developer/>). На момент написания статьи здесь были размещены драй-

веры под ArchiCAD версий 8.1 и 9. Кроме того, на сайте выложена компактная CHM-документация, описывающая основные принципы работы ODBC-драйверов для ArchiCAD. При написании статьи использовались драйверы версии 9.00.00.6201 для ArchiCAD 9 от 21 января 2005 года.

Сама установка драйверов не должна вызвать особых проблем — используется обычный Мастер установки Install Shield. По умолчанию драйверы устанавливаются в папку *C:\Program Files\Graphisoft\ArchiCAD Plan ODBC Driver v9.0.0*. Когда установка завершена, в систему добавляются новые драйверы ODBC — их можно увидеть в диалоге *ODBC Data Source Administration* (Пуск → Настройка → Панель управления → Администрирование → Источники данных (ODBC)). Отображение этой информации на мониторе компьютера показано на рис. 2.

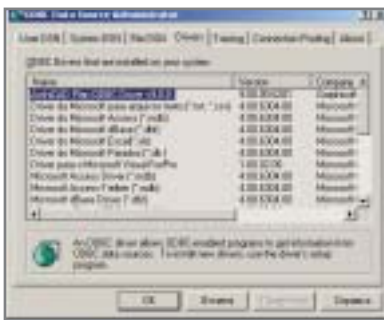


Рис. 2. Список ODBC-драйверов на компьютере

Для работы драйвера следует обязательно установить программу QuickTime — ее можно загрузить с сайта компании Apple (www.apple.com). При написании статьи использовался QuickTime версии 6.0.

Далее для настройки доступа к проекту ArchiCAD необходимо задать источник ODBC — исходный файл, из которого мы будем брать информацию о проекте. Таким источником может служить либо PLN-файл (стандартный файл ArchiCAD), либо PLA-файл (архивный файл ArchiCAD, который содержит все используемые в проекте библиотечные элементы: объекты мебели, окна, двери, модели оргтехники и т.д.). Для

настройки источника заходим на закладку *System DSN* диалога *ODBC Data Source Administration* и нажимаем кнопку *Add...* В появившемся диалоге задаем драйвер, через который будем в дальнейшем работать (конечно же, *ArchiCAD Plan ODBC Driver v9.0.0*), и нажимаем кнопку *Готово*.

После этих действий появится Мастер настройки, который попросит вас дать имя ODBC-проекта и его описание (например, *TEST_PROJECT* — Тестовый проект для статьи ArchiCAD и ODBC), а также обозначить путь до файла проекта ArchiCAD и библиотек по проекту. Вы можете указать путь к своему проекту ArchiCAD. Если же готового проекта нет, то либо скачайте необходимые материалы с сайта www.archicad.ru, либо обратитесь по электронной почте к автору этих строк.

Если всё прошло успешно, в списке *System DSN* появится новый проект *TEST_PROJECT* (рис. 3).



Рис. 3. Тестовый проект, с которым мы будем работать

Имя *TEST_PROJECT* очень важно для нашей работы — именно через него мы будем получать доступ к проекту.

Доступ к проекту

Итак, у нас есть проект ArchiCAD и настроен доступ к нему — фактически мы открыли ворота в проект. Теперь необходимо указать внешним приложениям, что такие ворота существуют.

В роли внешнего приложения может выступать любая программа: Visual Basic script, приложение к Microsoft Office (Word, Excel),

³XML (Extensible Markup Language) — расширяемый язык гипертекстовой разметки, используемый для создания и размещения документов в среде Internet. Язык XML использует структуру тегов и определяет содержание гипертекстового документа. XML позволяет автоматизировать обмен данными, не прибегая к существенному объему программирования.

Microsoft Access или самостоятельная программа, написанная вашими программистами на языках Delphi, C, C++, C# и т.д. Главное, что эта программа должна уметь устанавливать ODBC-соединение с проектом ArchiCAD. Давайте рассмотрим, как это делается с помощью Microsoft Excel.

Открываем Microsoft Excel, создаем книгу с именем *testODBC.xls* и заходим в редактор Visual Basic (*Сервис* → *Макрос* → *Редактор Visual Basic*). Пишем функцию, которая открывает связь с проектом TEST_PROJECT — для этого в списке проектов редактора находим проект VBA Project (*testODBC.xls*), щелкаем на нем правой кнопкой мыши и выбираем команду *Insert* → *Module*. А вот текст функции:

```
Function Main()
    Dim ConnDB

    Set ConnDB =
    CreateObject("ADODB.Connection")
    If IsObject(ConnDB) Then
        ConnDB.Open "TEST_
        PROJECT"
        MsgBox ("Установлено
        соединение с проектом " &
        StringDSN)

        ' здесь будем делать SQL-
        запросы в проект
        TEST_PROJECT

        ConnDB.Close
    End If
    Set ConnDB = Nothing
End Function
```

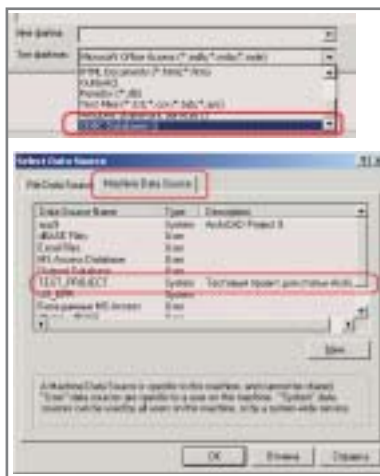


Рис. 4. Связываем базу данных Access с ArchiCAD-проектом

Функция достаточно проста: сначала мы создаем с помощью команды *CreateObject()* объект связи, а затем функцией *ConnDB.Open* устанавливаем связь с проектом TEST_PROJECT. Если связь установлена, выводим на экран сообщение об этом.

После этого сообщения можно делать SQL-запросы в проект. Но какова структура базы данных ArchiCAD-проекта? К каким таблицам можно создавать запрос? На сегодня полного описания структуры базы данных нет, но в выяснении структуры БД Виртуального здания нам может помочь Microsoft Access.

Структура базы данных ArchiCAD-проекта

Открываем Microsoft Access и создаем новую базу данных с именем *testODBC.mdb*. Выбираем команду *Файл* → *Внешние данные* → *Связь с таблицами...* после чего в стандартном диалоге открытия файлов задаем тип файла ODBC Databases() (рис. 4).

Далее на закладке *Machine Data Source* указываем наш тестовый проект TEST_PROJECT и нажимаем кнопку *OK*. В результате мы получаем список всех таблиц ArchiCAD-проекта, к которым возможен доступ. Их здесь 68: таблица стен, этажей, слоев, параметров объектов и т.д. Если вы хотите задать простейший запрос по всем стенам, который мы делали ранее, понадобится указать только одну таблицу — *WALLS*.

В нашем примере мы построим более сложный запрос: получим список инвентарных номеров всех объектов со слоя "Офисная мебель". Для такого запроса необходимо получить доступ к трем таблицам: слоев (*LAYERS*), объектов (*OBJECTS*) и к таблице параметров объектов (*PARAMETERS_OF_OBJECTS*).

С помощью клавиши **CTRL** набираем нужные нам таблицы и нажимаем *OK*. Access трижды покажет диалог выбора индекса в каждой таблице — ничего



Рис. 5. Таблицы в Microsoft Access

более не указывая просто нажимаем кнопку *OK*. В итоге на закладке *Таблицы* получаем ссылки на три таблицы ArchiCAD-проекта (рис. 5).

Далее с помощью Microsoft Access сформируем SQL-запрос.

Наш первый SQL-запрос в проект ArchiCAD

Переходим в Microsoft Access на закладку *Запросы* и выбираем команду *Создание запросов в режиме конструктора*. Появится диалог *Добавление таблицы*.

На этом этапе построим упрощенный запрос: получим список всех объектов с параметром "Инвентарный номер". Для построения такого запроса нам нужны две таблицы: *OBJECTS* и *PARAMETERS_OF_OBJECTS*. Находим таблицы на одноименной закладке и, дважды щелкнув на них, добавляем в запрос. Закрываем диалог *Добавление таблицы*, нажав на кнопку *Заккрыть*.

Следующим шагом нам нужно связать две таблицы. Единый параметр этих таблиц — код объекта: в таблице *OBJECTS* его имя *ID*, а в таблице *PARAMETERS_OF_OBJECTS* — *OBJECT_ID*. Методом *drag&drop* переносим параметр *ID* на параметр *OBJECT_ID* из другой таблицы — Access покажет тонкой линией связь между таблицами (рис. 6).

Теперь укажем, какие столбцы выводятся с помощью запроса. Прежде всего нам понадобится имя объекта — чтобы знать, с каким объектом мы работаем. В таблице *OBJECTS* соответствующий столбец именуется как *LIBRARY_PART_NAME*. Находим его в списке таблицы *OBJECTS* и дважды щелкаем на нем: эта переменная появится в первом столбце нижней части окна запроса.



Рис. 6. Наш первый SQL-запрос в графическом формате

Поскольку мы хотим отобразить только значения переменной "Инвентарный номер", нам нужно знать значение переменной VALUE и отобразить из таблицы PARAMETERS_OF_OBJECTS лишь те строки, у которых переменная NAME равна "Инвентарный номер". Дважды щелкаем на переменных NAME и VALUE и добавляем их в нижнюю часть окна запроса, рядом со столбцом LIBRARY_PART_NAME. Теперь в поле *Условия отбора* столбца NAME добавляем строку *Инвентарный номер* — и запрос готов.

На данный запрос можно посмотреть в "классическом" виде (то есть в виде текста). Для этого выполним команду *Вид → Режим SQL* (вернуться обратно — *Вид → Конструктор*):

```
SELECT
OBJECTS.LIBRARY_PART_NAME,
PARAMETERS_OF_OBJECTS.NAME,
PARAMETERS_OF_OBJECTS.VALUE FROM OBJECTS INNER JOIN
PARAMETERS_OF_OBJECTS ON
OBJECTS.ID =
PARAMETERS_OF_OBJECTS.OBJECT_ID WHERE (((PARAMETERS_OF_OBJECTS.NAME)="Инвентарный номер"));
```

Выполним наш запрос: команда *Запрос → Запуск*. В результате Microsoft Access отберет из ArchiCAD-проекта все объекты, имеющие параметр "Инвентарный номер", и выведет их на экран в виде таблицы (рис. 7).

А вот и первый положительный эффект от этой технологии: просматривая результат SQL-запроса, мы видим, что два шкафа у нас не имеют инвентарного номера. Надо исправлять...

Более сложные запросы

Усложним запрос: получим таблицу, которая будет содержать данные только о тех объектах, которые лежат на слое "Офисная мебель". Тут нужно сделать небольшое отступление и объяснить некоторые ограничения текущей версии ArchiCAD Plan ODBC Driver v9.0.0.

Эта версия позволяет одновременно работать с любым количеством таблиц, но, если сделать один SQL-запрос к трем и более таблицам, время запроса очень сильно возрастает. Например, если число объектов невелико, теоретически возможен запрос, который получает и таблицу всех объектов с определенного слоя, и значение заданного параметра объекта. Правда, в реальности, сделав такой запрос по своему проекту, результатов я так и не дождался. Так что советую разбивать сложные запросы на несколько более простых, каждый из которых работает не более чем с двумя таблицами. В нашем случае необходимо одним запросом получить код слоя "Офисная мебель", а затем, с помощью другого запроса, — список объектов, у которых задан определенный код слоя и есть значение параметра "Инвентарный номер".

Создаем первый запрос: получить код слоя "Офисная мебель". Из диалога *Добавление таблицы* указываем таблицу LAYERS и нажимаем *Добавить*. В поле конструктора появится нужная нам таблица с перечислением всех ее столбцов, из которых нас интересуют только два: ID и NAME (соответственно код слоя и его имя). Дважды щелкаем на них, и они появляются в нижней части конструктора. Не забываем указать условия отбора. В итоге у нас должен получиться запрос, показанный на рис. 8.

В виде обычного SQL-запроса (*Вид → Режим SQL*) он выглядит так:

```
SELECT LAYERS.ID,
LAYERS.NAME FROM LAYERS
WHERE
(((LAYERS.NAME)="Офисная мебель"));
```

По команде *Запрос → Запуск* получаем результат SQL-запроса: код слоя "Офисная мебель" в моем проекте равен "41". Теперь, немного усложняя наш первый SQL-запрос (из таблицы OBJECTS вытаскиваем столбец LAYER_ID и задаем условия отбора, равные "41"), получаем окончательный вариант таблицы. Сделайте это самостоятельно.

Конечно, если вы хорошо знаете SQL-язык, вам нет необходимости использовать Access — достаточно знать названия таблиц и заголовки столбцов. Но Microsoft Access дает уникальную возможность построить запрос с помощью графического интерфейса, проверить его и реализовать в своей программе — уже зная, что получается в результате такого запроса.

Реализация в Microsoft Excel

Теперь все знания о нашем проекте и запросах в него необходимо оформить в виде кода Basic-программы. В принципе это чисто техническая задача — любой человек, достаточно хорошо знакомый с Visual Basic, решит ее в течение получаса. И тем не менее, какие здесь могут быть подводные камни?

Во-первых, SQL-запрос из Visual Basic чуть отличается от запроса из Microsoft Access: двойные кавычки в запросе заменяются одинарными. То есть запрос Access:

LIBRARY_PART_NAME	NAME	VALUE
Полка для книг	Инвентарный номер	
Полка для журналов	Инвентарный номер	
Полка для книг	Инвентарный номер	
Полка для журналов	Инвентарный номер	
Полка для книг	Инвентарный номер	
Полка для журналов	Инвентарный номер	
Полка для книг	Инвентарный номер	CS CT 1 2.3
Полка для журналов	Инвентарный номер	CS CT 1 2.3
Полка для книг	Инвентарный номер	CS CT 1 2.3
Полка для журналов	Инвентарный номер	CS KRCO 1 2.4
Полка для книг	Инвентарный номер	CS TUM 1 2.1
Полка для журналов	Инвентарный номер	CS TUM 1 2.2
Полка для книг	Инвентарный номер	CS CT 1 2.4
Полка для журналов	Инвентарный номер	CS TUM 1 2.4
Полка для книг	Инвентарный номер	CS TUM 1 2.2
Полка для журналов	Инвентарный номер	CS KRCO 1 2.2
Полка для книг	Инвентарный номер	CS KRCO 1 2.2
Полка для журналов	Инвентарный номер	CS KRCO 1 2.1
Полка для книг	Инвентарный номер	CS KRCO 1 2.1
Полка для журналов	Инвентарный номер	CS CT 1 2.2
Полка для книг	Инвентарный номер	CS CT 1 2.2

Рис. 7. Результат работы SQL-запроса



Рис. 8. Запрос на код слоя "Офисная мебель"

```
SELECT LAYERS.ID,
LAYERS.NAME FROM LAYERS
WHERE (((LAYERS.NAME)="Офис-
ная мебель"));
```

переходит в запрос Visual Basic следующим образом:

```
StringSQL = "LAYERS.ID, LAY-
ERS.NAME FROM LAYERS
WHERE
(((LAYERS.NAME)='Офисная ме-
бель'))"
```

Во-вторых, делая SQL-запрос на Visual Basic, мы будем получать объекты Recordset — таблицы-результат. Работа с такими таблицами несколько усложнена, и поэтому их лучше преобразовывать в обычные таблицы-массивы. Будем делать это с помощью написанной нами функции *SqlSelectArr()*:

```
Sub SqlSelectArr(SqlArr, ConnDB, vSql)
Dim LocalRs
Set LocalRs =
ConnDB.Execute(vSql)
If Not LocalRs.EOF Then
SqlArr = LocalRs.GetRows
Else
Set SqlArr = Nothing
End If
LocalRs.Close
Set LocalRs = Nothing
End Sub
```

Функция получает на вход три параметра:

- *SqlArr* — таблица-массив, в которую записывается результат,
- *ConnDB* — база данных, в которую делается запрос,
- *vSql* — строка SQL-запроса

и работает следующим образом: с помощью команды *Execute* выполняется запрос в базу данных, а результат помещается в переменную *LocalRs* (объект *Recordset*). Далее значения *LocalRs* построчно переносятся в массив *SqlArr*.

Теперь соберем все полученные знания и реализуем на Visual Basic программу, которая сначала связывается с ArchiCAD-проектом, получает код слоя "Офисная мебель", а затем запрашивает значения переменной "Инвентарный номер" у всех объектов с заданного слоя и выводит результат в ячейки таблицы Excel. С этим должен справиться даже школьник. Вот код:

```
Function Main()
Dim ConnDB
Dim LayerList, ResultTable
Dim StringSQL As String
Dim LayerID As Integer

Set ConnDB =
CreateObject("ADODB.Connection")
If IsObject(ConnDB) Then
ConnDB.Open "TEST_PROJECT"
' MsgBox ("Установлено соеди-
нение с проектом " &
StringDSN)

StringSQL = "SELECT LAY-
ERS.ID FROM LAYERS
WHERE
(((LAYERS.NAME)='Офисная
мебель'))"
SqlSelectArr LayerList,
ConnDB, StringSQL

If IsArray(LayerList) Then
' MsgBox "Получили код
слоя 'Офисная мебель'"
LayerID = LayerList(0, 0)

StringSQL = "SELECT
OBJECTS.LIBRARY_PART_
NAME,
PARAMETERS_OF_OBJEC
TS.VALUE FROM OBJECTS
INNER JOIN PARAME-
TERS_OF_OBJECTS ON
OBJECTS.ID = PARAME-
TERS_OF_OBJECTS.OBJE
CT_ID WHERE
(((OBJECTS.LAYER_ID)=" &
LayerID & ") AND ((PARAME-
TERS_OF_OBJECTS.NAME
)='Инвентарный номер'))"
SqlSelectArr ResultTable,
ConnDB, StringSQL

If IsArray(ResultTable) Then
' MsgBox "Получили
список объектов"
For i = 0 To
UBound(ResultTable,
2)
Cells(i + 1, 1).Value =
ResultTable(0, i)
Cells(i + 1, 2).Value =
ResultTable(1, i)
Next
End If
End If

ConnDB.Close
End If
Set ConnDB = Nothing
End Function
```

Заключение

Итак, мы рассмотрели, как на практике осуществляется связь ArchiCAD-модели с внешними программами. Напомню, что ограничений у этой технологии практически нет: современные средства позволяют связать между собой две любые программы. Только фантазией заказчика ограничена и область ее применения. Выбор конкретного механизма, определяемый наличием необходимых специализированных средств, сложностью задачи, временем на реализацию проекта и т.д., также остается за заказчиком.

Помимо универсальности подхода надо отметить и удобство работы: специалисты работают в понятной им среде. Так, службы эксплуатации оперируют не разрозненными таблицами и не выполненными от руки чертежами, а наглядной моделью объекта эксплуатации. При этом среда ArchiCAD является просто графической оболочкой, через которую удобно вводить данные в базу: инвентарный номер присваивается каждому объекту так же, как если бы мы написали его от руки.

Другой пример: совместная работа архитектора и конструктора. Наладив связь между ArchiCAD и системой прочностных расчетов, можно оперативно получать расчет по зданию при внесении изменений в проект. Для этого необходимо сначала максимально точно построить модель будущего здания в ArchiCAD, а далее с помощью SQL-запросов считывать информацию о несущих конструкциях: по ним будет автоматически формироваться и передаваться на расчет каркасная модель здания. Сроки анализа, а следовательно и сроки проектирования существенно сократятся...

У вас есть вопросы? Пишите мне на электронный адрес или звоните — постараюсь ответить.

Денис Ожигин
CSoft
Тел.: (095) 913-2222
E-mail: denis@csoft.ru

Автор благодарит Дмитрия Онухова за неоценимую помощь в написании статьи. Также хотелось бы выразить признательность за отличную идею Карлу Оттенштейну (Karl Ottenstein) с форума ArchiCAD-Talk (<http://archicad-talk.graphisoft.com>).