

Altium Designer 7.

Создание библиотеки на основе базы данных

Начиная с этой статьи, все действия будут приводиться для версии Altium Designer Summer 08. Здесь мы рассмотрим вопросы создания и подключения библиотек, которые используют пользовательские базы данных, базы данных производителей комплектующих изделий или организаций, занимающихся контрактной сборкой.

Владимир Пранович,
К. Т. Н.

pranovich@bsu.by

Как правило, в стандартных библиотеках большинство компонентов отсутствует, и рано или поздно встают вопросы как о создании перечня элементов, так и о доступности компонентов и их монтажа на печатной плате. В предыдущих публикациях автор привел пример базы данных, с помощью которой можно упростить эту проблему. В этой статье мы рассмотрим вопрос создания библиотеки на основе базы данных для того, чтобы непосредственно из нее выбирать компоненты и применять их в электрической схеме.

Затраченное на подготовку базы данных время с лихвой окупится благодаря простоте поиска компонентов для проекта (вид изображения компонентов, посадочных мест и других моделей отображается непосредственно на соответствующей панели), так и отсутствию необходимости ввода всех параметров этих элементов в SCH-редакторе. Более того, можно будет отказаться от инсталляции всего множества созданных разнообразных специализированных библиотек, ориентированных на определенный класс элементов.

Проблема создания непосредственно самой базы данных частично затрагивалась автором в статье [1], где была показана возможность ее использования. Более подробно этот вопрос описан в [2], где приведен пример организации базы, а также синхронизация параметров схемы с параметрами, хранящимися в базе. В данной статье мы продемонстрируем использование такой базы данных непосредственно в качестве библиотеки. За основу возьмем пример созданной ранее базы [2], модифицировав ее так, чтобы применить и в новом качестве.

Судя по откликам, слишком краткое изложение последовательности действий в ряде операций вызвало у читателей, пытавшихся их повторить, определенные трудности, поэтому в данной статье мы подробнее прокомментируем все наши действия, хотя при этом нужно будет пойти на некоторый повтор идентичных операций.

Создание и организация библиотеки на основе базы данных приведены в документе AP0133 Using Components Directly from Your Company Database. Мы советуем внимательно его изучить. В данном документе описаны, например, параметры, по которым Altium Designer находит элемент, указанный в базе данных, имя компонента для использования

в электрической схеме и имя библиотеки, где он находится. Аналогичным образом указываются ссылки на посадочное место и библиотеку для него, а также для моделей 3D-графики и других моделей, которые можно использовать в проекте. Следует отметить, что так же можно вводить информацию и для служебных параметров, например параметра Description. Эта информация просто заносится в соответствующие столбцы в базы данных. Однако вначале предложим создать базу данных из уже существующих библиотек, потому что такой путь в большей степени позволяет раскрыть структуру взаимосвязей между базой данных и библиотеками. Более того, этот способ может стать основой для создания новой базы.

Дополнительно советуем изучить документ AP0143 Database Library Migration Tools.

Создание Database Library из существующих библиотек

Рассмотрим процесс добавления к проекту библиотеки на основе базы данных (рис. 1):

1. В качестве примера автор использует собственные библиотеки, а для данной статьи специально создана группа проектов **Workspace=EDA_Expert.DsnWrk** (1a). Все библиотеки собраны в одном проекте 2006.PkgLib (1b). Помимо библиотек для компонентов и посадочных мест проект содержит ряд других файлов, назначение которых было описано в ТвЭП № 8 '2007, (1c). Однако сейчас для нас главное — наличие в проекте хотя бы одной библиотеки с компонентами. В данном случае это, например, библиотека 2006-D.SchLib (1d), которую мы и возьмем за основу, но читатель может использовать для этого любую библиотеку из собственного проекта.
2. Командой **Project >> Add New to project >> Other** вызываем панель **Files**.
3. На панели **Files** находим вкладку **New**, в ней строку **Other Documents** и в открывшемся окне выбираем тип нового документа, в данном случае **DataBase Library File**. Таким образом, мы добавим к проекту новый файл. По умолчанию ему будет присвоено имя **Database_Libs1.DbLib**. Командой **File >> Save AS** присвоим ему новое имя, например 2008.DbLib.

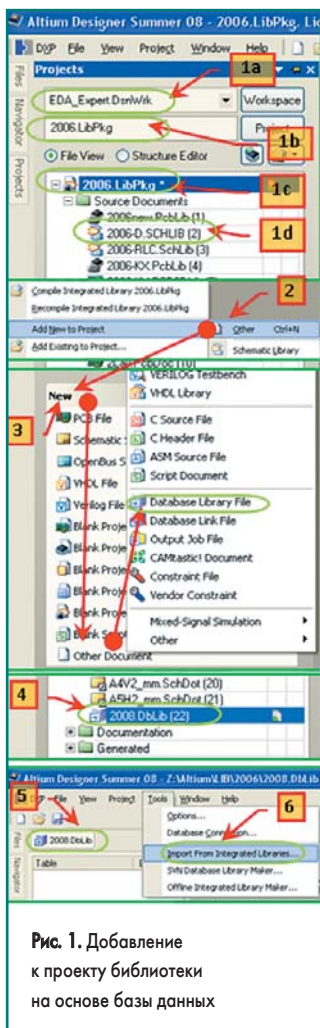


Рис. 1. Добавление к проекту библиотеки на основе базы данных

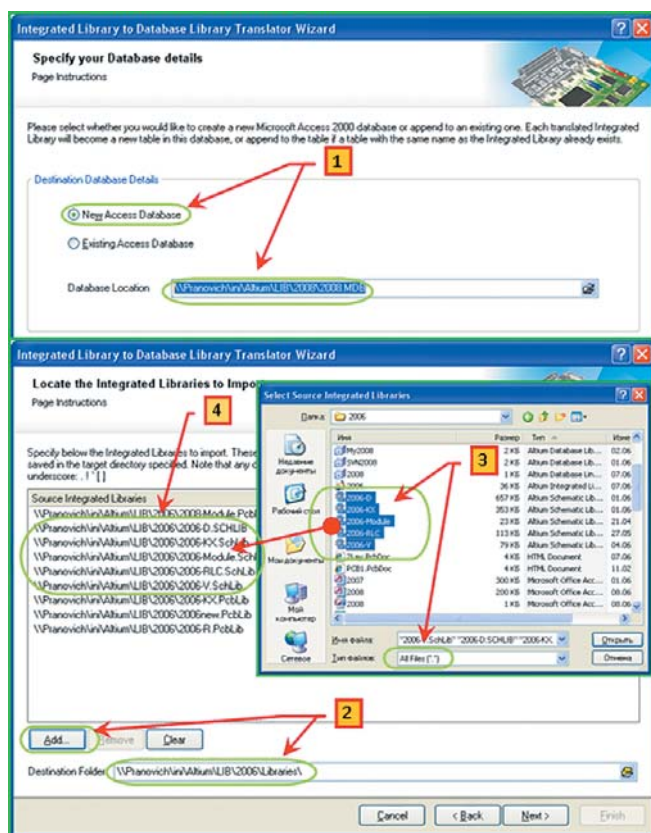


Рис. 2. Импорт данных из библиотек в базу данных

4. В результате в браузере проекта появится ссылка на созданный нами файл для библиотеки 2008.DbLib (библиотека на основе базы данных).
5. Сделаем активным (откроем) созданный нами документ 2008.DbLib.
6. Применим команду **Tool >> Import From Integrated Library**, вызвав транслятор библиотек.

Опишем работу транслятора библиотек подробнее (рис. 2):

1. В окне **Integrated Library to Database Library Translator Wizard** устанавливаем флаг **New Access Database**, для создания новой базы данных, а также укажем место и имя (например, `\\Pranovich\ini\Altium\LIB\2008\2008.MD`), где будет храниться созданная база. После проверки на доступность места нужно подтвердить данный путь (на рис. 2 это не показано).

2. В следующем окне можно добавить те библиотеки, импорт данных из которых будет произведен в базу данных. После нажатия кнопки **ADD** и указания пути, где находятся библиотеки, можно выбрать и конкретный состав библиотек для импорта. Если библиотеки лежат в разных папках, данную операцию следует повторить несколько раз.
3. Тип импортируемых библиотек по умолчанию — «интегральная», с расширением ***.IntLib**. Если вы используете только такие библиотеки, этого достаточно. Однако для примера мы импортируем и другие типы библиотек (библиотеки изображений компонентов для схемы ***.SchLib**, библиотеки посадочных мест ***.PcbLib**). В этом случае указываем тип расширения **All Files (*.*)**, а конкретный состав библиотек выбираем явно, выделив нужные в окне.
4. Список импортируемых библиотек.

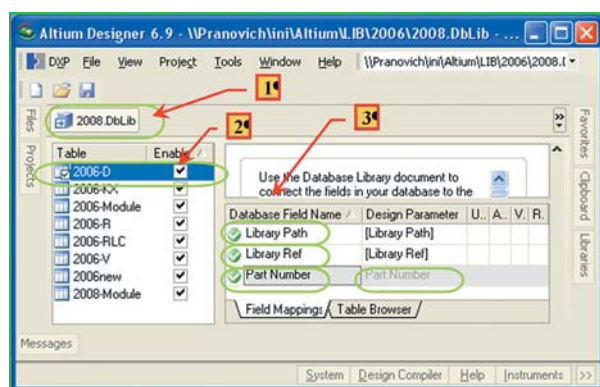


Рис. 3. Структура полей в базе данных для библиотеки

Новая библиотека (на основе новой базы данных) создана. Однако если при импорте мы не указали хотя бы одну интегральную библиотеку, элементы в созданной библиотеке будут отсутствовать, но сформируются поля с нужными параметрами для связи между базой и библиотеками.

Эти действия мы произвели только для того, чтобы посмотреть структуру библиотеки на основе базы данных и, соответственно, переделать уже имеющуюся базу данных [2] для поддержки формата связей с библиотекой DbLib или наполнить эту базу нужной дополнительной информацией.

Итак, рассмотрим структуру созданной библиотеки. Так как ранее мы создавали базу данных на основе электронных таблиц формата EXCELL, будем и далее ориентироваться на такой формат. В то же время автоматически импортированная база данных имеет формат ACCEES. Поэтому мы экспортируем базу данных в формат EXCELL, изучим структуру записей в базе данных и затем уже подключим к библиотеке базу данных в этом формате.

Итак, на рис. 3 указана структура созданной библиотеки 2008.DbLib и отмечено:

1. Имя созданной базы данных (2008.DbLib).
2. Список импортированных библиотек. Справа есть флаг доступности той или иной библиотеки. В этом окне дополнительно выделено имя первой в списке библиотеки, а именно 2006-D, справа внизу при этом будут отображены ее параметры. У автора это библиотека для компонентов микросхем. Признаком того, что библиотека доступна, является флаг **Enable**.
3. Столбец с параметрами из созданной базы данных. Справа от него столбец с идентичными параметрами непосредственно из библиотеки компонентов. Сейчас названия их совпадают, хотя это не обязательное требование. На данный момент для нас больший интерес представляют как раз названия самих параметров, и на этом остановимся подробнее.

Здесь показаны три обязательных параметра для подключения компонента из соответствующей библиотеки, а именно:

- **Library Path** — ссылка на файл библиотеки компонентов ***.SchLib**;
- **Library Ref** — имя компонента в указанной библиотеке;
- **Part Number** — имя компонента для новой библиотеки типа **Database Library**.

Заметьте, заголовки столбцов могут носить и другие имена, но информация из базы данных должна передаваться параметрам в библиотеке с именами такими названиями (**Design Parameter**, правая колонка, второй столбец). Нам нет необходимости придумывать специальные имена параметров, оставим их такими же. Но если у вас уже есть база данных со столбцами, где прописаны указанные имена и пути, то нет необходимости их переименовывать. Достаточно только установить взаимное соответствие между требуемыми столбцами в существующей базе данных и их аналогами в проекте. Последний параметр (**Part**

Таблица 1. Наименование и назначение параметров для подключения компонентов и моделей в базе данных

Параметр в базе	Параметр в проекте	Назначение	Примечание	Есть в примере
Library Path	[Library Path]	Имя и путь к библиотеке изображения компонента		Да
Library Ref	[Library Ref]	Имя компонента в библиотеке		Да
Description	[Description]	Значение параметра Description компонента для схемы		Да
Footprint Path n	[Footprint Path n]	Имя и путь к библиотеке посадочного места	Данных параметров может быть несколько. Для первого посадочного места или модели буква "n" в конце отсутствует. Для второго и последующих посадочных мест вместо буквы "n" записывается число 2 или номер в списке следующего посадочного места или модели. Использование нескольких параметров актуально только с точки зрения варьирования посадочных мест для различных условий монтажа компонентов на печатной плате	Да
Footprint Ref n	[Footprint Ref n]	Имя посадочного места в библиотеке		Да
PCB3D Path n	[PCB3D Path n]	Имя и путь к библиотеке трехмерной модели посадочного места		Нет
PCB3D Ref n	[PCB3D Ref n]	Имя трехмерной модели посадочного места в библиотеке		Нет
Sim Description	[Sim Description]	Ссылки на параметры и файлы моделей для симулятора. Ранее мы не касались вопросов работы с моделями для симулятора, и подключение данных моделей рассматривать не будем. Отметим, что через Database Library может быть подключена только одна модель для симуляции.		Нет
Sim Excluded Parts	[Sim Excluded Parts]			Нет
Sim File	[Sim File]			Нет
Sim Kind	[Sim Kind]			Нет
Sim Model Name	[Sim Model Name]			Нет
Sim Netlist	[Sim Netlist]			Нет
Sim Parameters	[Sim Parameters]			Нет
Sim Port Map	[Sim Port Map]			Нет
Sim Spice Prefix	[Sim Spice Prefix]			Нет
Sim SubKind	[Sim SubKind]			Нет

Number) имеет одно ограничение — он должен быть уникальным. Действительно, в базе данных не должно быть компонентов с одинаковыми названиями, так как именно по этому критерию производится поиск компонента в библиотеке и визуальное отображение его моделей в соответствующей панели Altium Designer.

В таблице 1 приведены все параметры, с помощью которых можно подключать к базе посадочные места, модели из других существующих библиотек. Там же даны комментарии по назначению и использованию параметров. Именно ориентируясь на таблицу 1, мы и модифицируем существующую базу и дополним ее ссылками на новые компоненты.

Теперь мы можем приступить к модификации базы данных, рассмотренной в [2].

Модификация базы данных для использования в качестве Database Library

Сначала определим минимальный набор параметров, необходимых для создания полнофункциональной базы данных с поддержкой требований **Database Library**. Ограничимся в данном примере только подключением компонентов для схемы и посадочного места для топологии. Более того, будем использовать ссылку только на одно посадочное место для каждого компонента. 3D-модели подключать вообще не будем, как и другие модели. Пока выполним только тот минимум, который необходим для разработки проекта и топологии печатной платы.

Не будем приводить весь процесс создания базы данных, а представим только структуру и назначение записей. Порядок расположения столбцов не имеет значения, вы можете его видоизменить под свои настройки. Мы отметим

только обязательные и рекомендованные для заполнения столбцы, а также напомним структуру пользовательских столбцов. Названия столбцов должны быть размещены в первой строке таблицы.

Таблица 2. Наименование и назначение столбцов в базе данных. Формат EXCELL

	A	B	C	D	E	F	G	H
1	Footprint Ref	Footprint Path	Library Ref	Library Path	LocalLink	GlobalLink	SkadLink	Type
2	CAPC1005X06[0402]N	2006new.PcbLib	C_Gost	2006-RLC.SchLib		CC10n/0402		C
3	CAPC10.3x10.3x10.3	2006new.PcbLib	CP_Gost	2006-RLC.SchLib		CE100u/50V/Gsmd		C
4	CAPCP3216X18[A-case]N	2006new.PcbLib	CP_Gost	2006-RLC.SchLib		CT10u/6V3/1206		C
5	SOIC127P600X175-14M	2006new.PcbLib	74HC02D_1G	2006-D.SCHLIB		74AC02		D
6	SOIC127P600X175-14M	2006new.PcbLib	74HC02D_4+1G	2006-D.SCHLIB		74AC02-1		D

	I	J	K	L	M	N
1	Modify	PDF	DescriptionPdf	ValueSCH	NoteSCH	Description
2	26.5.07			10n		CAP CER 10000PF 16V 10% X7R 0402
3	11.3.08	http://www.panasonic.../pic_fc_series.pdf		100u/50V		CAP 100UF 50V ELECT FC SMD
4				10u/6V3		CAP TANT 10UF 6.3V 10% SMD
5	3.5.08	http://www.onsemi.../MC74AC02-D.PDF		74AC02		IC GATE NOR QUAD 2INPUT 14-SOIC
6	3.5.08	http://www.onsemi.../MC74AC02-D.PDF		74AC02		IC GATE NOR QUAD 2INPUT 14-SOIC

	O	P	Q	R	S
1	Supplier	Supplier P/N	Vendor	Vendor P/N	ValueBOM
2	Digi-Key	490-1313-2-ND	Murata	GRM155R71C103KA01D	CAP CER 10000PF 16V 10% X7R 0402, GRM155R71C103KA01D, Murata
3	Digi-Key	PCE3260TR-ND	Panasonic - ECG	EEV-FC1H101P	CAP 100UF 50V ELECT FC SMD, EEV-FC1H101P, Panasonic - ECG
4	Digi-Key	718-1126-1-ND	Vishay/Sprague	293D106X96R3A2TE3	CAP TANT 10UF 6.3V 10% SMD, 293D106X96R3A2TE3, Vishay/Sprague
5	Digi-Key	MC74AC02DG-ND	ON Semiconductor	MC74AC02DG	IC GATE NOR QUAD 2INPUT 14-SOIC, MC74AC02DG, ON Semiconductor
6	Digi-Key	MC74AC02DG-ND	ON Semiconductor	MC74AC02DG	IC GATE NOR QUAD 2INPUT 14-SOIC, MC74AC02DG, ON Semiconductor

	T	U	V	W	X
1	NoteBOM	Provider	Package	Package Description	Marking
2					
3				0.406" L x 0.406" W x 0.402" H (10.30mm x 10.30mm x 10.20mm)	
4				3216-18 [EIA] 1206	
5				14-SOIC	
6				14-SOIC	

Итак, в таблице 2 буквами A, B, C, D, F обозначены те столбцы, по которым производится идентификация библиотечного элемента и связь с компонентом для электрической схемы и посадочным местом для печатной платы (в таблице сокращен или удален текст неинформационных параметров):

- **Footprint Ref** — имя посадочного места. Заполнить ячейку в данном столбце можно, например, путем поиска и копирования нужного названия посадочного места в одной из библиотек, либо просто скопировать такую же информацию из другой ячейки.
- **Footprint Path** — имя библиотеки, где находится посадочное место. Как правило, число используемых библиотек посадочных мест ограничено, и эта информация может быть просто введена или скопирована из соседней ячейки.
- **Library Ref** и **Library Path** — имя компонента (изображение компонента для схемы) и библиотека, где данный компонент находится (для отображения в схеме). Действия такие же, как в предыдущих пунктах.
- **LocalLink** — столбец зарезервирован для значений уникального номера внутри одного проекта. Примеры использования уникального номера будут приведены далее в отдельном разделе. При формировании общей базы данных это поле можно не заполнять или вообще не использовать данный параметр.

• **GlobalLink** — это аналог указанного выше параметра **Part Number**. В этом столбце не должно быть одинаковых записей, и данные в таблице 2 рекомендуем всегда сортировать соответственно его содержанию. Именно под этим компонент будет добавлен далее в схему. Поэтому принцип присвоения названия вы должны определить сами. Автор не будет навязывать используемую им систему присвоения имен и приводить ее не будет. Однако вы можете о ней судить по приведенным в таблице наименованиям.

Перейдем теперь к рекомендуемым автором столбцам в базе данных. Основная часть названий столбцов заимствована из описанной в статье [2] базы данных. Вы можете добавлять и видоизменять столбцы и их назначение (далее они приведены так, как использованы автором):

- **SkladLink** — столбец зарезервирован под ссылку на место хранения указанных компонентов на складе.
- **Type** — используется для определения на схеме начальных букв в параметре **Designator**, так как одно и то же изображение компонента можно использовать для разных типов элементов, например для аналоговых (DA) и цифровых (DD) микросхем. Более того, при разработке проекта не только для российского рынка, но и для зарубежного, можно использовать и другую буквенную кодировку.
- **Modify** — дата внесения последнего изменения в строку; используется только для контроля.
- **PDF** — ссылка на основной (Datasheet) PDF-документ, где приведено описание компонента. На этом параметре остановимся немного позже, так как при правильной записи ссылки вы можете ее вызвать непосредственно из схемы. Однако это потребует еще одного дополнительного параметра. Пока она присутствует в базе только для облегченного поиска документа другими средствами.
- **Description Pdf** — краткое описание элемента, взятое из PDF. Как правило, именно его можно использовать в качестве зарезервированного параметра **Description** в библиотечном компоненте схемного изображения элемента.
- **ValueSCH** — именно значение этого параметра автор использует в качестве отображения номинала или краткого названия компонента на схеме.
- **NoteSCH** — это дополнительный параметр, который может быть отображен на схеме, например точность для резисторов, ТКЕ для конденсаторов и т. п.
- **Description** — краткое стандартизированное описание компонента. Автору импонирует формат, предложенный на <http://www.digi-key.com/>: соответствующая ячейка просто копируется оттуда или заполняется по аналогии.
- **Supplier, Supplier P/N, Vendor, Vendor P/N** — наименования каталога, номера в каталоге для данного компонента, производителя

и обозначение компонента. Назначение данных столбцов понятно, а более подробно эти параметры описаны в [2].

- **ValueBOM** — слияние записей предыдущих пяти столбцов. Это наиболее близкая запись, которая может быть размещена в графе «Наименование» для перечня элементов проекта.
- **NoteBOM** — зарезервировано для дополнительных записей в графе «Примечание» перечня элементов.
- **Provider** — зарезервировано для указания организации, поставщика компонента.
- **Package** — обозначение посадочного места в известных стандартах, например в EIA, JEDEC.
- **Package Description** — описание посадочного места.
- **Marking** — обозначение на корпусе посадочного места (для сборочного чертежа).

Подключение базы данных к Database Library

Теперь, когда у нас есть модернизированная база данных, мы должны подключить ее к библиотеке, а затем последнюю установить. Процесс подключения базы данных аналогичен тому, как мы это делали ранее с обычной базой [2], однако мы приведем это описание полностью, включая особенности для Database Library.

1. Открываем файл Database Library 2008.DbLib, созданный в предыдущем разделе (рис. 4).
2. Изменяем тип Database с Microsoft Access на Microsoft Excel.
3. Нажимаем кнопку **Browse** и указываем ссылку на нашу базу данных, в примере это файл **GlobalDb.XLS**, который имеет один лист с именем **All**. На этом листе и находится вся информация нашей базы. Сразу отметим,

что в базе может быть несколько листов. Соответственно группироваться информация на листах может, например, по однородным типам компонентов. Однако мы рассмотрим пример только одного листа.

4. Если вы работаете над проектом на разных компьютерах, советуем установить флаг **Store Path Relative to Database Library**. В этом случае путь к базе данных при переносе проекта можно будет не устанавливать заново.
5. Нажимаем кнопку **Connected**, и в левом окне появится список доступных листов базы данных, в нашем случае это единственный лист **All**. Внизу в окне появится таблица с перечнем параметров (названий столбцов базы данных).
6. Устанавливаем связь между базой данных и параметром библиотеки. В нашем случае эту связь будет обеспечивать параметр **GlobalLink**.
7. Не будем создавать себе сложностей и оставим это имя параметру и библиотеке.
8. Выбранный параметр для связи будет замаскирован в таблице. Настройка остальных параметров при использовании базы данных только для обновления параметров в схеме подробно описана автором в [2] и к вопросу, освещаемому в этой статье, не имеет отношения.

Инсталляция библиотеки Database Library

Рассмотрим процесс инсталляции Database Library и настройки свойств поиска библиотечных компонентов.

1. Вызываем панель **Library** (рис. 5). Если она еще у вас не настроена, следует в панели **Access** вызвать список **System**, выбрать панель **Library** и установить ее в любом мес-

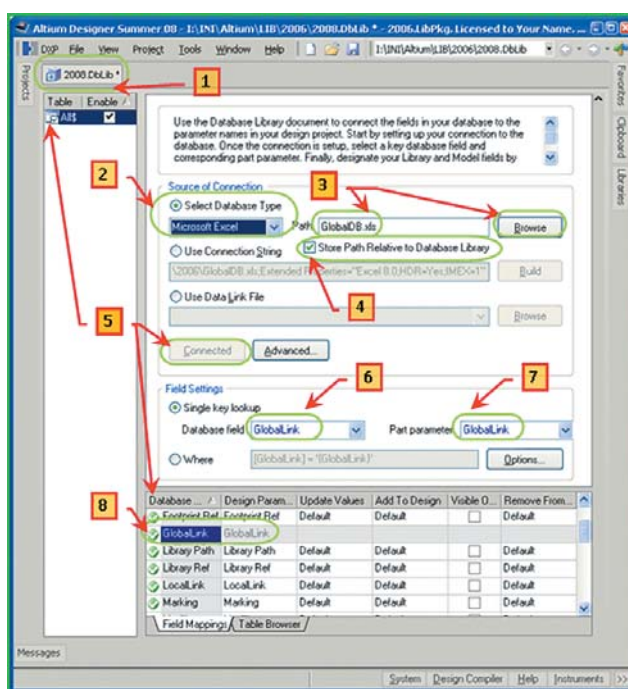


Рис. 4. Подключение базы данных к Database Library

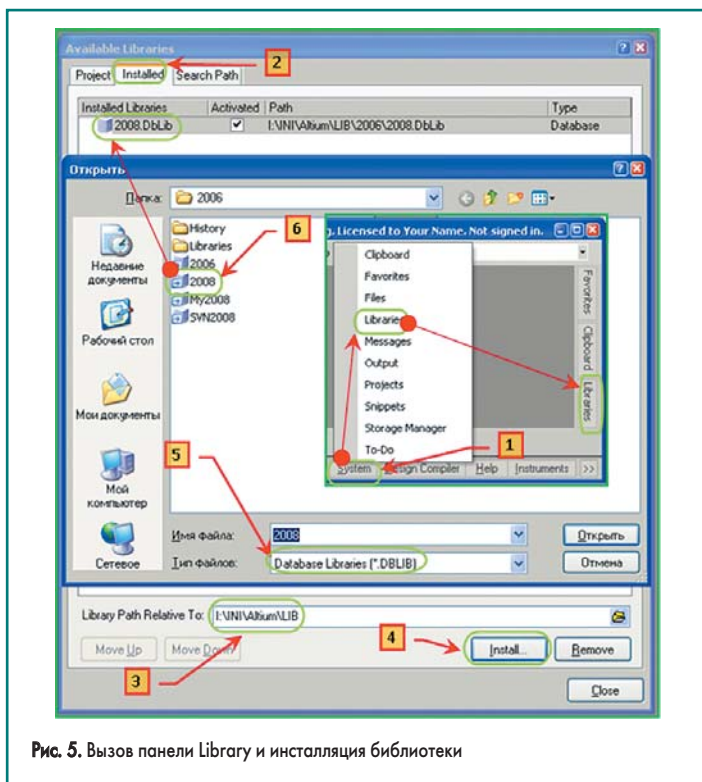


Рис. 5. Вызов панели Library и установка библиотеки

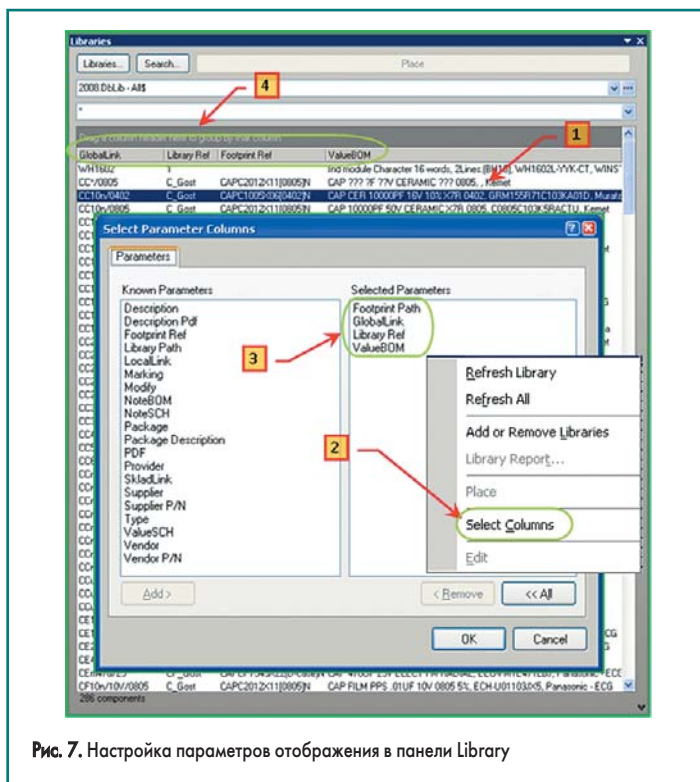


Рис. 7. Настройка параметров отображения в панели Library

- те на экране, либо на одной из сторон общего окна (на рис. 5 — справа).
- Открываем панель **Library**, а в ней — вкладку **Installed**. Поскольку база данных у нас общая, нам нужно установить библиотеку из этой вкладки — тогда библиотека будет доступна всем будущим проектам.
 - Как правило, собственные библиотеки лежат в иной папке, нежели сам пакет Altium Designer, и вы можете указать путь к своим библиотекам, чтобы в следующий раз эта папка открывалась сразу при установке библиотек.
 - Для установки новой библиотеки нажимаем соответствующую кнопку.
 - Указываем тип устанавливаемой библиотеки. В нашем случае это Database Library (*.DbLib).
 - Выделяем устанавливаемые библиотеки (в примере одна, а именно созданная ранее 2008.DbLib). Наша библиотека появится в списке установленных библиотек. Установленный флаг **Activated** означает, что библиотека доступна для любых проектов.

Настройка библиотеки Database Library

На рис. 6 представлен вид панели Library при отображении компонентов из базы данных, где в частности показано:

- Имя (2008.DbLib) библиотеки и имя (All) листа базы данных.

Примечание: если бы в базе данных было несколько листов, они все отображались бы отдельными строками в этом окне.

- В библиотеке специально выбран элемент, указанный в первой строке таблицы 2. Этот элемент носит то же имя, что указано в параметре **GlobalLink**. Однако на настройках формы представления информации в данном окне остановимся далее.

- Это типичный вид изображения компонента для схемы. Изображение взято из библиотеки 2006-RLC.SchLib для компонента C_Gost в полном соответствии с таблицей 2.
- К компоненту подключены две модели:

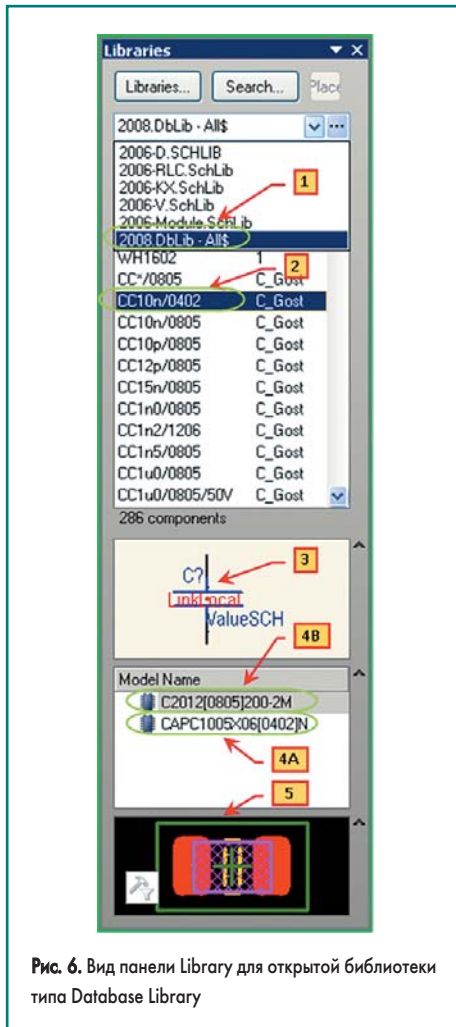


Рис. 6. Вид панели Library для открытой библиотеки типа Database Library

- C2012(0805)200-2M — ссылки на эту модель в базе данных не было. Однако она присутствует в списке, так как была подключена непосредственно в библиотеке 2006-RLC.SchLib к компоненту C_Gost. Отметим: все модели, подключенные непосредственно к компонентам в библиотеках *.SchLib, будут доступны и в библиотеках, сформированных на основе базы данных;
- CAPC1005X06[0402]N. Этой модели не было в исходной библиотеке 2006-RLC.SchLib. К компоненту данная модель подключена дополнительно, в нашем примере это сделано в соответствии с параметрами в таблице 2 из библиотеки 2006new.PcbLib.

Надо признать, что наличие дополнительных моделей, которые подключены к компоненту C_Gost не через базу данных, только осложняет правильный выбор посадочного места. Поэтому при использовании только библиотек типа Database Library рекомендуем вообще отказаться от подключения моделей непосредственно в библиотеке (Schematic Library *.SchLib) изображений посадочных мест. Для подключения в этом случае нужно использовать только записи в базе данных.

- Вид выбранной модели.

В окне (рис. 6, указатель 2) может быть отображена и дополнительная информация. Поскольку на момент выбора компонента еще нет ни отображения его вида, ни его моделей, вполне резонно отобразить в этом окне не только имя GlobalLink, но и имена моделей и, например, параметр ValueBOM (табл. 2). Для этого (рис. 7):

- Переведите указатель на поле списка компонентов базы.
- Правый «клик» мышки вызывает контекстное меню, в котором следует выбрать строку **Select Columns**.

3. В открывшемся окне **Select Parameter Columns** переносим из левой в правую колонку требуемые параметры, например: Footprint Ref, GlobalLink, Library Ref, ValueBOM. Именно эти параметры и будут отображены в данном окне.

4. Располагаем выбранные столбцы в удобном для нас порядке простым перетаскиванием заголовков столбцов.

Теперь список компонентов из базы данных содержит достаточную информацию для выбора требуемого элемента.

Процесс создания и подключения библиотеки на основе базы данных можно считать завершенным. Пользоваться такой библиотекой можно точно так же, как и любой другой. Однако создание и подключение моделей нового компонента стало значительно быстрее и проще. Теперь достаточно добавить еще одну строку к базе данных с нужной информацией. Более того, это не требует перекомпиляции библиотеки.

Примечание. Если вы хотите, чтобы база данных была доступна для редактирования, ее следует открыть прежде загрузки Altium Designer.

На рис. 8 указаны параметры установленного на схему компонента из данной библиотеки. Укажем все особенности нового компонента и те параметры, которые взяты из базы данных.

1. Параметр **Description**. Значение данного параметра взято из соответствующего столбца базы данных.
2. Имя компонента присвоено по параметру **GlobalLink**.
3. Имя библиотеки дает ссылку на файл **2008.DbLib**, который подключает базу данных.

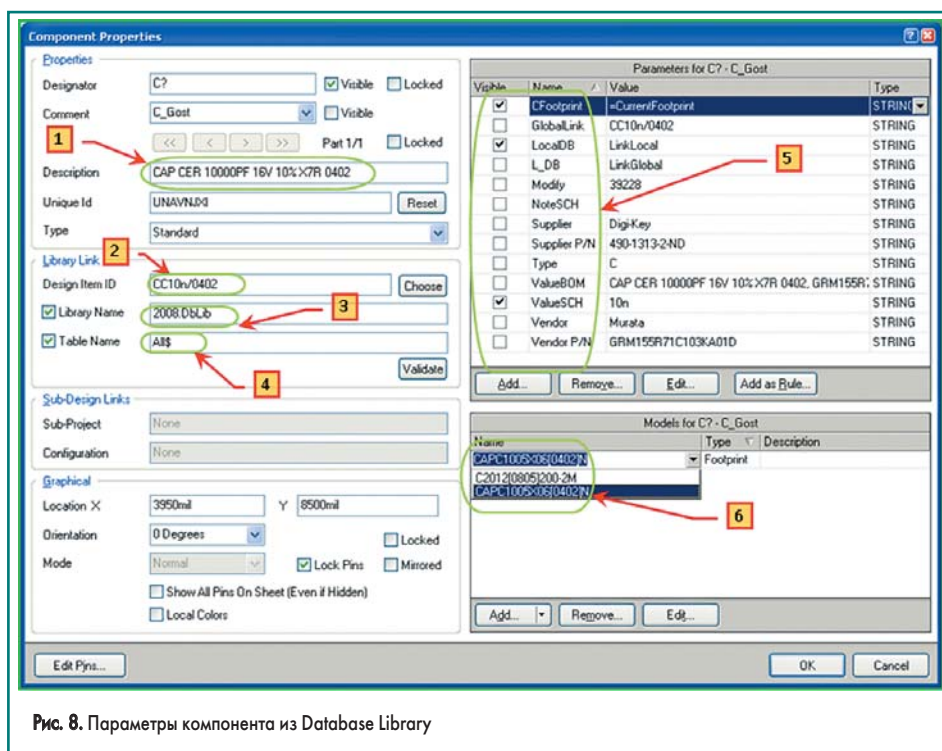


Рис. 8. Параметры компонента из Database Library

4. Имя листа в базе данных. Заметьте, для компонента из «обычной» библиотеки данный параметр был недоступен.
5. Здесь все параметры компонента — и те, что введены непосредственно в библиотеку, и те, что были введены в базу данных. Причем при совпадении имен параметров их значение берется из базы данных.
6. Список подключенных моделей. Здесь также присутствуют и те модели, что подключены в библиотеке, и те, что подключены через базу данных.

Итак, библиотека на основе базы данных успешно создана и подключена. Подключение других моделей или иных параметров можно осуществить аналогично рассмотренным.

Литература

1. Пранович В. Система проектирования Altium Designer 6 // Технологии в электронной промышленности. 2007. № 1.
2. Пранович В. Altium Designer 6 в примерах // Технологии в электронной промышленности. 2007. № 5–8.