



> Altium Vault – ПЕРВОЕ ЗНАКОМСТВО

Сложность современных разработок и скорость развития технологий заставляют инженеров работать с большими объемами данных, которые постоянно меняются. Отслеживать такие изменения пользователю невозможно даже при разработке устройств средней сложности. Многие производители систем автоматизированного проектирования (САПР) пытаются решить эту задачу, но в основном локально — в рамках одной САПР.

Обычно для решения подобных проблем на предприятиях используют PDM-системы, которые достаточно дороги. Кроме того, многие PDM-системы ориентированы на MCAD-системы и без проблем работают только с "родными" для них файлами этих систем. Проекты, разработанные в Altium Designer, для таких систем являются "чужими", поэтому организовать полноценную работу между Altium Designer и внешней PDM-системой достаточно сложно. Если хранение готовых проектов во внешней PDM-системе можно обеспечить, то организовать полноценную работу на уровне компонентов и проектов с учетом всего их жизненного цикла — задача не из легких. Часть подобных задач пользователи Altium Designer решали с помощью контроля версий, внешних баз данных или просто ведя каталоги применяемости и учета проектов. Все эти решения не дают необходимого результата, и говорить об автоматизации и интегрированном решении здесь не приходится.

В связи с этим и была разработана технология Vault, которая является встроенной PDM/PLM-системой для Altium Designer. Обеспечивая организованное хранилище для шаблонов, компонентов, проектов и их жизненных циклов и поддерживая возможности администрирования и контроля версий, Vault позволяет организовать централизованный контроль над данными проектирования в Altium Designer.

В частности, Vault в удобной форме позволяет:

- **организовать полный контроль версий** — он обеспечивает хранение всех эле-

ментов проекта (схем, плат, компонентов, символов и посадочных мест), а также всех этапов их жизненного цикла. Это дает возможность отследить историю изменений и при необходимости выбрать нужную версию для применения в текущем проекте;

- **обеспечить отслеживание применяемости компонента в различных проектах** — это помогает определить входимость схем и плат в какой-либо проект, а также применяемость ЭРИ (электрорадиоизделий) в различных проектах. В случае снятия компонента с производства или отказа от его применения можно найти зависимые проекты и внести в них изменения;

- **обеспечить управление жизненным циклом и для проектов, и для компонентов** — на каждой стадии жизненного цикла с данными работают разные пользователи в соответствии с правами доступа к хранилищу;

- **обеспечить проверку целостности данных** — при сдаче проекта в хранилище, то есть при выпуске проекта, происходит ряд формальных проверок как для схемы и платы. Кроме того, выполняется автоматическая синхронизация проекта, то есть проверка соответствия схемы и платы в проекте, а также применяемости компонентов, которые должны быть утверждены для передачи в хранилище. Если один из этапов проверки завершается неудачей, то проект не может быть сдан в хранилище;

- **обеспечить администрирование** — только администратор может иметь полный доступ к хранилищу и назначать привилегии пользователей и групп пользователей;

- **организовать работу с поставщиками компонентов** — головная боль разработчиков — ведение учета поставщиков для каждого типа компонента — переложена на Vault. Теперь в базе не нужно создавать отдельные компоненты, например, для чип-резисторов, если их производят и поставляют разные организации. Достаточно сделать ссылки для такого

компонента на разные коды в базе поставщика, и информация о производителе и поставщике будет обновляться в зависимости от конкретной ситуации. Это, в свою очередь, обеспечивает актуализацию стоимости изделия и наличия комплектующих для данного проекта на складе.

Использование Vault никак не противоречит использованию общей PDM/PLM-системы на предприятии, которая обеспечивает оборот данных из различных систем проектирования. При таком подходе применение Vault обеспечивает организацию данных для Altium Designer до передачи их в хранилище более высокого уровня. Именно этот подход позволяет эффективно внедрить Altium Designer в систему управления производственными процессами на крупных предприятиях.

На сегодняшний день для работы с хранилищами Vault используются два приложения:

- **Altium Vault Server (AVS)** — полномасштабное корпоративное решение, представляющее собой централизованный сервер для хранения данных. AVS позволяет создать сервер для хранения данных в корпоративной локальной сети. Такой сервер позволяет добавлять пользователей и давать им различные права, определяющие их роль в процессе разработки (жизненном цикле) проекта. Для использования AVS требуется лицензия, которая определяет количество пользователей, имеющих права доступа к работе с хранилищем. Доступ к хранилищу обеспечивается авторизацией пользователя в AVS, которая определяется администратором Vault, в рамках локального сервера. Сервер AVS рекомендуется для крупных компаний и в минимальной конфигурации предоставляет доступ к хранилищу для пяти пользователей. При этом количество пользователей, работающих с сервером, может быть увеличено путем приобретения дополнительных лицензий. Использование AVS наиболее целесообразно на крупных предприятиях,

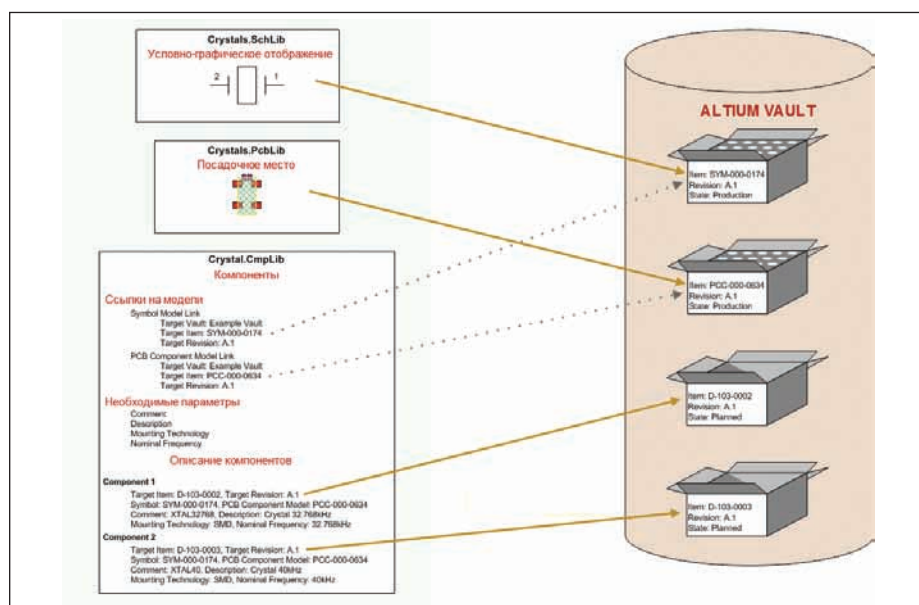


Рис. 1. Схема создания БД ЭРИ в Vault

где процесс производства разбит на большое число этапов и в нем участвует большое число пользователей. В таких ситуациях применение Vault позволит организовать хранение данных о компонентах и проектах, а также управлять этим хранением, разграничивая доступ к хранилищу для разных специалистов. Возможность отслеживания применяемости и поставки позволяет всегда иметь актуальную информацию о компонентах проекта, что в итоге позволяет существенно экономить время на разработку. Имеющаяся в основе Vault система контроля версий обеспечивает многопользовательскую работу над проектом;

■ **Altium Personal Vault (APV)** — упрощенная модель хранилища, которая в плане работы с данными не уступает AVS. Отличие этого хранилища заключается лишь в отсутствии административных возможностей. Работа с APV не требует наличия отдельных лицензий, но требует наличия действующей подписки на Altium Designer. Количество пользователей, которые могут работать с хранилищем APV, должно быть не более четырех, при этом для работы с хранилищем не требуется авторизация. Использование APV не предполагает администрирование сервера, то есть нет возможности создавать пользователей и группы пользователей: все пользователи APV — равноправны. Такое хранилище наиболее подходит небольшим предприятиям, где отсутствует дифференцированный подход к разработке и вся работа

над проектом выполняется одним пользователем.

Остановимся подробнее на некоторых технических аспектах работы с хранилищем Vault и порядке взаимодействия с ним со стороны пользователя в Altium Designer. Здесь сразу стоит различать два варианта взаимодействия — хранение компонентов и хранение проектов. По большому счету, эти два варианта взаимосвязаны, и хранение проекта требует того, чтобы все используемые в проекте компоненты были предварительно зарегистрированы в хранилище и имели определенный статус.

Для хранения компонентов в Vault необходимо создать библиотеки условно-графических отображений (УГО) (SCHLIB) и посадочных мест (PCBLIB) в Altium Designer и зарегистрировать их в хранилище (рис. 1). После этого необходимо создать библиотеку компонентов (CMPLIB), которая раньше отсутствовала в Altium Designer и появилась только с развитием идеологии Vault. Такая библиотека должна быть привязана к хранилищу, после чего в ней будут доступны нужные УГО и посадочные места. Таким образом, достигается полный контроль версий над всеми объектами базы индивидуально (УГО, посадочные места и компоненты), а также обеспечивается повторная применяемость как для УГО, так и для посадочных мест.

При создании библиотеки компонентов в Vault база данных имеет удобный для пользователя интерфейс (рис. 2). Здесь пользователь, который является администратором базы, выполняет следующие операции (на рис. 2 они помечены цифрами):

1. **Задаёт место хранения базы в одном из разделов Vault.** Здесь же указывается описание жизненного цикла компонента, в соответствии с которым будет происходить утверждение компонента и выпуск его в эксплуатацию.
2. **Определяет набор параметров,** который описывает все компоненты этой базы. Эти параметры впоследствии будут использоваться при формировании конструкторской документации, а также для формирования запроса на поиск компонента в базе.
3. **Добавляет набор символов (УГО) и посадочных мест,** которые могут быть использованы в базе. Такой ограничительный список позволяет иметь максимальный контроль над базой, давая возможность в одном окне отобразить все применяемые объекты.
4. **Просматривает УГО и посадочные места** для проверки корректности отображения нужной версии.
5. **Задаёт названия компонентов и заполняет соответствующие атрибуты.** Удобство заполнения атрибутов обеспечивает возможность создания групп компонентов по типовым параметрам. Например, для резисторов можно создать группы по типу корпуса, по номиналу, по мощности и т.д. Создание группы определяет отдельный раздел в базе данных и позволяет одним действием заполнить одинаковый атрибут для группы. Такие атрибуты, как поставщик и производитель, не задаются в библиотеке и, по сути, могут вообще не интересовать разработчика.

Со стороны пользователя основанная на Vault база данных компонентов выглядит организованной структурой (рис. 3).

Здесь имеется возможность поиска необходимого компонента путем выбора его из структуры базы [1] или с помощью типового поиска в базе данных [2]. Для компонентов в хранилище можно указать код по базе поставщика [3], который влечет за собой загрузку всей необходимой информации об этом компоненте, в том числе наименования поставщика и производителя. Для каждого компонента можно указать необходимое количество таких кодов, при этом на стадии закупки будет выбран любой из аналогичных элементов. Для каждого из указанных поставщиков сразу отображается актуальная цена [4]. При необходимости можно изменить режим отображения данных о компоненте [5]. Например, вместо выбранного режима *Supply Chain* (см. 5 на рис. 3) можно указать другие ре-

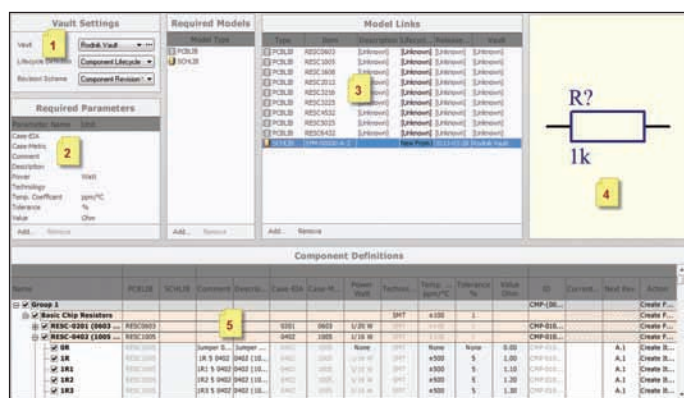


Рис. 2. Интерфейс для создания компонентов в БД Vault

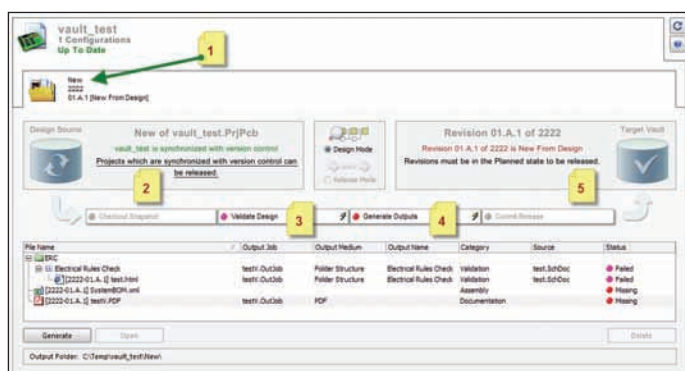


Рис. 4. Процесс сдачи проекта в хранилище Vault

жимы, просмотреть все содержимое компонента и его применяемость в различных проектах.

При регистрации проекта в Vault он имеет определенный статус *Planned*, который будет изменяться в рамках жизненного цикла проекта. Конечным статусом проекта должен быть статус *Released*, то есть проект, сданный в хранилище и прошедший необходимый набор проверок. Для загрузки проекта в хранилище используется специальный режим работы с проектом *View* → *PCB Release View* (рис. 4).

В этом режиме выполняются все настройки проекта для дальнейшей сдачи его в хранилище в следующем порядке:

1. **Главные настройки проекта**, такие как место хранения проекта в структуре хранилища, наименование конфигурации, которая определяет статус проекта (разработка, производство и др.) и уровень его проверки, а также файл настроек для проверки.
2. **Контроль версий (Checkout Snapshot)**. На первой стадии передачи проекта в Vault требуется, чтобы все файлы проекта были зарегистрированы в системе контроля версий. Это обеспечивает

соблюдение важных требований: сохранение текущей версии всех файлов проекта и фиксацию изменений в системе контроля версий. При этом стоит помнить, что при регистрации контроля проекта в системе контроля версий регистрируется вся директория с проектом, поэтому не рекомендуется хранить в этой директории не относящиеся к проекту файлы.

3. **Проверка проекта (Validate Design)**. На второй стадии происходит проверка всех правил, которые ранее были предварительно настроены в файле OUTJOB и указаны в главных настройках [1]. К таким правилам относятся проверка схемы (Electrical Rule Check – ERC), проверка платы (Design Rule Check – DRC), проверка синхронизации между платой и схемой, а также проверка посадочных мест. Если хотя бы один пункт проверки завершается неудачей, дальнейший процесс невозможен. Например, на рис. 4 не прошла проверка ERC, поэтому передача проекта в хранилище была остановлена.

4. **Генерация выходных файлов (Generate Outputs)** – формирование выходной

документации, описанной в файле OUTJOB. При этом в соответствующих разделах хранилища будут сохранены всевозможные чертежи, отчеты и файлы для производства.

5. **Выпуск проекта (Commit Release)** – сохранение новой версии проекта в хранилище. Изменение сохраненной версии проекта будет невозможно, пока проект находится в режиме *Release Mode*, то есть до тех пор, пока вы не переключите его в режим *Design Mode*. Выпущенный проект можно сдать в общую PDM-систему предприятия (если она есть) для учета в едином документообороте по изделию.

Подводя итог, хотелось бы отметить, что использование Vault позволяет организовать работу над проектами, выполненными в системе Altium Designer. Многие из упомянутых задач, решение которых возлагается на Vault, можно решить альтернативными способами, но для этого в каждой задаче будет использоваться свой инструментарий. Здесь все управление от базы данных ЭРИ до проектов возлагается на единый инструментарий, формализующий многие процессы, которые ранее выполнял пользователь. Такая формализация обязывает пользователя не пропускать значимые этапы верификации проекта и позволяет ему не вести учет сохраненных файлов, так как обеспечение целостности и актуальности данных проекта – задача Vault. Использование единого хранилища Vault позволяет ввести администрирование в процесс разработки и хранения данных Altium Designer, что, в свою очередь, автоматизирует контроль и снижает роль "человеческого фактора" при проверке корректности и целостности проектов.

Алексей Сабунин,
к.ф.н., технический эксперт
отдела САПР ЗАО "НПП "РОДНИК"
E-mail: sabunin@rodnik.ru

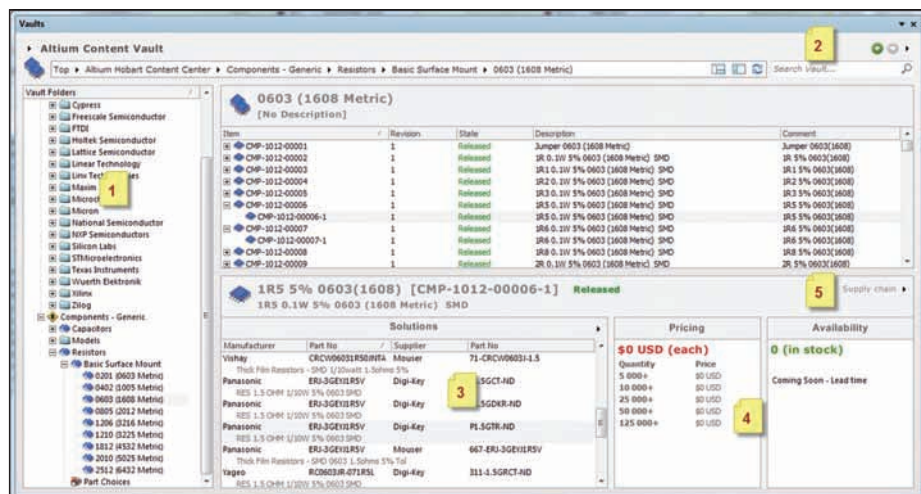


Рис. 3. Интерфейс доступа к компонентам БД в Vault