

# Altium Concord Pro.

## Управление компонентами

Павел Демидов, Application Engineer, Altium (pavel.demidov@altium.com)  
Александр Фень, Field Application Engineer, Altium (alexander.fen@altium.com)

Компоненты являются основой любого электронного устройства на базе печатных плат, поэтому на этапе проектирования таких устройств крайне важно обеспечить пользователя САПР удобными инструментами управления компонентами, их поиска и создания, контроля их корректного применения в проектах. В данной статье будут рассмотрены функциональные возможности серверного программного продукта **Altium Concord Pro**, предназначенного для управления компонентами, используемыми при проектировании электронных устройств в САПР *Altium Designer*.

### Понятие управляемого компонента в Altium Concord Pro

Исторически *Altium Concord Pro* является развитием продукта *Altium Vault*, и, как и его предшественник, он в первую очередь предназначен для систематизации и надежного хранения данных проектирования, создаваемых в *Altium Designer*: компонентов, проектов, шаблонов, выходных данных и т.д. [1, 2]. Управление компонентами – это одна из основных задач, решаемых средствами *Altium Concord Pro*.

На сегодняшний день предлагается два варианта развертывания *Altium Concord Pro*: в облаке (на платформе *Altium 365* в виде выделенного *Workspace*), либо на собственных ресурсах компании, которая приобрела *Concord Pro* (“самоуправляемый” вариант размещения). Рассмотренные далее возможности по управлению компонентами идентичны для обоих вариантов.

Управляемый компонент, как следует из названия, – это компонент, который управляется сервером *Concord Pro*, то есть сервер отвечает за хранение этого компонента и всех связанных с ним данных, за управление жизненным циклом ревизий этого компонента, за валидацию компонента при его создании и отправке на сервер, за связь компонента с цепочкой поставок. В отличие от традиционной парадигмы хранения компонентов в файловых библиотеках *Altium Designer* (таких, как отдельные библиотеки *SchLib/PcbLib* и интегрированные библиотеки *IntLib*), где в основе компонента лежит его схемный символ, а все остальные данные (параметры, модели, ссылки) “подключаются” к этому символу, в основе управляемого компонента лежит серверный объект – *Component*

*Item*, который объединяет в себе все данные, определяющие компонент. Эту концепцию хорошо отражает интерфейс инструмента создания отдельного управляемого компонента – редактора *Component Editor* – в режиме *Single Component Editing*. Каждая область рабочего окна относится к какой-то группе данных, определяющих компонент (рис. 1):

#### 1 Component (Компонент)

В этой области задаются ключевые свойства компонента, в том числе его название, описание, уникальный идентификатор, папка, тип и ссылка на шаблон компонента.

#### 2 Models (Модели)

Здесь добавляются ссылки на доменные модели, используемые компонентом.

#### 3 Parameters (Параметры)

Здесь определяется набор параметров, которые будут применены к компоненту. Этот набор может включать в себя параметры из шаблона компонента, а также дополнительные пользовательские параметры; можно также добавить ссылки на веб-сайты и даташиты.

#### 4 Part Choices (Варианты выбора компонента)

Здесь вы можете осуществить поиск и добавить варианты выбора компонента (*Part Choices*) – физические компоненты от производителей, которые могут использоваться при изготовлении печатной платы и печатного узла. Данные от производителей могут быть взяты как из глобального источника

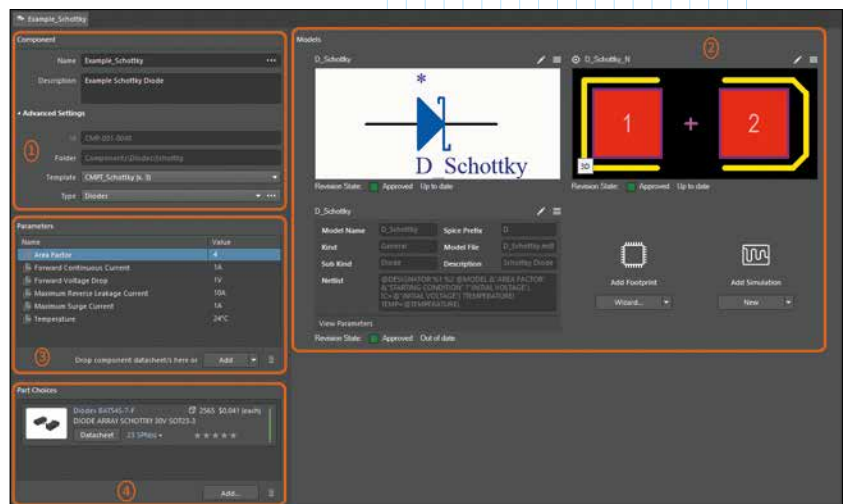


Рис. 1. Интерфейс редактора *Component Editor* в режиме *Single Component Editing*

(Altium Parts Provider), так и из локальной базы данных предприятия.

Каждое такое определение компонента сохраняется в объект *Component Item* в виде отдельной ревизии. Жизненный цикл ревизий компонентов (как и любых других серверных объектов *Concord Pro*) находится под управлением системы; каждое их состояние может быть определено как допустимое или недопустимое для применения в проектах. Таким образом, формируются наборы утвержденных компонентов, которые могут быть повторно использованы в новых проектах, в прототипах и в готовых изделиях.

Помимо создания отдельного компонента, существует интерфейс для определения сразу множества компонентов: редактор *Component Editor* в режиме пакетного редактирования – *Batch Component Editing* (рис. 2). В этот режим можно переключиться из режима редактирования отдельного компонента, либо открыв на изменение множество управляемых компонентов, – тогда этот интерфейс откроется автоматически. У каждого определения компонента будет общий набор параметров и ссылок на нужные доменные модели. Пакетное редактирование будет полезным, когда имеет смысл управлять компонентами как набором – например, набором чип-резисторов.

## Типы и шаблоны компонентов

Для систематизации хранимых на сервере *Concord Pro* компонентов и унификации процесса их создания предусмотрены специальные средства.

Как упоминалось выше, при создании компонента определяется его тип, который и является ключевым атрибутом, используемым для обеспечения целевого поиска по управляемым компонентам. На сервере может быть задана сколь угодно сложная иерархия типов компонентов, что необходимо для их разделения по категориям, которые будут отражены в панели компонентов (об этом подробнее сказано далее).

Тип и другую относящуюся к компоненту информацию можно заранее указать в шаблоне компонента. В частности, шаблоны компонентов были разработаны с учетом информации о параметрах. Это позволяет в едином расположении определить набор тех параметров, которые станут доступны в компоненте, ссылающемся на этот шаблон. С помощью шаблона можно управлять тем, какие из этих параметров должны быть видимы, задать значения по умолчанию и даже обозначить параметры как обязательные. Чтобы обеспечить соответствие компонента принятым на предприятии требованиям, во время выпуска

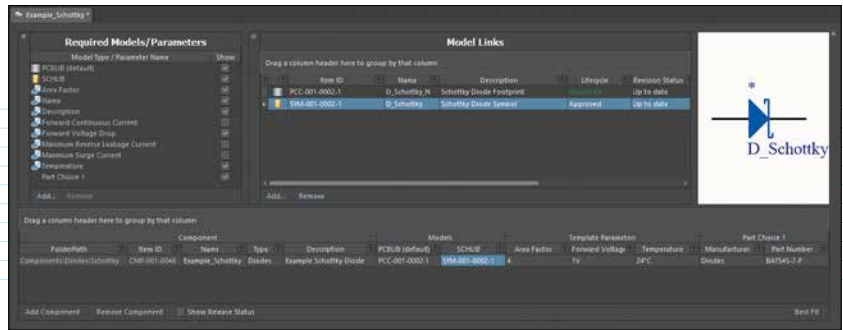


Рис. 2. Интерфейс редактора *Component Editor* в режиме *Batch Component Editing*

компонента (записи на сервер) осуществляются валидационные проверки.

Кроме того, редактор шаблонов компонентов (его интерфейс показан на рис. 3) позволяет явно задавать типы данных параметров компонентов с учетом их единиц измерения. Поддерживаются все распространенные префиксы единиц измерения. Вы можете вводить значение параметра в удобном для вас формате (к примеру, *2.2k*, *2k2*, *4M*, *2.5GHz*) – сервер сумеет распознать числовые значения таких записей. Это дает возможность, например, осуществить поиск компонентов по диапазону параметров с целью их размещения в проекте.

## Жизненный цикл компонента

Каждый объект на сервере *Concord Pro*, в том числе управляемый компонент, состоит из набора ревизий, где каждая новая ревизия используется для сохранения новых данных, которые изменяются и выпускаются на сервер. Таким образом, ревизии объекта отражают прогресс изменений этого объекта. Важно показать и текущее состояние всех ревизий объекта – на какой стадии своего жизненного цикла (*Lifecycle*) находится ревизия. Для каждого состояния можно указать, допустимо ли использовать в проекте ревизию объекта в этом состоянии или нет. Например, если ревизия компонента перешла в состояние *Obsolete* (устаревшая),

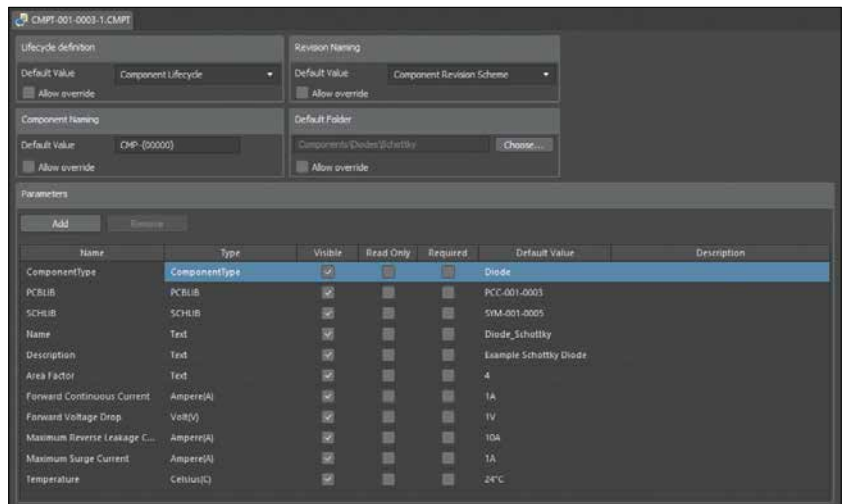


Рис. 3. Интерфейс редактора *Component Template Editor*

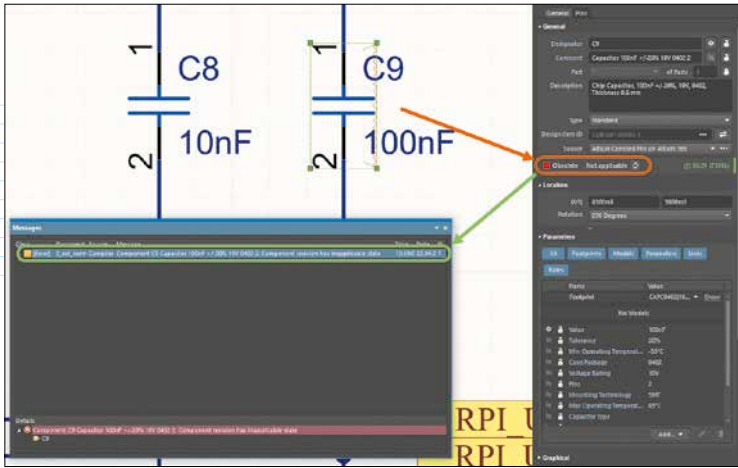


Рис. 4. Валидация проекта с компонентом в неприменимом состоянии

она считается неприменимой – это будет отражено в свойствах компонента и в панели сообщений при валидации проекта (рис. 4). Такой проект будет невозможно выпустить (записать на сервер).

### Доменные модели

С точки зрения проектировщика, управляемый компонент объединяет в рамках единой сущности всю информацию, необходимую для представления этого компонента во всех областях проектирования. В этом отношении его можно рассматривать как контейнер, “корзину”, в которой хранятся все параметры и доменные модели: схемный символ (условно-графическое обозначение), посадочное место компонента, имитационные модели.

Касательно представления компонента в различных областях проектирования, управляемый

компонент содержит в себе не сами доменные модели, а только ссылки на соответствующие объекты моделей, которые также хранятся на сервере управляемых данных. Соответственно, эти модели обладают всеми возможностями и преимуществами серверных объектов *Concord Pro*: контроль ревизий, управление жизненным циклом, отслеживание применимости и т.д.

### Поиск и отслеживание применимости компонентов

Поиск управляемых компонентов и их размещение в проекте осуществляется через панель *Components* в *Altium Designer*. Эта панель является эффективным интерфейсом, который позволяет взаимодействовать со всеми компонентами – как управляемыми (серверными), так и библиотечными (файловыми). Для управляемых компонентов, каждая “библиотека” представляет собой группу компонентов определенного типа (или подтипа, если на сервере создана более сложная структура компонентов), что позволяет ограничить отображаемый список только компонентами соответствующего типа (рис. 5а). Кроме того, для управляемых компонентов, размещенных на сервере *Concord Pro*, панель предоставляет возможность параметрического поиска на основе фильтров (фасетного поиска) с помощью указания параметров целевых компонентов (рис. 5б).

Помимо прочего, для выбранного в панели *Components* компонента отображается, в каких проектах он уже используется (рис. 6), что может быть полезным для оптимизации номенклатуры применяемых на предприятии компонентов при разработке новых изделий.

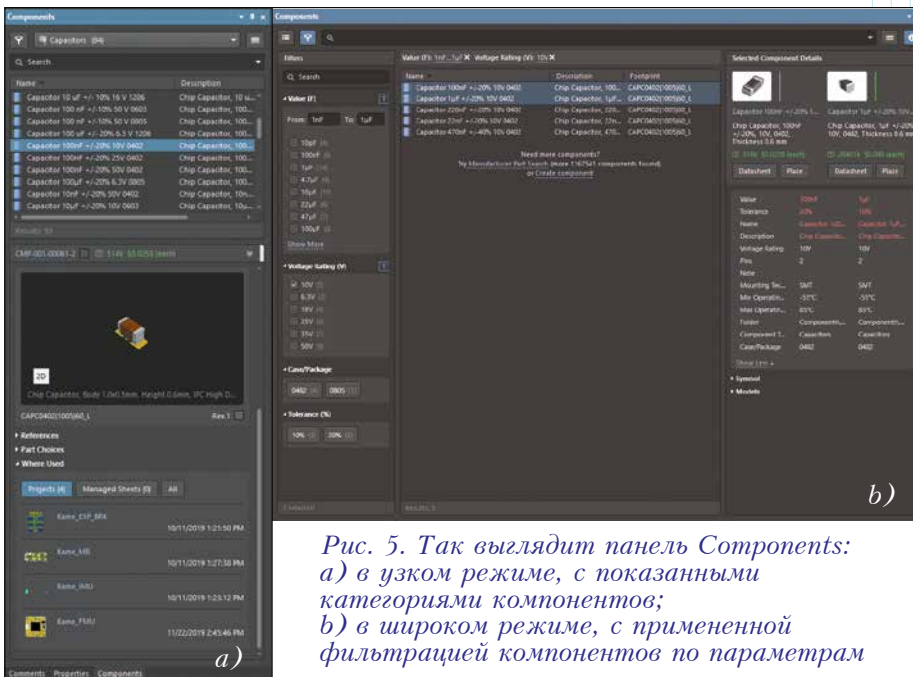


Рис. 5. Так выглядит панель *Components*: а) в узком режиме, с показанными категориями компонентов; б) в широком режиме, с примененной фильтрацией компонентов по параметрам

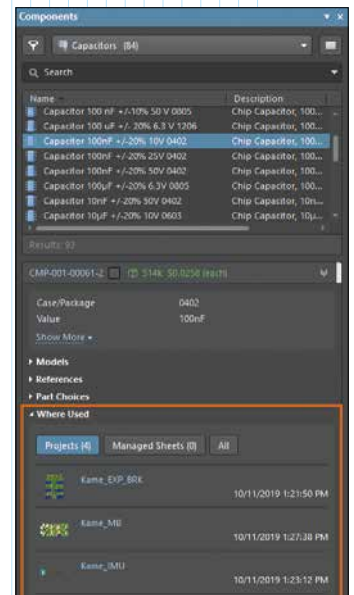


Рис. 6. Область *Where Used* панели *Components*



## Запрос отсутствующего компонента

Количество компонентов, доступных инженеру при проектировании платы, может варьироваться от нескольких сотен элементов, распределенных по отдельным библиотекам символов и моделей, до сотен тысяч (в этом случае они обычно хранятся в специальной базе данных компании). Тем не менее, независимо от того, сколько доступно сейчас, всегда будут возникать ситуации, когда нужные компоненты отсутствуют, и их нужно создать и сохранить для повторного использования в проектах.

В таких случаях в небольших компаниях инженер просто на время примеряет на себя роль библиотекаря и сам создает недостающие компоненты. Однако в больших организациях, где есть специальный отдел для расширения и поддержки библиотеки компонентов, имеет смысл отправлять запросы на новые компоненты в этот отдел. Для этого система *Altium Designer* в сочетании с *Altium Concord Pro* предлагает функциональную возможность запроса компонентов – *Part Requests*.

Инженер создает и отправляет запрос на создание одного или множества компонентов, который получают пользователи сервера, определенные как библиотекари. После этого ему приходит уведомление о том, когда будет завершена обработка запроса и компоненты станут доступны, либо о том, что запрос отклонен (и почему). Пользователь, создающий запрос, предоставляет всю доступную ему информацию (номера компонентов у производителей и поставщиков, описания, даташиты в виде *PDF* и *URL*) и может даже подготовить для библиотекаря заготовки объектов компонентов, с которыми тот сможет начать работу.

Возможность создания запроса доступна как из среды *Altium Designer*, так и через веб-интерфейс сервера (рис. 7).

## Миграция в *Concord Pro* существующих библиотек

Как правило, к моменту перехода на серверные решения для управления проектными данными (такие, как *Altium Concord Pro*) у предприятия уже имеются файловые библиотеки компонентов предыдущего поколения (*SchLib*, *PcbLib*, *IntLib*, *DbLib*, *SVNDbLib*). В этом случае можно осуществить миграцию, то есть преобразовать их в серверные управляемые компоненты с помощью специального средства – *Library Migrator*.

*Library Migrator* предлагает два режима интерфейса.

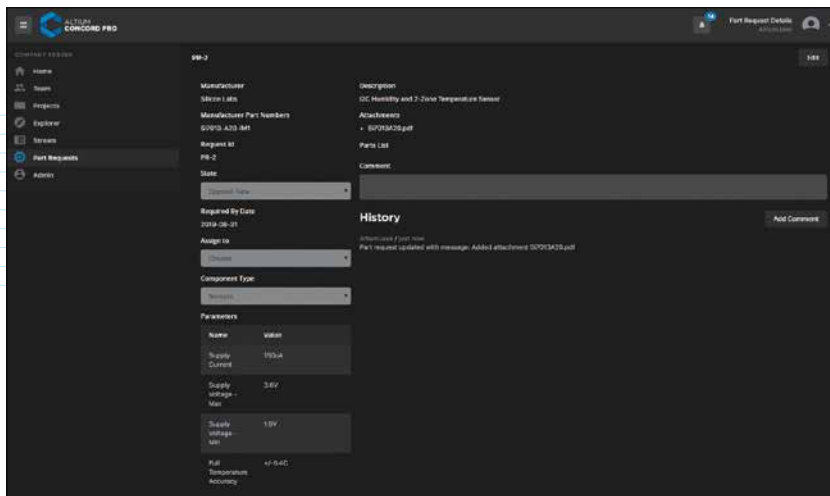


Рис. 7. Пример запроса компонента в веб-интерфейсе *Altium Concord Pro*

Когда используется минималистичный, простой режим (*Simple*), миграция выбранных файловых библиотек компонентов в серверные управляемые компоненты происходит за один этап. При этом средство миграции берет на себя классификацию по типам, назначение целевых папок, добавление параметров и типов значений, а также передачу всех связанных данных. Кроме того, предусмотрен расширенный режим (*Advanced*), в котором обеспечивается полный предварительный просмотр планируемой миграции библиотек, доступны все соответствующие данные и настройки (рис. 8). И хотя миграция по умолчанию осуществляется в один клик, *Library Migrator* предлагает и расширенные настройки в панели свойств для лучшего контроля над тем, как будет осуществляться процесс.

## Заключение

Подводя итог, можно выделить ряд преимуществ управляемых компонентов, которые позволяют повысить производительность при разработке электронных устройств, избежать ошибок и ускорить вывод качественных электронных изделий на рынок. В их числе:

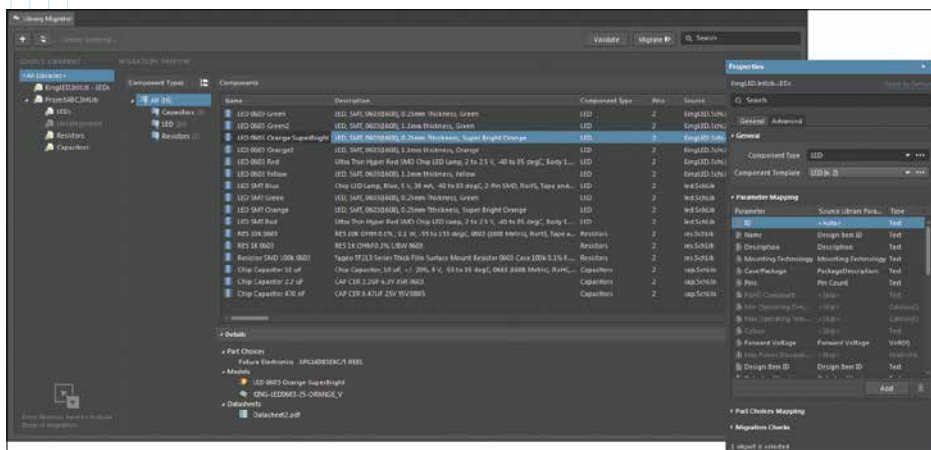


Рис. 8. Интерфейс *Library Migrator* в расширенном режиме

✓ **Формирование набора утвержденных для проектирования компонентов**

Применяемые на предприятии компоненты записываются (выпускаются) на сервер управляемых данных для их повторного размещения в проектах. Возможность контроля ревизий и состояний жизненного цикла позволяет предприятию создать набор утвержденных компонентов, которые формально могут быть использованы проектировщиками.

✓ **Выбор физических компонентов на этапе проектирования**

Для любого управляемого компонента можно выбрать, какие именно физические компоненты могут использоваться при установке на плате.

✓ **Динамическая информация о цепочке поставок**

Актуальные данные из агрегированной базы данных компонентов *Altium Parts Provider*, взаимодействующей с выбранными поставщиками и собирающей от них информацию, позволяют проектировщику знать о текущей стоимости и доступности выбранных компонентов и их поставщиков.

✓ **Шаблоны компонентов**

Основанные на параметрах и классификации компонентов шаблоны обеспечивают, чтобы каждый новый компонент определенного типа включал в себя корректный набор параметров, совместимый

с составом изделия, а также корректные схемы именованья ревизий и жизненных циклов.

✓ **Отслеживаемость использования компонентов**

Управляемые компоненты можно отследить на всех этапах их использования. Если компонент становится устаревшим, легко выяснить, в каких проектах он был задействован и какие проекты следует обновить. Если символ или посадочное место содержит в себе ошибку, можно найти все компоненты, которые используют их, и внести необходимые исправления.

✓ **Валидация жизненного цикла компонентов**

Если компонент находится в состоянии “конец срока службы”, “устарел” или “заменен”, пользователь будет предупрежден об этом до того, как попытается изготовить плату с этим компонентом.

✓ **Прямое редактирование компонентов**

При необходимости отредактировать управляемый компонент его можно открыть на изменение непосредственно на сервере управляемых данных. 🐼

**Список литературы**

1. Фень А. Обзор *Altium Concord Pro* // Современная электроника, 2020, №2.
2. Документация *Altium Concord Pro* // [www.altium.com/ru/documentation/altium-concord-pro](http://www.altium.com/ru/documentation/altium-concord-pro)

◆ Выставки ◆ Конференции ◆ Семинары ◆

ASTANA

# Powerexpo

9-я КАЗАХСТАНСКАЯ МЕЖДУНАРОДНАЯ ВЫСТАВКА

## ЭНЕРГЕТИКА

ЭЛЕКТРОТЕХНИКА И ЭНЕРГЕТИЧЕСКОЕ МАШИНОСТРОЕНИЕ



www.POWEREXPOASTANA.kz



МВЦ “EXPO”,  
Нур-Султан, Казахстан

## 4 | 5 | 6 ноября 2020

  
ВИЭ

  
СВЕТОТЕХНИКА

  
КАБЕЛЬ И ПРОВОД

  
ЭЛЕКТРОТЕХНИКА

  
ЭНЕРГОСБЕРЕЖЕНИЕ

**ОРГАНИЗАТОРЫ:**

 Итека  
Тел.: +7 (7172) 58-34-34;  
E-mail: snezhana.s@iteca.kz

**ОФИЦИАЛЬНАЯ ПОДДЕРЖКА:**

  
Министерство по инвестициям  
и развитию Республики Казахстан

  
KEGOC

  
Министерство энергетики  
Республики Казахстан

  
CMK

  
Торговое представительство  
Российской Федерации  
в Казахстане