|  |
| --- |
| [Название организации] |
| Библиотека общих решений интерфейса пользователя ALX.Common.UI |
| Документация к сборке |

|  |
| --- |
| Сорочан Алексей Александрович  13.1.2023 |

Оглавление

[Общие сведения 2](#_Toc124512633)

[Структура сборки 2](#_Toc124512634)

[Dialog 2](#_Toc124512635)

[MessageBox 3](#_Toc124512636)

[SplashScreen 4](#_Toc124512637)

[Controls 5](#_Toc124512638)

[ErrorControl 5](#_Toc124512639)

[Forms 6](#_Toc124512640)

[LoadingForm 6](#_Toc124512641)

# Общие сведения

Сборка представляет собой набор классов, для работы с интерфейсом пользователя. Основная цель создания данной библиотеки классов заключается в обобщении и упрощении вызова диалогов с пользователем, к примеру таких как сообщения (MessageBox) и диалоги (Dialog).

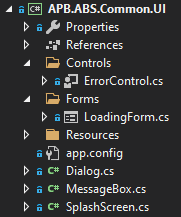
Также следует отметить, что библиотека классов поддерживает [HTML-форматирование](https://docs.devexpress.com/WindowsForms/4874/common-features/html-text-formatting?v=19.1) для отображения текста сообщений. Для форматирования доступно:

* Изменение стиля внешнего вида (SkinName). Значение по умолчанию «Office 2010 Blue»;
* Изменение шрифта текста и его размера (FontName и FontSize). Значение по умолчанию «Tahoma, 10».
* Текст кнопок диалога, к примеру таких как «Да», «Нет», «Отмена» и другие.

Значения по умолчанию могут быть изменены в разделе файла конфигурации «ALX.Common.UI.Properties.Settings».

# Структура сборки

На рисунке ниже изображена структура классов данной библиотеки.



Классы и компоненты будут более подробно рассмотрены ниже.

## Dialog

Класс, который служит для отображения компонентов интерфейса пользователя (в том числе пользовательских) в виде диалогового окна. Методы класса, которые доступны для вызова извне, однотипны, отличаются лишь названием, которое отражает то, какие кнопки результата диалога будут предложены пользователю для взаимодействия. Каждый из таких методов принимает два параметра:

1. Control. Компонент пользовательского интерфейса, который будет отображен в диалоге;
2. Title. Строка заголовка диалога.

Далее переданный компонент отображается при помощи класса [XtraDialog](https://docs.devexpress.com/WindowsForms/DevExpress.XtraEditors.XtraDialog?v=19.1). Одна из перегрузок метода Show принимает класс аргументов, который генерируется в privet методе класса Dialog. К аргументам диалога относятся такие параметры отображения как:

* Применение настроенного стиля отображения;
* Присваивание набора кнопок диалога;
* Установка заголовка диалогового окна;

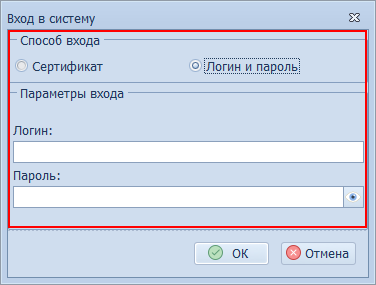
Также следует отметить, что для аргументов диалога добавляется обработчик события отображения, в котором переопределяются параметры кнопок диалога (иконка, текст, ширина, выравнивание).

Результатом вызова такого метода является элемент перечисления [DialogResult](https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.dialogresult?view=netframework-4.7.2&f1url=%3FappId%3DDev16IDEF1%26l%3DEN-US%26k%3Dk(System.Windows.Forms.DialogResult)%3Bk(TargetFrameworkMoniker-.NETFramework%2CVersion%253Dv4.7.2)%3Bk(DevLang-csharp)%26rd%3Dtrue).

Ниже представлен пример обращения к одному из методов класса Dialog.

|  |
| --- |
| private void loginButton\_Click(object sender, EventArgs e)  {          LoginControl control = new LoginControl();          DialogResult dialogResult = Dialog.CallDialogOkCancel(                  control: control, title: "Вход в систему");  } |

Результат обращения к классу Dialog изображен на рисунке ниже. Где красной рамкой выделен пользовательский компонент (LoginControl) из примера.



## MessageBox

Класс является надстройкой для отображения сообщений, привычных [MessageBox](https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.forms.messagebox?view=windowsdesktop-7.0) или [XtraMessageBox](https://docs.devexpress.com/WindowsForms/114566/controls-and-libraries/messages-notifications-and-dialogs/xtramessagebox?v=19.2). Помимо вывода текста сообщений в формате HTML, так как это было [описано выше](#_Общие_сведения), методы класса в сообщении отображают иконку, соответствующую типу сообщения: информация, предупреждение или ошибка.

Аналогично подходу, который применяется для отображения диалогов в классе [Dialog](#_Dialog), отображение сообщения происходит по переданным в метод, заголовку и тексту сообщения, через подготовку класса аргументов, который принимает одна из перегрузов метода Show для класса [XtraMessageBox](https://docs.devexpress.com/WindowsForms/114566/controls-and-libraries/messages-notifications-and-dialogs/xtramessagebox?v=19.2). Для каждого из типа сообщения предопределены заголовки по умолчанию, который можно переопределить при необходимости.

В данном классе, также реализован набор методов с вопросом и вариантами выбора действия для пользователя:

1. QuestionWarningOkCancel – Диалог-предупреждение с вопросом (ОК, Отмена);
2. QuestionWarningYesNo – Диалог-предупреждение с вопросом (Да, Нет);
3. QuestionSendCancel – Диалог с вопросом (Отправить, Отмена);
4. QuestionErrorYesNo – Диалог-ошибка с вопросом (Да, Нет);
5. QuestionYesNo – Диалог с вопросом (Да, Нет);
6. QuestionYesNoCancel – Диалог с вопросом (Да, Нет, Отмена).

Ниже представлен пример обращения к одному из методов класса Dialog.

|  |
| --- |
| private void removeButton\_Click(object sender, EventArgs e)  {      DialogResult dialogResult = MessageBox.QuestionYesNo(              caption:$"Отчет: <u>{report.Name}</u>{Environment.NewLine}" +                      $"Состояние отчета: <u>{report.State.GetEnumDescription()}</u>{Environment.NewLine}" +                      $"{Environment.NewLine}Удалить отчет?");          if (dialogResult != DialogResult.Yes) return;          RemoveReport(report);  } |

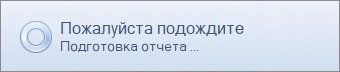
## SplashScreen

Данный класс применяется для отображения окна загрузки с анимацией (спиннер), где применяется привычный [Wait Form](https://docs.devexpress.com/WindowsForms/10824/controls-and-libraries/forms-and-user-controls/splash-screen-manager/wait-form?v=19.2) с некоторыми доработками. Так, к примеру, в методе вызова окна загрузки заданы значения по умолчанию для заголовка и текста окна. Также отличительной особенностью от стандартного решения DevExpress является то, что в последнем случае, для вариантов перегрузки метода [ShowForm](https://docs.devexpress.com/WindowsForms/devexpress.xtrasplashscreen.splashscreenmanager.showform.overloads?v=19.2), где указывается достаточное количество входных параметров, в обязательном порядке должно быть указано родительское окно (owner), тогда как в классе SplashScreen, в том случае если родитель не указан, создается скрытая форма. Что удобно и применимо в тех случаях, когда нет возможности указать родительское окно для экрана загрузки. Также следует отметить, что для метода ShowWaitForm, заданы значения по умолчанию для входных аргументов: заголовок и текст.

Пример создания анимации загрузки приведен ниже.

|  |
| --- |
| private void openReportButton\_Click(object sender, EventArgs e)  {          try          {                  // вызов загрузки данных                  SplashScreen.ShowWaitForm();                  // подготовка данных для отображения                  SplashScreen.SetWaitFormDescription("Подготовка отчета");          }          finally          {                  SplashScreen.CloseWaitForm();          }  } |

Результат работы вызова окна загрузки изображен на рисунке ниже.



## Controls

Каталог Controls предназначен для размещения пользовательских компонентов широкого применения. К примеру, в нем расположен компонент ErrorControl.

### ErrorControl

Пользовательский компонент ErrorControl служит для визуального представления исключений ([Exception](https://learn.microsoft.com/ru-ru/dotnet/api/system.exception?view=net-7.0)), которые возникают в клиентском приложении. Компонент отображает:

* Краткое описание типа исключения (сведения, предупреждение, ошибка), который представлен членами собственного перечисления MessageTypes;
* Текст исключения ([Exception.Message](https://learn.microsoft.com/ru-ru/dotnet/api/system.exception.message?view=net-7.0));
* Детали исключения. Представлены в виде полного текста ошибки ([Exception.ToString()](https://learn.microsoft.com/ru-ru/dotnet/api/system.exception.tostring?view=net-7.0)).

Также, в компоненте добавлена кнопка «Копировать», которая позволяет скопировать полный текст сообщения в буфер обмена.

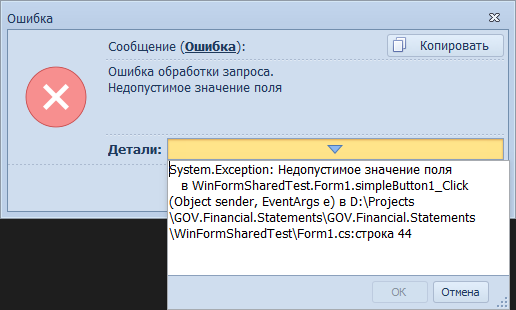
Возможны два варианта использования компонента:

1. Создание экземпляра компонента для добавления на форму в приложении, с последующим вызовом метода Setup компонента, при помощи которого устанавливаются параметры исключения для отображения.
2. Вызов метода Error, из класса [MessageBox](#_MessageBox), который отобразит диалоговое окно с деталями исключения.

Ниже приведен пример обработки исключения при помощи второго способа.

|  |
| --- |
| try {   /\* \*/   }  catch (Exception exception)  {          if (exception is WarningException warning)             MessageBox.Warning(warning.Message);          else MessageBox.Error(caption: $"Ошибка обработки запроса." +              $"{Environment.NewLine}{exception.Message}",              exception: exception);  } |

Результат обработки исключения при помощи компонента ErrorControl изображен на рисунке ниже.



Также можно добавить компонент ErrorControl из ToolBox на форму. Ниже приведен пример настройки компонента для обработки исключения.

|  |
| --- |
| try { /\* \*/ }  catch (Exception exception)  {          ErrorControlArgs args = new ErrorControlArgs          {                  MessageType = exception is WarningException                     ? MessageTypes.Warning                     : MessageTypes.Error,                  Message = exception.Message,                  ErrorDetails = exception.ToString()          };          errorControl.Setup(args);  } |

## Forms

Каталог предполагает размещение в нем оконных форм общего применения. В частности, форма c анимацией загрузки.

### LoadingForm

Данная форма является стандартной реализацией [окна загрузки](https://docs.devexpress.com/WindowsForms/10824/controls-and-libraries/forms-and-user-controls/splash-screen-manager/wait-form?v=19.2). На форме отображается заголовок и текст, а также анимация загрузки, которая вызывается в классе [SplashScreen](#_SplashScreen). К стандартной реализацию был добавлен метод ChangeLoadingPicture, который вызывается в конструкторе формы. Как можно догадаться из названия метода, он позволяет заменить стандартную анимацию загрузки от DevExpress, которая устанавливается в зависимости от стиля интерфейса (SkinName). Стандартный спиннер заменен на анимированный логотип Банка.