

Альтернативные титры XAML для VMIX

Общие положения

VMIX имеет возможность использовать в качестве титров XAML или DDL файлы реализующие пользовательский элемент User Control .NET Framework 3.5. Такое решение имеет ряд преимуществ и недостатков, соответственно по сравнению с функционалом титров, сделанных в GT Designer

Среди преимуществ можно выделить ряд основных:

- более гибкое использование слоев и взаимодействий элементов (автоматическое изменение размеров элементов в зависимости от наполнения)
- более широкий функционал разметки и дизайна, близкий по функционалу к HTML (табличные блоки и пр.)
- привязка (binding) значений, свойств текстовых полей¹ к свойствам других элементов титра. К примеру, привязав цвет текстового поля к свойству Background нескольких элементов Grid, можно одной командой API VMIX (SetTextColour) изменить цвет сразу нескольких элементов, или, скажем, можно привязать текстовое значение управляющего поля к ширине элемента или его положению на экране, что позволит интерактивно изменять внешний вид титра, изменяя значения текстового этого поля
- возможность анимировать практически любое свойство объектов.
- анимация через ключевые точки на таймлинии. Удобство создания анимаций
- возможность использовать Code Behind с альтернативной логикой и альтернативным управлением (*существенное* расширение функционала)

Однако, есть и ряд недостатков:

- Доступны всего две анимации (TransitionIn и TransitionOut)
- нет прямой поддержки png секвенций
- невозможно управлять анимациями через API (TitleBeginAnimation). Возможно только простейшие Play/Pause, SetPosition (перемещение по таймлинии)
- TransitionOut проигрывается **только** при OverlayOut

Для дизайна и разработки титра используется Microsoft Blend for Visual Studio²

¹ В контексте использования в VMIX интересны именно текстовые поля, т.к. у VMIX есть возможность управлять их содержимым, цветом и т.д. через API

² Документация по XAML и WPF есть на сайте Microsoft:

<https://learn.microsoft.com/ru-ru/visualstudio/xaml-tools/?view=vs-2022>

Среда разработки Blend for Microsoft Visual Studio

Установка

Для установки необходимо скачать установщик Visual Studio с официального сайта³. Далее выбрать для установки следующие пункты:

- Workloads => .Net desktop development (см. Рис. 1а)(можно снять все галки кроме Blend for Visual Studio, минимизировав таким образом установку) (см. Рис. 1б)
- Individual components => .NET Framework 3.5 development tools

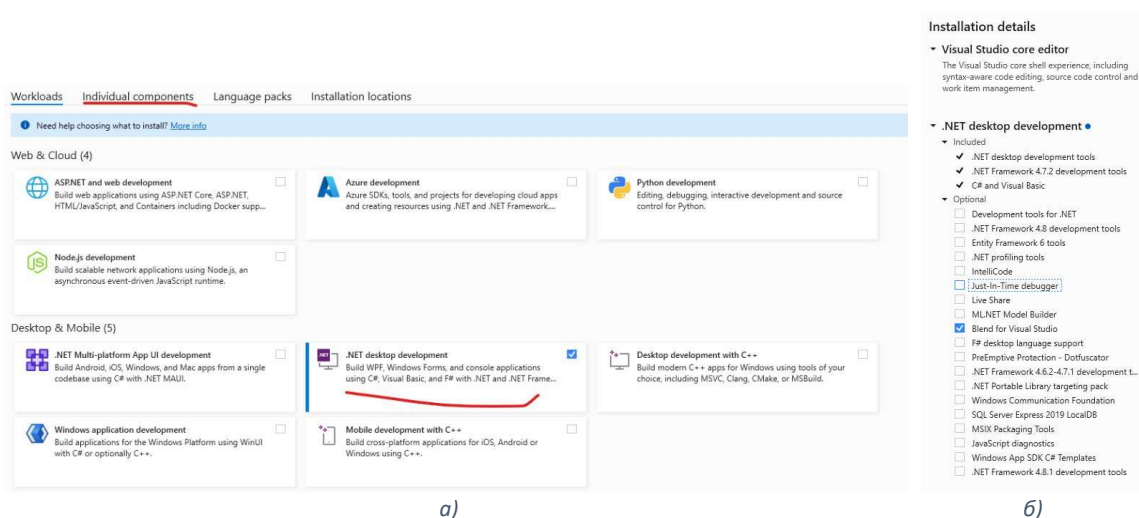


Рис. 1. Окно установщика Visual Studio

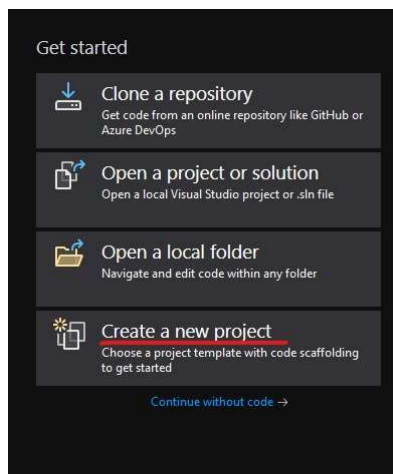
Далее запускаем процесс установки, ждем завершения. Можно приступить к созданию титров

Начало нового проекта в Blend

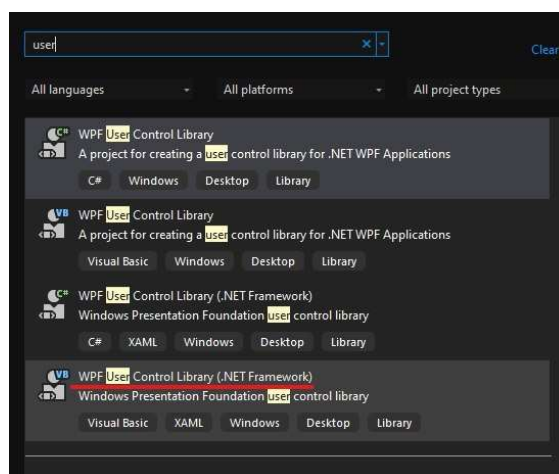
При открытии Blend будет открыто диалоговое окно (см. Рис. 2), в котором необходимо выбрать **Create a new project** (см. Рис. 2а)=>**WPF User Control (.NET Framework) VB** (см. Рис. 2б)=> **Название вашего проекта и целевой версии .NET Framework** (см. Рис. 2б) **необходимо выбрать 3.5**⁴=>**CREATE**

³ <https://visualstudio.microsoft.com/>

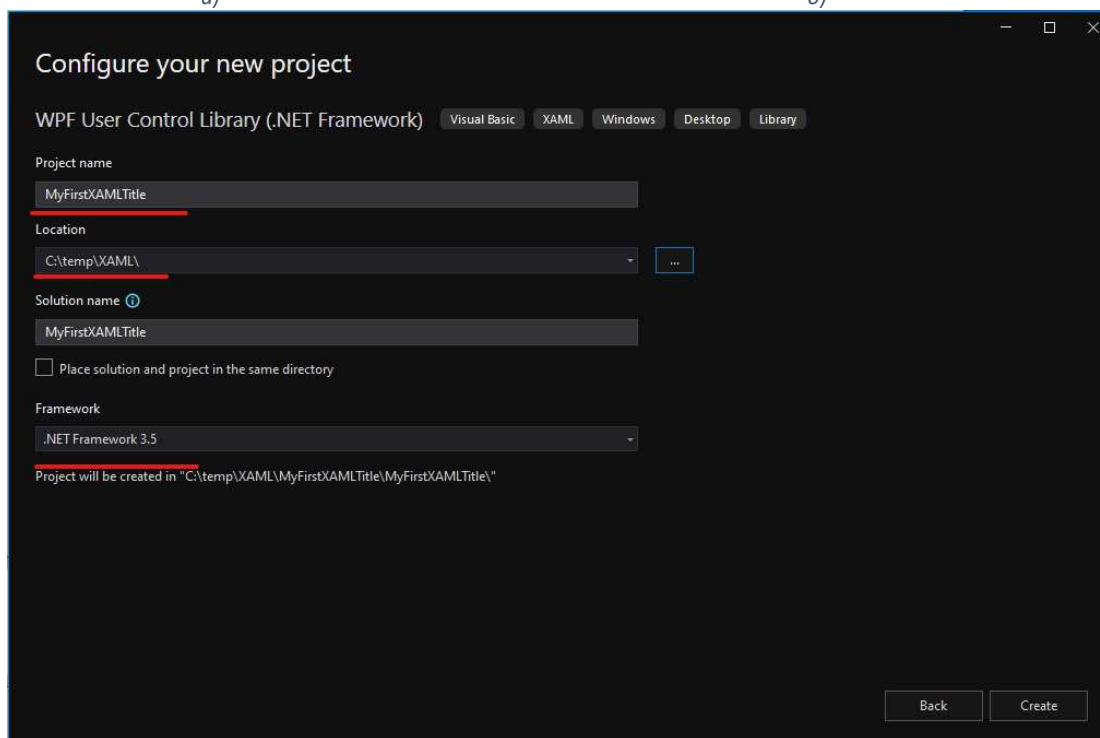
⁴ <https://www.vmix.com/help27/DeveloperInformation1.html>



а)



б)



в)

Рис. 2. Диалог выбора типа будущего проекта. а) Создание нового проекта, б) Выбор типа проекта, в) Назначение названия и целевой версии .NET Framework

После этого откроется основное окно Blend с вашим проектом (см. Рис. 3). По умолчанию ваш XAML будет называться UserControl1.xaml. Так же будет присутствовать файл UserControl1.xaml.vb (он не понадобится для задач проектирования титра)

Внешний вид окна Blend

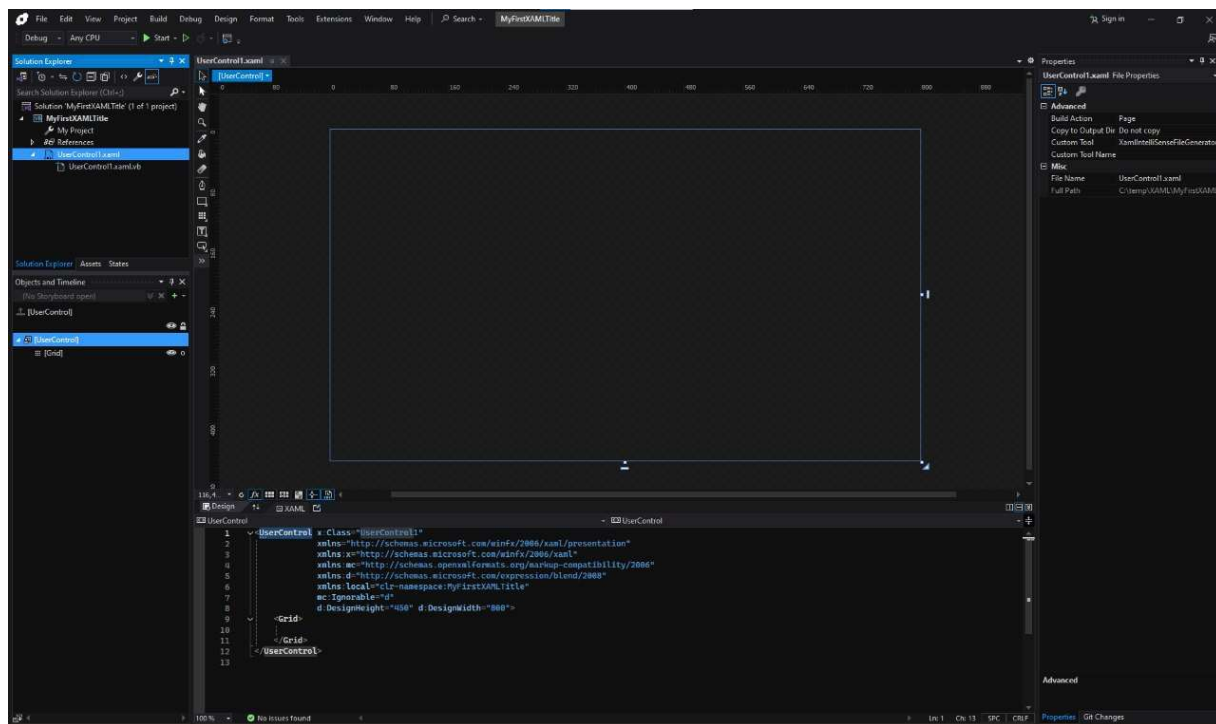


Рис. 3 Общий вид окна Blend при создании нового проекта

Окно содержит следующие области:

- Solution Explorer – навигатор по вашему проекту
- Objects and Timeline – окно управления деревом элементов и таймлиниями
- Design – окно, в котором отображается визуализация вашего титра и будет воспроизводиться анимация. Сбоку этого окна расположены иконки инструментов.
- Properties – основное окно, в котором будут производиться настройки параметров элементов и шагов анимации
- Code XAML- окно редактирования кода (чем меньше вносить изменений в этом окне – тем меньше ошибок будет появляться, при редактировании через элементы GUI Blend код меняется сам)

Основные графические элементы титров и их параметры

Среди основных инструментов (окно Design на Рис. 3) выделим основные:

- **Grid** – элемент сетка. Элемент контейнер (слой), т.е. может содержать в себе другие элементы. Расположение элементов внутри сетки (Margin) рассчитывается относительно границ сетки. Может иметь вид таблицы⁵
- **TextBlock** – основной текстовый элемент. Именно он будет отображаться в окне TitleEditor в VMIX и его содержимое доступно для управления через API VMIX, поэтому их различные поля (поля **Text** и **Foreground**⁶) следует использовать для привязки свойств других элементов к значениям этих полей
- **Rectangle** – элемент-фигура прямоугольник. Может быть использован для подложки текста. В отличие от Grid имеет параметры Stroke и Radius (скругление краёв). Схожие элементы **Ellipse** и **Line**
- **Image** - элемент для интеграции изображений в титр. Он так же, как и TextBlock доступен для редактирования через API VMIX⁷

Все элементы имеют ряд общих настроек. Среди основных, с которыми будет происходить взаимодействие можно выделить следующие:

- **Layout** (см. Рис. 4):
 - **Width/Height** – ширина/высота элемента может быть как в абсолютных единицах, так и в режиме **Auto** (настраивается в зависимости от размеров содержимого). Режим **Auto** часто применяется в титрах с динамическими полями, в которых размер зависит от текстового наполнения.
 - **Row/Column** – указывает в какую ячейку таблицы сетки поместить элемент (в случае если Grid в режиме таблицы)
 - **Horizontal/Vertical Alignment** – выравнивание элемента и принцип отсчета **Margin**
 - **Margin** – координаты относительно границ родительского элемента
 - **Min/Max Height/Width** – минимальные/максимальные значения размеров элемента. Ограничивает вариации размера элемента.
- **Appearance** (см. Рис. 5):
 - **Opacity** – прозрачность
 - **Visibility** – видимость элемента. Visible/Hidden/Collapsed. Collapsed – режим, когда элемент не просто скрыт, а скрыт и не занимает места в Layout (имеет нулевые размеры)
 - **Effect** – доступно два вида эффектов: Blur, DropShadow
 - **ClipToBounds** – ограничивает видимость вложенных элементов границами родительского

⁵ Параметры ColumnDefinitions и RowDefinitions позволяют создавать составные элементы, скажем, когда титр состоит из двух функционально различных частей с динамически изменяемыми границами, расположенные рядом друг с другом, или, к примеру, для реализации эффекта отражения.

⁶ Поле **Text** – собственно текстовое наполнение элемента. **Foreground** – цвет текста, доступно для изменения через команду «**SetTextColour**» из API VMIX

⁷ Поле **Source** элемента **Image** доступно для изменения через вкладку **Images** в окне **Title Editor**, и также через команду «**SetImage**» из API VMIX

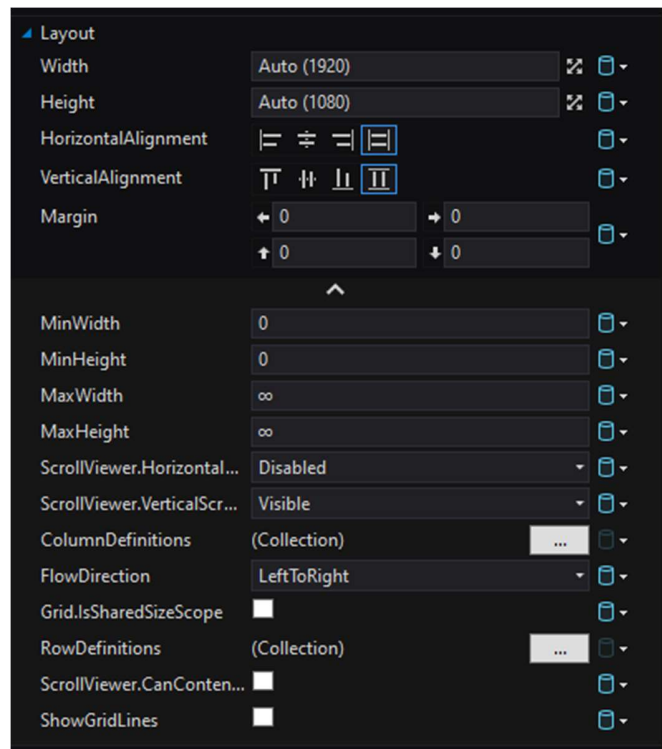


Рис. 4. Панель свойств Layout (расширенные настройки)

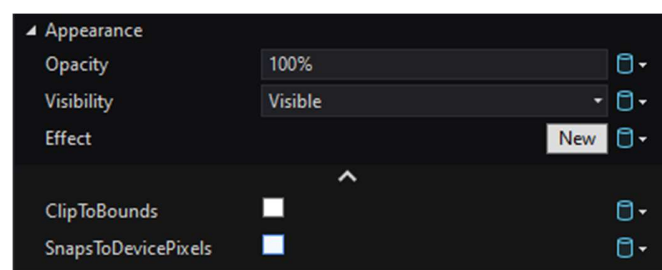


Рис. 5. Панель Appearance (расширенные настройки)

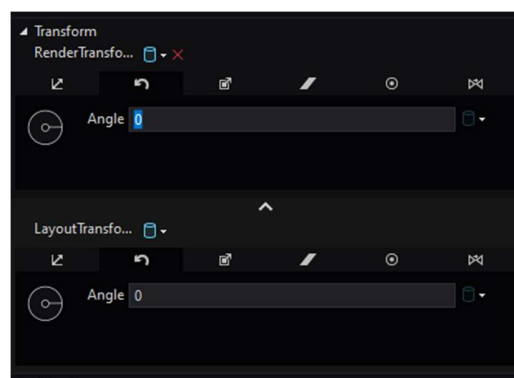


Рис. 6. Панель Transform

- **Transform** (см. Рис. 6). Применяет относительную трансформацию для элемента, не меняя его параметров ширины, длины и т.д. Среди параметров трансформации:
 - *Translate* – перемещение
 - *Rotate* – поворот
 - *Scale* – изменение масштаба (отрицательные значения отображают относительно заданной оси)
 - *Skew* – наклон
 - *CenterPoint* – точка, относительно которой будет происходить трансформация
 - *Flip* – отражение относительно выбранной оси. По факту добавляет знак «-» к соответствующему параметру *Scale*

Существует два вида трансформации, которые похожи по функционалу, но имеют некоторые различия:

- **RenderTransform** применяется после компоновки элемента (см. Рис. 7а, Рис. 8а)
- **LayoutTransform** применяется до компоновки элемента (см. Рис. 7б, Рис. 8б)

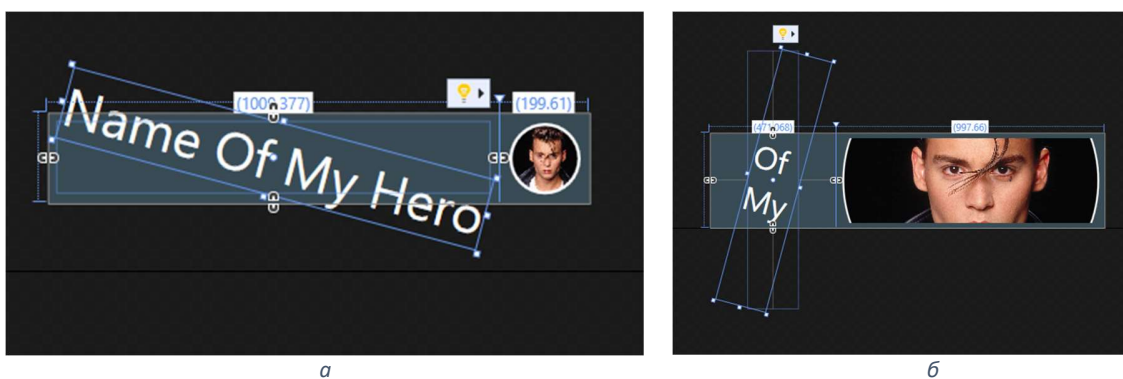


Рис. 7. Разница между Rotate Transform (15 градусов) для а) RenderTransform; б) LayoutTransform



Рис. 8 Разница между Scale (X: 0,5) Transform для а) RenderTransform; б) LayoutTransform

На этом знакомство с основными инструментами завершено. Перейдем к созданию нашего первого титра.

Создание титра. Анимация параметров.

Техническое задание

Допустим, необходимо создать титр, изображенный на Рис. 9

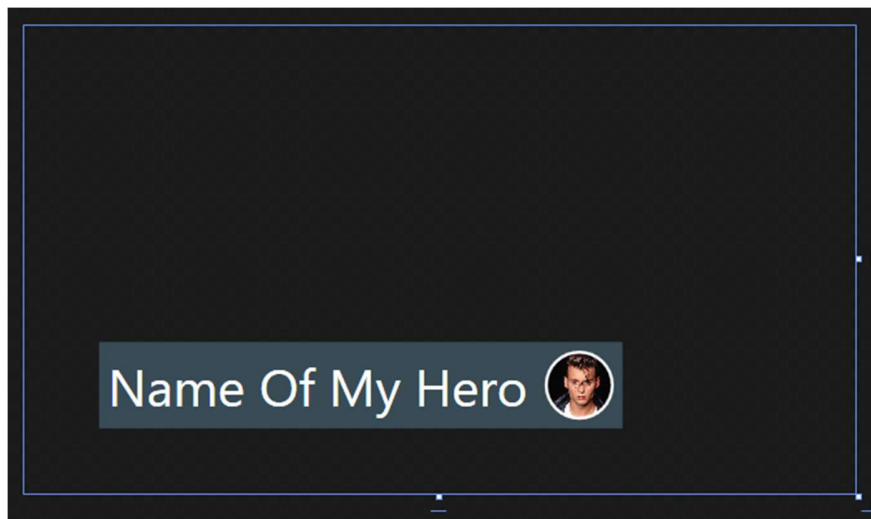


Рис. 9. Общий вид конечного дизайна титра

Требования к титру:

- Титр состоит из двух областей: область текста и область аватара.
- Аватар располагается справа от текста, имеет круглую форму, с окантовкой заданной ширины.
- Масштаб изображения в нем должен автоматически подстраиваться.
- Аватар необходимо иметь возможность динамически менять.
- Длина текстового поля – динамическая, положение аватара привязано к длине текстового поля.
- Нужно предусмотреть управление цветом подложки для титра.
- Появление/исчезание титра мягкое (нелинейное) из-под нижней границы кадра

Переименование UserControl1.xaml.

Для того, чтобы переименовать наш XAML, необходимо правой кнопкой мыши вызвать контекстное меню (см. Рис. 10) и выбрать Rename. После переименования сохранить и переоткрыть проект, чтобы панели Design обновилось.

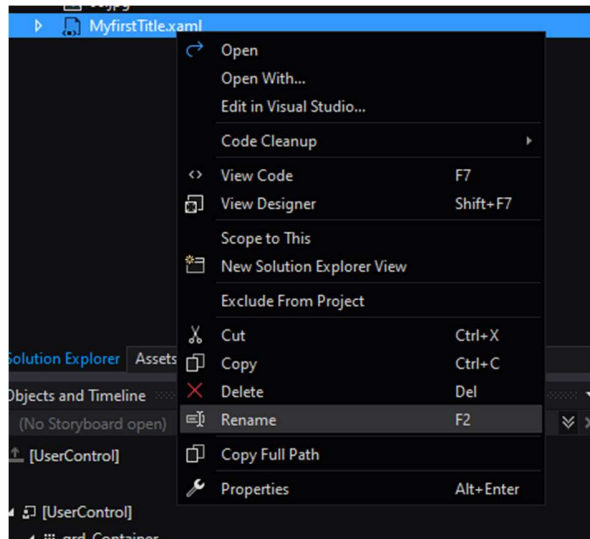


Рис. 10. Контекстное меню основного XAML файла

Внесение изменений в шапку XAML

Для корректной работы XAML с VMIX необходимо (см. Рис. 11) в шапке XAML файла удалить стройки `d:DesignHeight="450" d:DesignWidth="800"`. А также заменить строку

«`xmlns:local="clr-namespace:....."`» на

`xmlns:mtl="clr-namespace:vMixTitleLibrary;assembly=vMixTitleLibrary"`

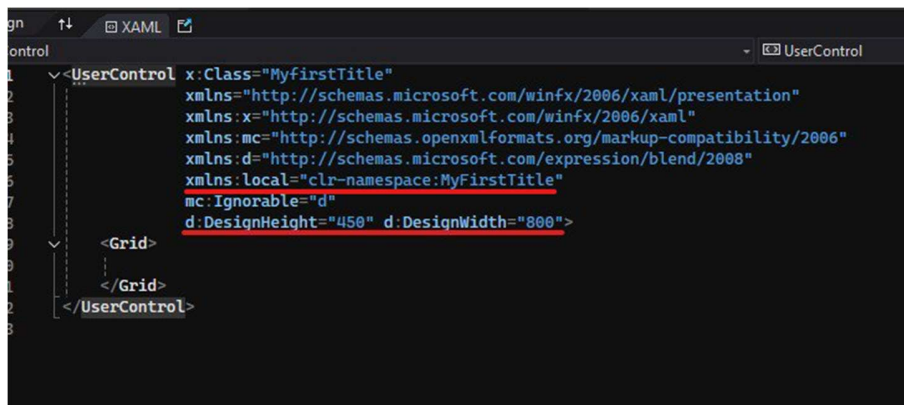


Рис. 11. Внесение изменений в шапку XAML Файла

Создание новых элементов

При создании новых элементов всегда присваивайте им уникальные имена. Для удобства используйте шаблон *type_Name*, например *grd_Main*, *txt_Title*, *img_MyImage* и т.д. Так будет легче ориентироваться в проекте. И привязки (*bindings*) оперируют именно именами элементов.

Задание размеров титра

Выберите главный элемент **UserControl** (см. Рис. 12). Зададите для него в панели **Layout** (см. Рис. 4above) Основные графические элементы титров и их параметры ширину и высоту равную 1920x1080

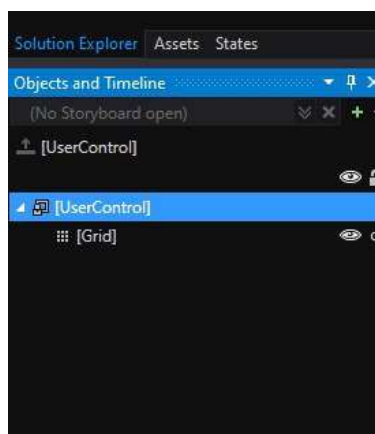


Рис. 12

Добавление элементов

Переименуйте родительский **Grid** в **grd_Container**. С помощью инструментов на панели **Design** (см. Рис. 3). Создайте несколько новых элементов:

- **Grid** и сразу же дадите название **grd_TextContainer**. Значения ширины и высоты назначьте **Auto**. **Background** выберите для наглядности красным
- **TextBlock** с именем **txt_MyText**. Значения ширины и высоты назначьте **Auto**. **Foreground** – белый, **Background** – синий (впоследствии можно отменить, он используется на данном этапе для понимания границ элементов и отступов)

Теперь необходимо поместить **txt_MyText** внутрь **grd_TextContainer**. Для этого необходимо перетащить его, как показано на Рис. 13

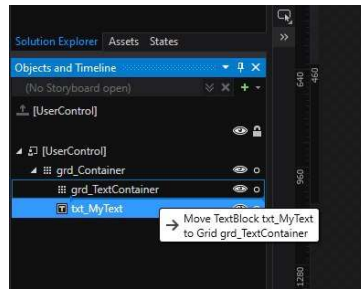


Рис. 13. Перемещение текстового элемента внутрь контейнера Grid

Зададите отступы для **txt_MyText** через Margin на панели **Layout** (см. Рис. 14)

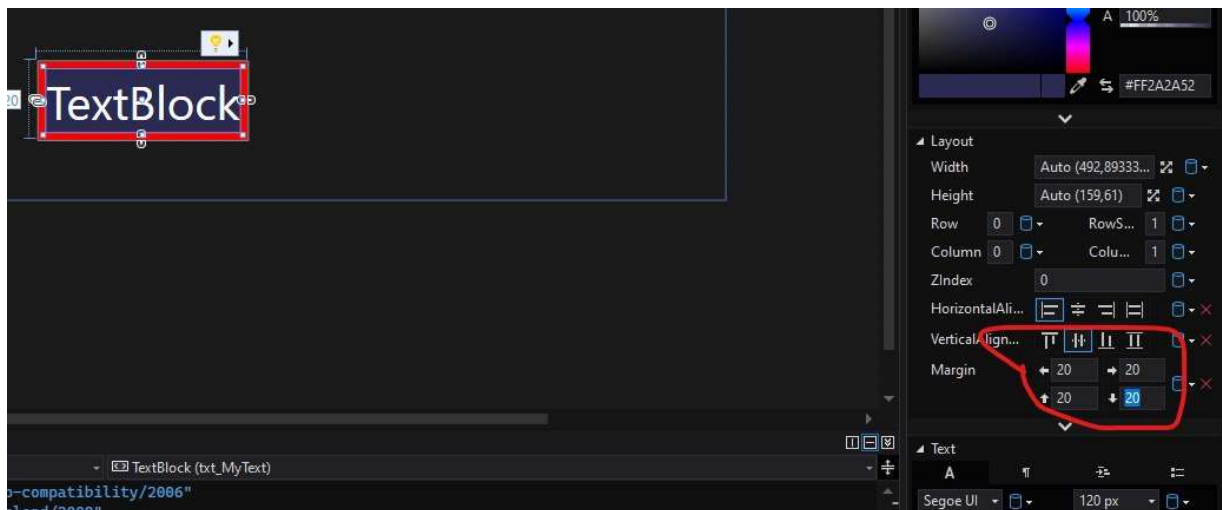


Рис. 14. Задание отступов для текстового элемента внутри родительского контейнера

Проверьте, что логика отступов и изменения размеров нашего титра работают правильно. Если всё устраивает – отключите Background у **txt_MyText**

Создание привязки цвета Background и параметра Foreground управляющего текстового элемента

Добавьте в наш корневой **grd_Container** управляющий **TextBlock** с именем **txt_BGColor**. Задайте для него **Appearance=>Visibility=>Collapsed** (см. Рис. 5), чтобы он не участвовал в формировании нашего дизайна, а был лишь скрытым элементом. Далее выберите **grd_TextContainer** и в панели **Brush=>Background** (см. Рис. 15) вызовите окно создания привязки, и в нем выберите справа элемент к которому надо привязаться, слева выберите свойство, которое необходимо привязать от этого элемента (в нашем случае **Foreground (SolidColorBrush)**)

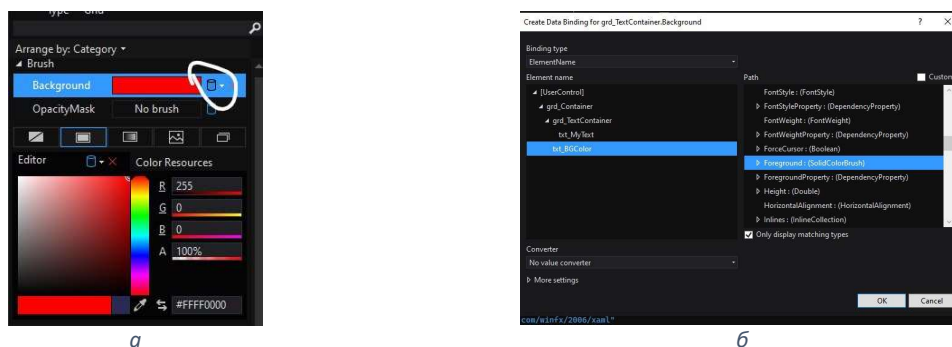


Рис. 15. Создание привязки свойства *Background* и *Foreground* для элемента.
а) вызов меню назначения привязки; б) окно назначения привязки

Нажмите «ОК». Привязка создана. Теперь при изменении цвета текста (***Foreground***) в элементе ***txt_BGColor*** будет синхронно меняться цвет подложки (***Background***) у ***grd_TextContainer***

Добавление аватара

Для того, чтобы координаты аватара были всегда «привязаны» в правому краю текстового блока, необходимо добавить свойство таблицы для ***grd_TextContainer***. Тогда текстовый блок разместится в первой виртуальной «колонке» этой таблицы, которая будет менять свою ширину в зависимости от содержимого текстового элемента, а вторая «колонка», в которой будет размещен аватар, будет всегда «рядом» с первой.

Для придания нашему контейнеру ***grd_TextContainer*** свойств таблицы добавьте ***ColumnDefintitions***, нажав на кнопку напротив соответствующего поля в панели **Layout** (см. Рис. 4) для ***grd_TextContainer***. В появившемся окне редактирования ***ColumnDefinitions*** (см. Рис. 16) добавьте два **ColumnDefinition** с настройкой **Layout=>Width=>Auto**

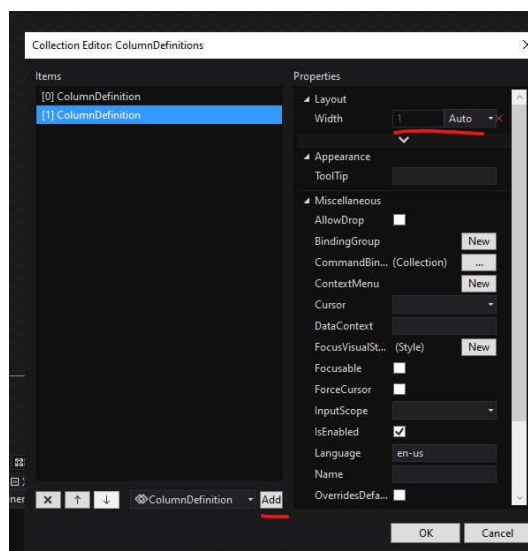


Рис. 16. Окно редактирования *ColumnDefinitions*

Добавьте элемент **Image** с именем ***img_MyImage*** в корневой контейнер ***grd_Container***. Этот элемент будет доступен в VMIX для изменения источника изображений. Также, как и для

txt_BGColor сделайте его **Collapsed**. Он понадобится нам лишь для привязки. Выберите какое-нибудь изображение в панели **Common**=>**Source**

Добавьте элемент **Ellipse** с именем **el_Avatar** в **grd_TextContainer**. Создайте привязку **Width** и **Height** с **ActualHeight txt_MyText**. Оба параметра надо привязать именно к параметру высоты, чтобы в итоге получить круг. Зададим цвет **Brush**=>**Stroke** белым (или сделайте привязку к **Foreground txt_MyText**, чтобы цвет обрисовка аватара совпадал с цветом шрифта), а также **Appearance**=>**StrokeThickness**=>7 (или любой другой по вашему желанию). И **Margin** такой-же как и у **txt_MyText**, либо сделайте привязку к этому параметру.

Для **txt_MyText** **Layout**=>**Column**=> задать значение 0 (см. Рис. 4)

Для **el_Avatar** **Layout**=>**Column**=> задать значение 1 (см. Рис. 4)

Далее необходимо на панели **Brush** для **el_Avatar** выбрать **TileBrush** (см. **Error! Reference source not found.а**). И для поля **ImageSource** (см. **Error! Reference source not found.б**). Для этого необходимо включить режим **Custom** и прописать «Source» в поле **Path**⁸. Режим **Stretch** выбрать **UniformToFill**. Это будет автоматически изменять размер изображения под размер элемента **Ellipse**

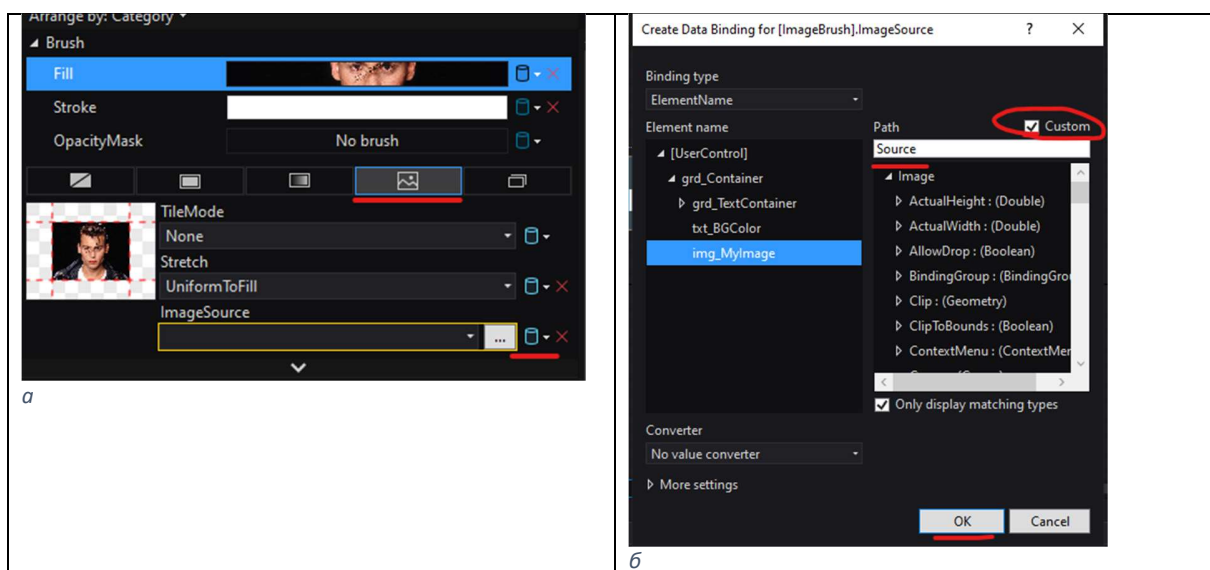


Рис. 17. Настройка Brush для Ellipse.

а) Панель Brush для Ellipse; б) Создание привязки ImageSource к полю Source элемента Image

Если все было сделано правильно, дизайн титра можно считать законченным. Можно приступать созданию анимаций

⁸ Привязка параметров к полю **Text** происходит аналогично через режим **Custom** и набор «Text» в поле **Path**

Создание анимации

Для добавления новой Таймлинии, нажмите клавишу добавления новой таймлинии (см. Рис. 18а). Далее необходимо выбрать название для вашей будущей таймлинии⁹. Откроется окно (см. Рис. 18б), но без ключевых точек. Для добавления ключевой точки достаточно переместить полосу прокрутки в нужное место на таймлинии и произвести изменения необходимого для анимации параметра, ключевая точка появится автоматически.

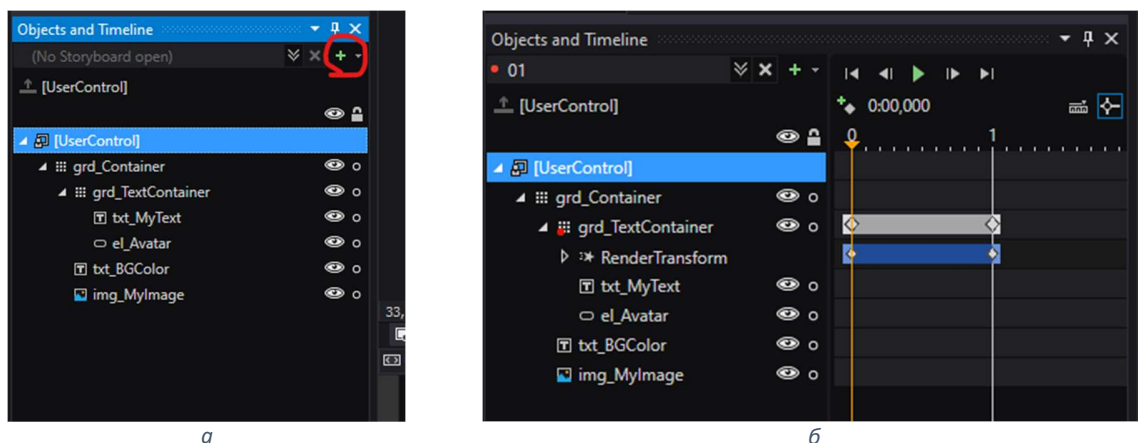


Рис. 18. Создание анимации.

а) Добавление новой Таймлинии;

б) Панель редактирования таймлайнов с добавленными ключевыми точками

Для анимации проявления («01») первой ключевой точкой добавим
Transform=>RenderTransform=>TranslateY=>500

Второй: Transform=>RenderTransform=>TranslateY=>0

Для анимации ухода («02») аналогично, только в обратном порядке

Для изменения параметра сглаживания (easing) кривой изменения для ключевой точки, необходимо выбрать ключевую точку (не начальную) и тогда на панели Properties появится панель Easing. Для «смягчения» анимации и придания ей нелинейного характера отредактируйте поля x1,y1,x2,y2, как показано на Рис. 19

⁹ По умолчанию VMIX подгружает две первые таймлинии в качестве основных, остальные будут недоступны. Первая из них будет проигрываться при запуске титра через Play и OverlayIn, вторая таймлиния будет проигрываться **ТОЛЬКО** при OverlayOut. Т.к. процедура подружки этих таймлайнов не всегда работает согласно алфавитному порядку и иногда меняются местами TransitionIn и TransitionOut (берутся первые два элемента из массива Resources.Keys и итерируются через For Each). Лучше называть TransitionIn анимацию «01», TransitionOut «02».

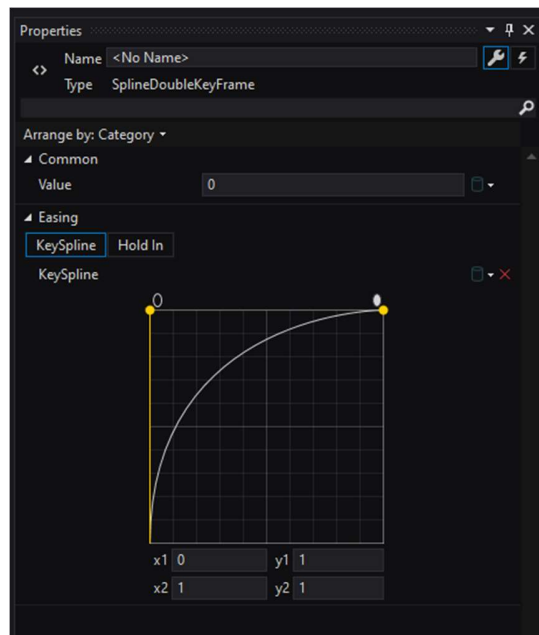


Рис. 19. Панель свойств ключевой точки

После того, как созданы анимации для появления и исчезания титра, сохраните проект. Теперь можете добавить титр в VMIX. Как видно на Рис. 20, все поля доступны для редактирования

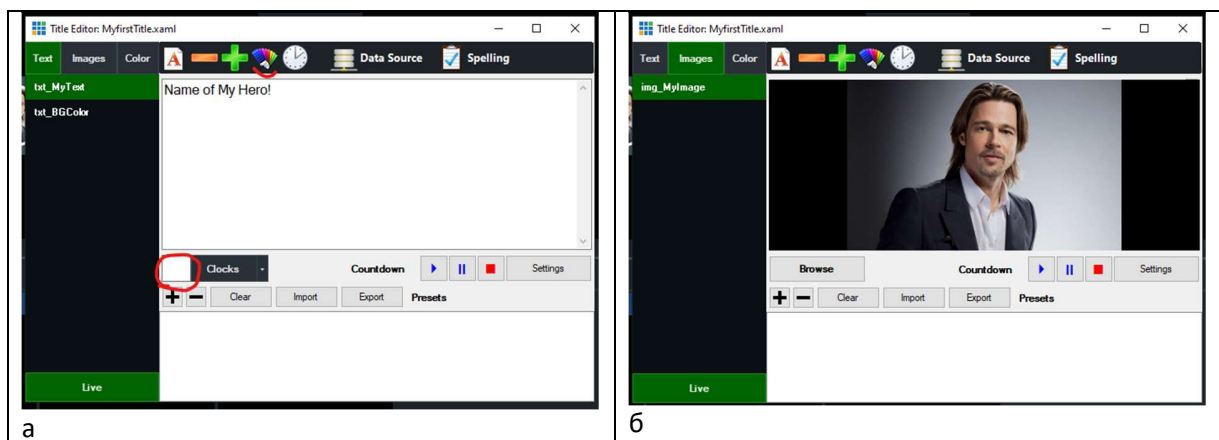


Рис. 20. Окно редактирования титра VMIX.

- а) Редактирование текстовых полей. Красным выделены элементы управления цветом текста
б) Редактирования источников изображений

Демо проект в Blend находится в папке **./BlendProj**

Для более продуктивного усвоения информации рекомендуется произвести все вышеописанные действия самостоятельно

Плагин для управления анимациями (в режиме бета)¹⁰

Т.к. изначально в VMIX при загрузке титра XAML доступны только две анимации без возможности выбора типа анимации при воспроизведении, в отличие от титров GT Title, возникла необходимость разработать решение, которое позволило бы управлять текущей активной таймлинией в титре. XAML титры можно скомпилировать в .dll файл с внутренней логикой. Поэтому был разработан плагин, позволяющий динамически загружать любой работающий XAML титр и через специальное текстовое поле, в котором пользователь указывает название таймлинии, воспроизводить через Play именно ту таймлинию, которую пользователь укажет в специальном текстовом поле, которое динамически добавляется к существующим полям в титре.

Плагин находится в **./Plugin/XAMLplugin.dll**¹¹

Для использования плагина его необходимо скопировать в место, где располагается ваш XAML титр, переименовать его так же, как назван ваш XAML титр (см. Рис. 21). И в VMIX уже добавлять файл dll

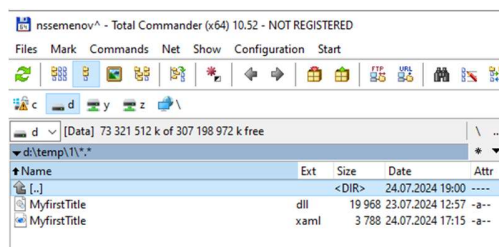


Рис. 21

После добавления dll в VMIX. У титра появится поле **txt_Transition** (см. Рис. 22). По умолчанию, если это поле пустое – играет первая таймлиния в логике VMIX. В это поле необходимо написать название вашей таймлинии, тогда при команде Play будет воспроизводиться таймлиния с названием, указанным в **txt_Transition**. В случае, если указанное название не совпадает ни с одной таймлинией – играет первая таймлиния в логике VMIX

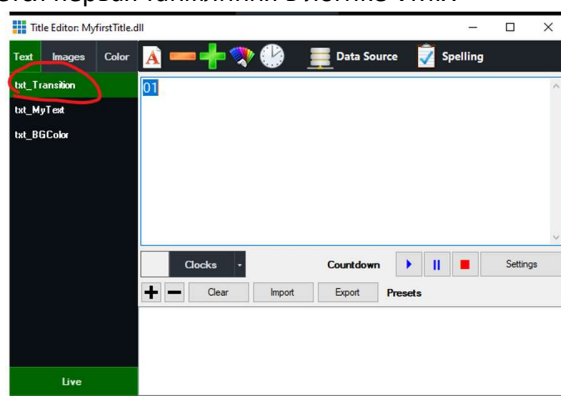


Рис. 22. Окно редактирования титра dll

¹⁰ Исходный код: https://github.com/semenovnick/vmix_XAML_Control_Plugin

¹¹ Плагин не будет работать из сетевых источников. Необходимо скопировать на локальный диск (это связано с настройками безопасности NET Framework)