

Работа с утилитой для создания шаблонов печати на смарт-картах

Rutoken KeyBox Card Template Designer — утилита, предназначенная для создания шаблонов печати на смарт-картах, используемых на принтере EDIsecure XID 8300.

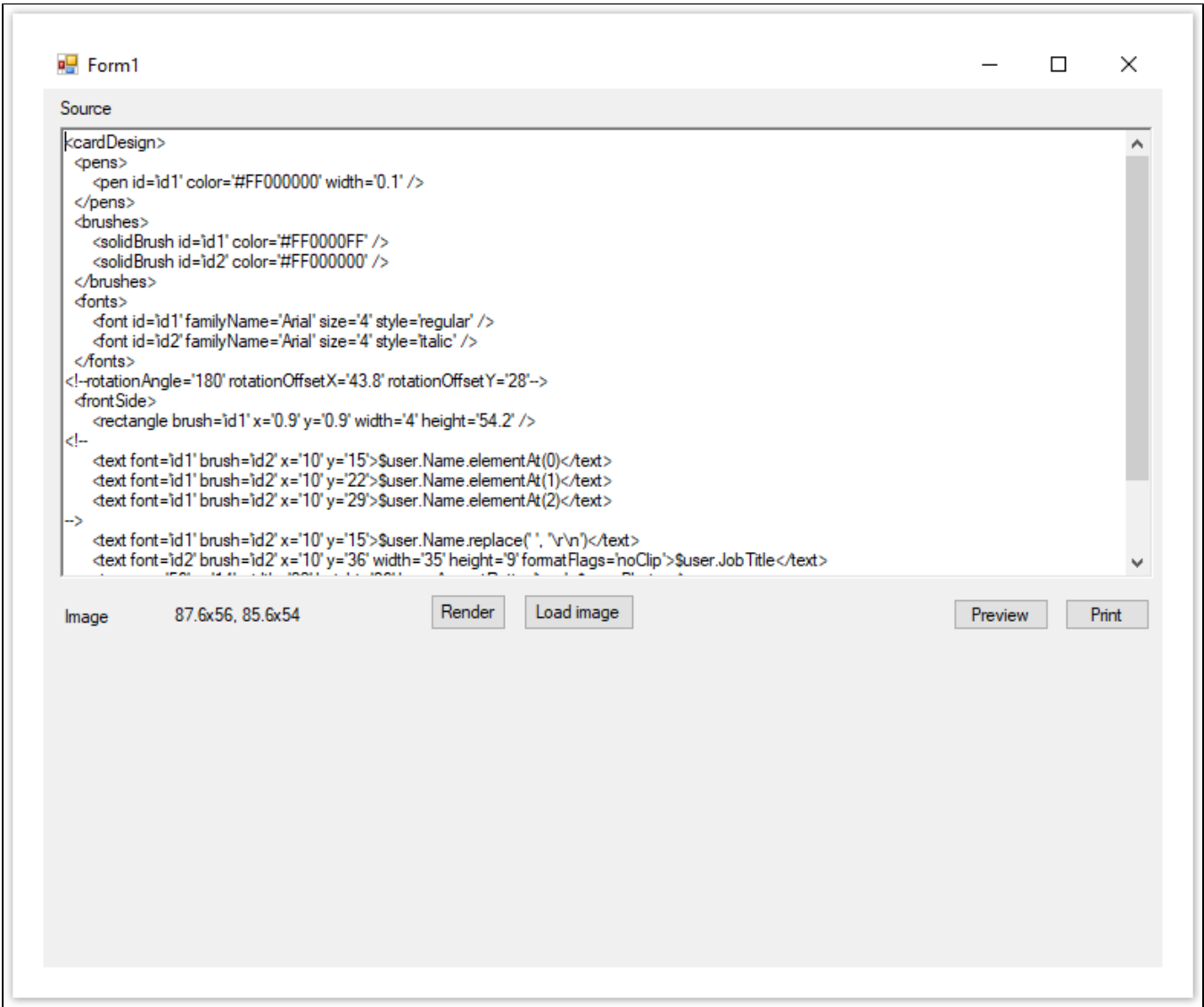
Интеграция РутOKEN KeyBox с принтером позволяет:

- выпускать смарт-карты используя считыватели принтера (контактный и бесконтактный) без печати;
- выпускать смарт-карты пользователям используя считыватели принтера (контактный и бесконтактный) с печатью на карте изображения и текста;
- печатать на смарт-картах изображение или текст без выпуска карты пользователям.

Card Template Designer предоставляется по запросу службой технической поддержки.

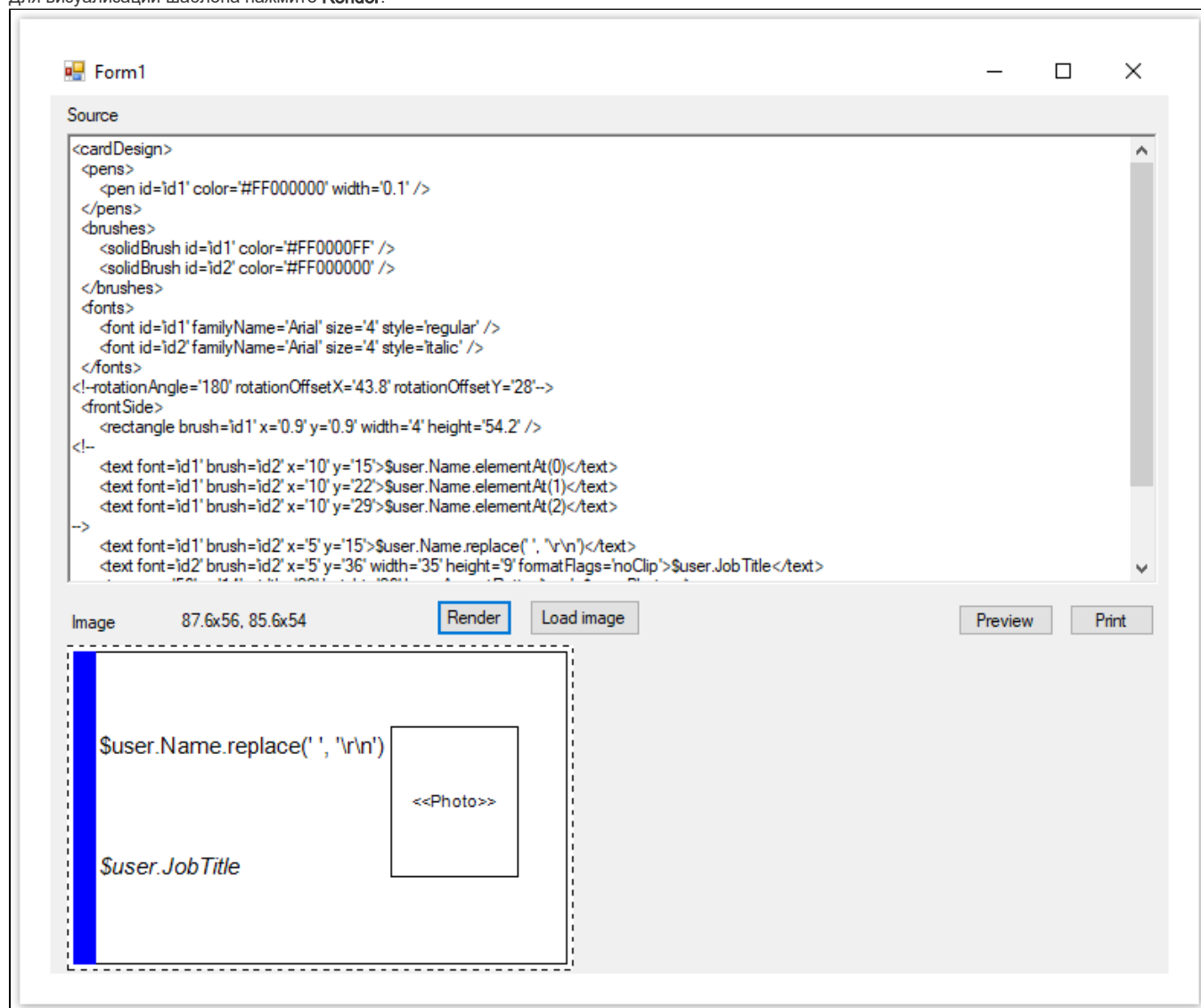
Для начала работы с утилитой запустите из дистрибутива приложение cardTemplateDesigner.exe.

После запуска в окне приложения будет загружен базовый шаблон печати.



Чтобы создать шаблон печати, в поле **Source** внесите необходимые изменения.

Для визуализации шаблона нажмите **Render**.



При создании шаблонов печати используйте [Windows GDI + API](#) в синтаксисе XML.

Всё содержимое шаблона должно находиться в секции `<cardDesign>...</cardDesign>`.

Текст, расположенный между секциями `<!-- ... -->`, считается закомментированным и не учитывается при создании шаблона.

При создании шаблона используются следующие элементы:

- имя шаблона;
- перо;
- кисть;
- шрифт;
- текст;
- перенос строк;
- изображение;
- прямоугольник;
- сторона карты;
- вращение.

Отображаемое имя шаблона

Имя шаблона отображается в интерфейсе Рутокен KeyBox.

Задается в значении параметра **displayName** в секции `<cardDesign>`.

Пример

```
<cardDesign displayName=" " >
...
</cardDesign>
```

Перо

Задаётся в секции **<pens>** с параметрами:

- pen id;
- color;
- width.

Пример

```
<pens>
<pen id='id1' color='#FF000000' width='0.1' />
</pens>
```

Описанная строка определяет цвет линии и толщину. Далее pen id задаётся как параметр в других секциях. Например, как контур прямоугольника:

```
<rectangle pen='id1' x='0.9' y='0.9' width='4' height='54.2' />
```

Для задания цвета используется шестнадцатеричное значение и степень прозрачности.

Например, #FFA52A2A, где FF — непрозрачность коричневого цвета A52A2A.

Таблица цветов и прозрачности приведена в [приложении](#).

Кисть

Задаётся в секции **<brushes>** с параметрами:

- brush id;
- color.

Пример

```
<brushes>
<brush id='id1' color='#FF0000FF' />
</brushes>
```

В дальнейшем **brush id** задаётся как параметр в других секциях. Например, как цвет прямоугольника:

```
<rectangle brush='id1' x='0.9' y='0.9' width='4' height='54.2' />
```

Шрифт

Задаётся в секции **<fonts>** с параметрами:

- font id;
- familyName;
- size;
- style: regular, bold, italic, underline, strikethrough.

Пример

```
<fonts>
<font id='id1' familyName='Arial' size='3' style='regular' />
<font id='id2' familyName='Arial' size='4' style='italic' />
</fonts>
```

Описанная структура определяет id, название, размер и начертание шрифтов. В дальнейшем **font id** задаётся как параметр в других секциях. Например, в секции **<text>**:

```
<text font='id1' brush='id2' x='10' y='15'>${user.Name.elementAt(0)}</text>
<text font='id2' brush='id2' x='10' y='36'>${user.JobTitle}</text>
```

Текст

Задается в секции `<text>` с параметрами:

- font;
- brush;
- x;
- y;
- width (опциональный);
- height (опциональный);
- horizontalAlignment: left, right, center (опциональный);
- verticalAlignment: top, bottom, center (опциональный);
- formatFlags (см. [подробное описание значений](#)):
 - directionRightToLeft;
 - directionVertical;
 - displayFormatControl;
 - fitBlackBox;
 - lineLimit;
 - measureTrailingSpaces;
 - noClip;
 - noFontFallback;
 - noWrap.

Пример

```
<text font='id2' brush='id2' x='10' y='36' width='35' height='9'
formatFlags='noClip'>$user.JobTitle</text>
```

Описанная строка определяет шрифт, цвет, положение, размер и формат вывода должности пользователя, подставляемой из Active Directory.

Атрибуты пользователя Active Directory, выводимые на печать

Данные пользователя (ФИО, должность, подразделение и т.д.) подставляются в шаблон из атрибутов Active Directory.

Синтаксис для подстановки значений часто используемых атрибутов:

- \$user.Name;
- \$user.LogonName;
- \$user.PrincipalName;
- \$user.FirstName;
- \$user.LastName;
- \$user.Email;
- \$user.TelephoneNumber;
- \$user.Country;
- \$user.State;
- \$user.Locality;
- \$user.Organization;
- \$user.OrgUnit;
- \$user.Street;
- \$user.JobTitle.

Для печати данных из произвольного атрибута используйте синтаксис: `$user.attribute('имя атрибута Active Directory')`.

Пример

```
$user.attribute('Notes')
```

Перенос строк

Для переноса текста по строкам (например, разбивки ФИО пользователя, подставленного из Active Directory на несколько строк) можно использовать параметры **replace** (замена символа) или **elementAt** (извлечение элемента из строки).

Пример 1

Фамилия, имя и отчество пользователя подставляются из атрибута Active Directory и пишутся в одну строку в шаблоне через пробелы:

```
<text font='id1' brush='id2' x='10' y='15'>$user.Name</text>
```

Пример 2

Фамилия, имя и отчество пользователя подставляются из атрибутов Active Directory, пробелы заменяются на перевод строки (`\r` — переход к началу строки, `\n` — переход на новую строку).

В результате в шаблоне ФИО разбивается на три строки:

```
<text font='id1' brush='id2' x='10' y='15'>$user.Name.replace("\r\n")</text>
```

Аналогичным образом может быть перенесено на несколько строк значение должности или подразделения пользователя. Вместо пробела может быть любой другой символ.

Для переноса извлеченного из Active Directory значения ФИО по строкам, без замены текста применяется `elementAt`.

ФИО пользователя можно вывести в шаблоне построчно следующим образом:

```
<text font='id1' brush='id2' x='10' y='15'>$user.Name.elementAt(0)</text>
<text font='id1' brush='id2' x='10' y='22'>$user.Name.elementAt(1)</text>
<text font='id1' brush='id2' x='10' y='29'>$user.Name.elementAt(2)</text>
```

где:

- 0 — первое слово (например, Имя);
- 1 — второе слово (например, Фамилия);
- 2 — третье слово (например, Отчество).

Изображение

Задается в секции `<image>` с параметрами:

- `x`;
- `y`;
- `width` (опциональный);
- `height` (опциональный);
- `keepAspectRatio` (true или false);
- `data` (опциональный).

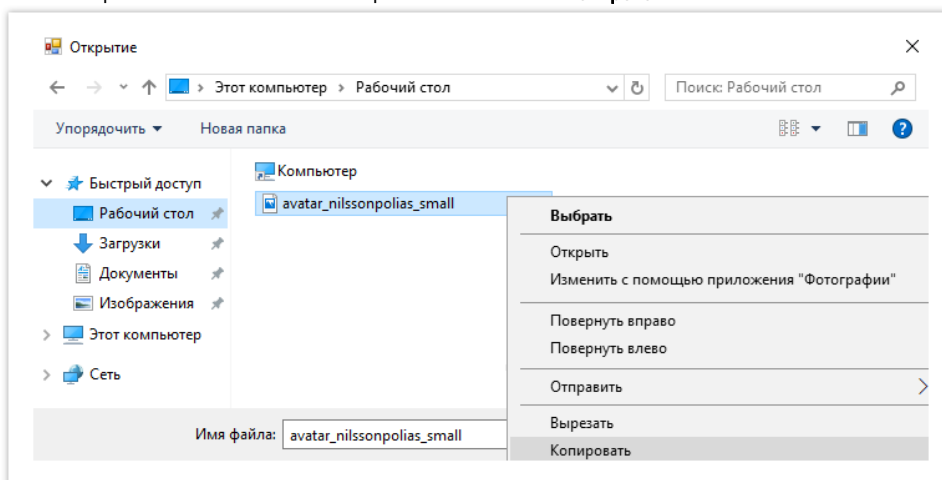
Пример

```
<image x='56' y='14' width='22' height='26'
keepAspectRatio='true'>$user.Photo</image>
```

Описанная строка определяет положение, размер и следование пропорциям изображения, подставляемого из атрибута Active Directory (переменная `$user.Photo`).

Помимо переменной можно подставить и изображение в кодировке base64 (например, для добавления в шаблон логотипа компании) и затем визуализировать его выполнив следующие действия:

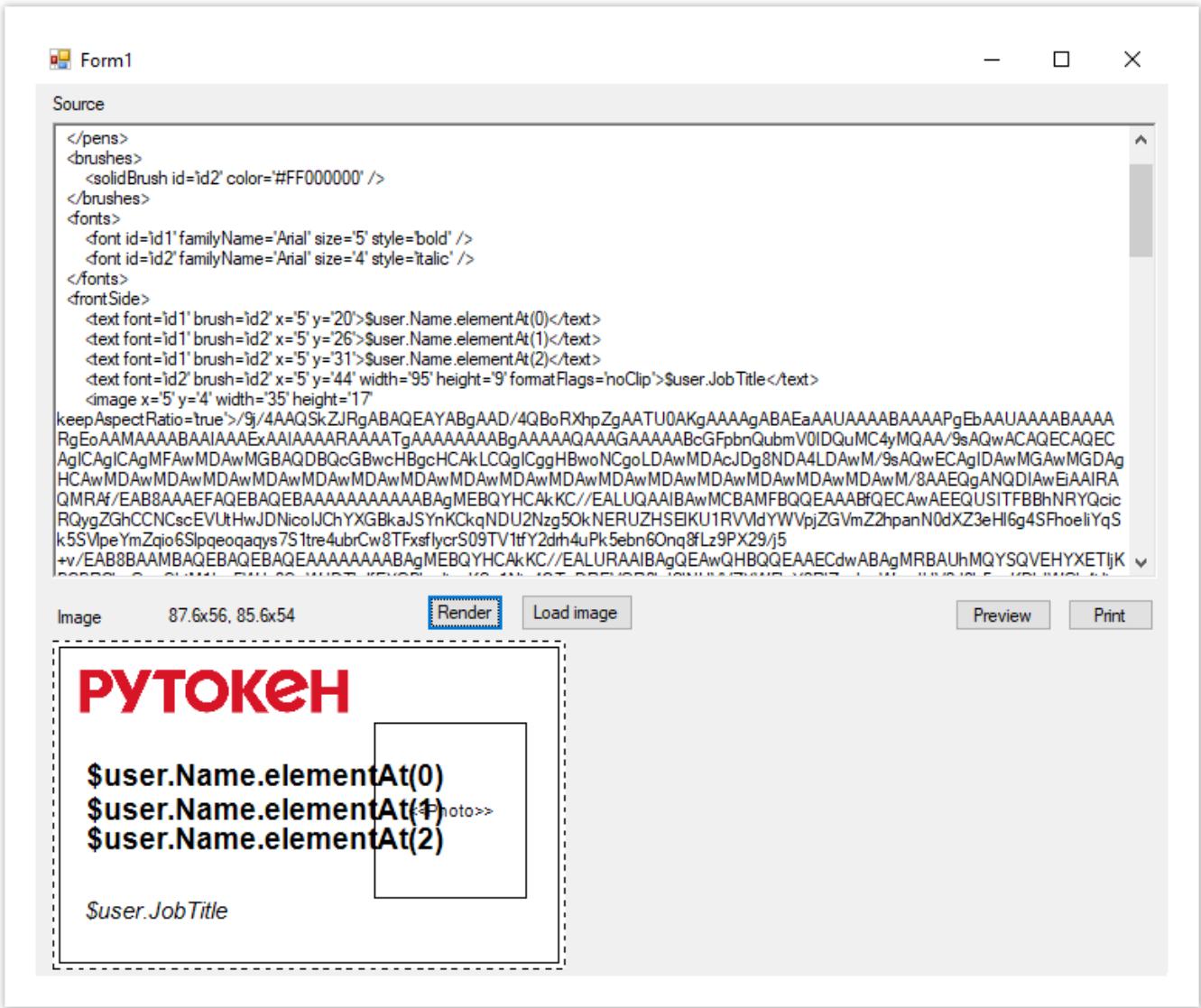
1. Нажмите **Load image** и перейдите в окно выбора изображения в формате JPEG. Размер изображения подбирается исходя из желаемого дизайна шаблона и размеров смарт-карты. Стандартный размер изображения, выгруженного из Active Directory, 100x100 пикселей.
2. Кликните правой кнопкой мыши по изображению и нажмите **Копировать**.



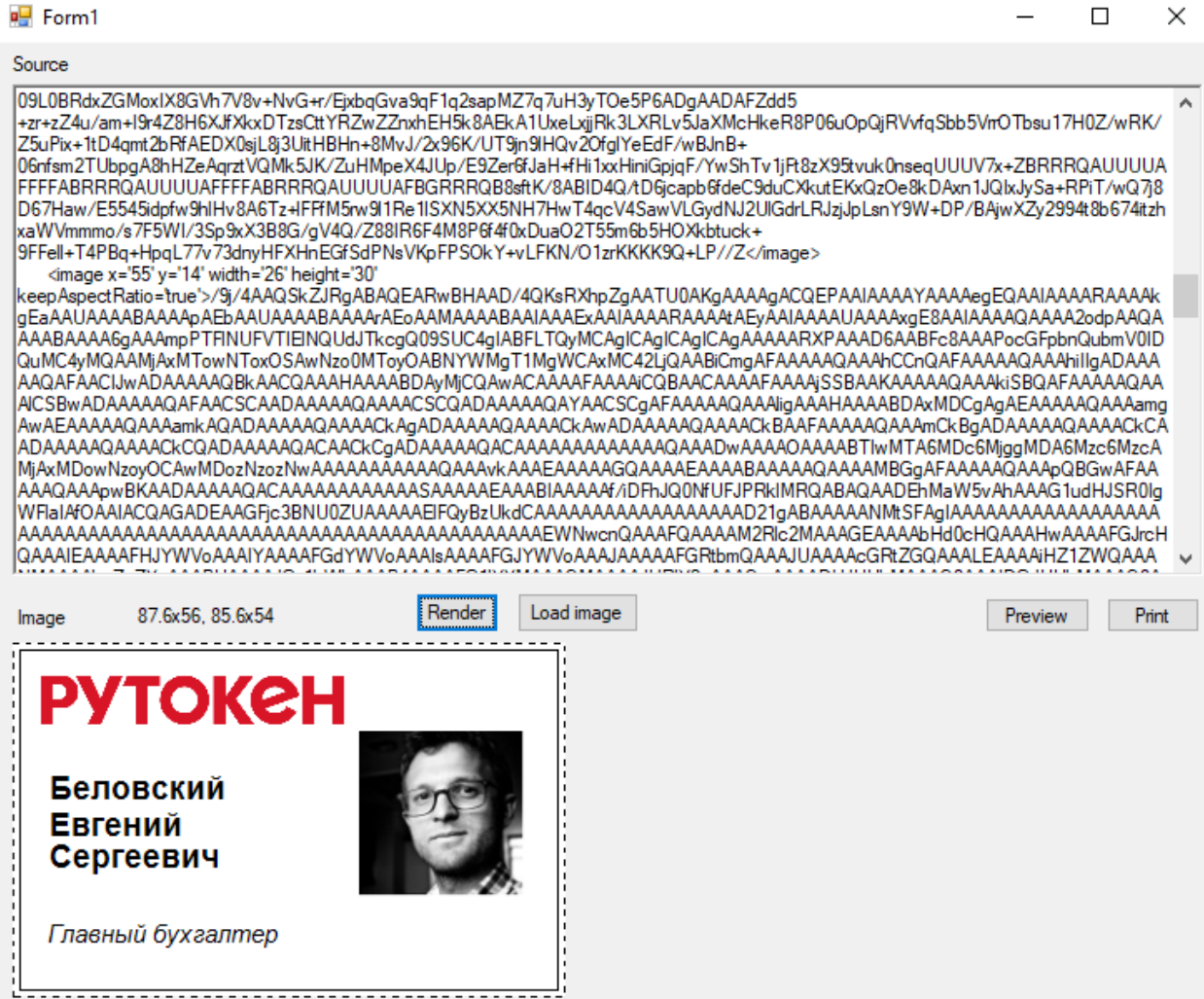
3. Выберите изображение и нажмите **Открыть**.
4. В секции `<image>` удалите `$user.Photo` и нажмите **Вставить**.
5. Нажмите **Render** для визуализации.

Нижe на рисунке приведен пример шаблона со статическим логотипом компании и подстановкой данных пользователя из Active Directory:

- Фамилии, имени и отчества с разбивкой по строкам, написанным жирным шрифтом.
- Фотографии.
- Должности, написанной курсивом.



Далее приведен пример ожидаемого результата печати по шаблону. Для визуализации вместо переменных `$user.Name.elementAt`, `$user.JobTitle` и `$user.Photo` в шаблоне прописаны данные пользователя и загружена его фотография.



Прямоугольник

Задается в секции `<rectangle>` с параметрами:

- `rep` (опциональный);
- `brush` (опциональный);
- `x`;
- `y`;
- `width` (опциональный);
- `height` (опциональный).

Стороны карты

Сторона карты задается в секциях `<frontSide>` (передняя) и `<backSide>` (задняя). Внутри секции располагаются элементы, определяющие то, что должно быть напечатано на стороне (`rectangle`, `text`, `image` и т.д.).

Вращение

Содержимое стороны карты может быть развернуто при необходимости. Для разворота в секции для стороны карты используются параметры:

- `rotationAngle`;
- `rotationOffsetX`;
- `rotationOffsetY`.

Пример

```
<frontSide rotationAngle='180' rotationOffsetX='43.8' rotationOffsetY='28'>
...
</frontSide>
```

На иллюстрации приведен пример перевернутого на 180 градусов базового шаблона печати с фотографией.

