Indeed-Id ESSO Template Wizard

Руководство по эксплуатации



© Компания «Индид», 2009 – 2018. Все права защищены.

Этот документ входит в комплект поставки продукта. Информация, содержащаяся в этом документе, может быть изменена разработчиком без уведомления пользователя.

8 (800) 333-09-06 телефон бесплатной горячей линии

ООО Индид ИНН/КПП 7801540219/780601001, ОГРН 1117847053103

8 (800) 333-09-06 или support@indeed-id.com служба поддержки пользователей

http://indeed-id.ru/ web-сайт компании

Оглавление

| Введение | 4 |
|----------------------------------------------------------------------------------------------|-------------|
| Условные обозначения | 4 |
| Общие сведения | 4 |
| Алгоритм работы утилиты | 4 |
| Установка Indeed-Id ESSO Template Wizard | 5 |
| Порядок создания шаблона приложения | 5 |
| Сбор информации о приложении | 5 |
| Создание слепка шаблона | 5 |
| Определение списка параметров для формы и контролов на форме | 5 |
| Выбор способа матчинга приложения | 6 |
| Настройка быстрого запуска приложения | 6 |
| Выбор алгоритма заполнения для Win32 приложения | 6 |
| Работа с Indeed-Id ESSO Template Wizard | 9 |
| Win32 мастер шаблонов | 9 |
| Web мастер шаблонов | 19 |
| Редактирование шаблона приложения | 28 |
| Матчинг приложения | 29 |
| Описание приложения | 31 |
| Операции приложения | 31 |
| Формы приложения | 34 |
| Матчинг Win32 формы | 37 |
| Матчинг Web формы | 37 |
| Перехват и заполение окон Windows Security для Basic/Digest аутентификации в Int Explorer | ernet 39 |
| Алгоритм заполнения | 39 |
| Ограничения генерируемых паролей | 43 |
| Параметры быстрого запуска | 45 |
| Уникальные (кастомные) поля | 45 |
| Сбор программных логов | 48 |
| Часто задаваемые вопросы | 48 |

Введение

Приветствуем вас и благодарим за использование продукта Indeed Enterprise Single Sign-On. Данное руководство поможет ознакомиться с принципом работы компонента **Indeed-Id ESSO Template Wizard**, параметрами его установки и настройки.

Условные обозначения

В Руководстве используются следующие условные обозначения:



Важная информация. Разделы, содержащие важную информацию, необходимую для успешной работы.



Дополнительная информация. Разделы, содержащие дополнительную информацию.

Общие сведения

Утилита Indeed-Id ESSO Template Wizard предназначена для подготовки win32– и web–шаблонов приложений, в которые предполагается получать доступ средствами Indeed Enterprise Single Sign–On (Indeed ESSO).

Алгоритм работы утилиты

Indeed-Id ESSO Template Wizard обрабатывает форму целевого приложения и формирует так называемый «слепок формы» или макет – набор данных, описывающих элементы формы (например, имя заголовка, идентификаторы кнопок и полей ввода данных). На основе данных, содержащихся в шаблоне клиентское приложение (Indeed ESSO Arent) отследит запуск целевого приложения на рабочей станции пользователя и, в зависимости от настроек для конкретного пользователя, предоставит или не предоставит в него доступ.



Полученный слепок является заготовкой для создания шаблона приложения. Создание полноценного шаблона осуществляется вручную в соответствии с рекомендациями, приведенными в данном руководстве.

Установка Indeed-Id ESSO Template Wizard

Indeed-Id ESSO Template Wizard следует устанавливать на рабочих станциях, где установлены приложения, для которых необходимо подготовить шаблон. Запустите программу установки **IndeedID.ESSO.Template.Wizard.msi** и следуйте указаниям Мастера установки.

После завершения установки компоненты Indeed-Id ESSO Template Wizard (Web-мастер шаблонов и Win32 мастер шаблонов) появятся в разделе **Все программы –> Indeed-Id –> Enterprise SSO**.

Порядок создания шаблона приложения

Сбор информации о приложении

- 1. Набор окон целевого приложения, которые будет обрабатывать SSO шаблон.
 - При этом можно ориентироваться на основные операции, которые способен выполнять шаблон:
 - вход в приложение
 - разблокировка
 - смена пароля в приложении.

Если пользовательский сценарий предполагает отказ от пароля (пользователь не знает свой пароль и использует аутентификатор или прозрачную аутентификацию для работы с целевым приложением) то все окна, в которых пользователю предлагается ввести пароль для приложения необходимо включить в список обрабатываемых окон. Помимо самих целевых окон важно обратить внимание на контролы, которые должны обрабатываться шаблоном. Уже на этом этапе вы сможете определить, понадобятся ли **Кастомные поля** в данном приложении.

- 2. Набор способов запуска приложения (через ярлыки, командные файлы, щелчком на файле данных приложения, запуская исполняемый файл приложения и т.п.). Эта информация необходима для настройки быстрого запуска приложения
- Набор исполняемых файлов, создающих процессы, отображающие графический интерфейс приложения (окна, на которые должен реагировать SSO Агент). Для каждого такого файла необходимо узнать имя, описание, расположение. Если у приложения только один ехе файл, то достаточно той информации, которую шаблонизатор соберёт во время работы.

Создание слепка шаблона

Определив список необходимых окон и операций необходимо можно преступить к созданию слепка (или слепков) шаблона. Для одного окна может потребоваться несколько слепков, также, в процессе отладки шаблона может возникнуть потребность создать дополнительные слепки, поэтому доступ к рабочей станции с установленным целевым приложением и Indeed-Id ESSO Template Wizard необходим на протяжении всего создания шаблона. Процесс создания слепка шаблона описан в разделе **Работа с Indeed-Id ESSO Template Wizard**.

Определение списка параметров для формы и контролов на форме

Для каждого целевого окна приложения необходимо определить список параметров, которые будут использованы для матчинга и алгоритма заполнения. Для Web и Win32 приложений есть разные наборы возможных параметров (см. **Win32 мастер шаблонов** и **Web Мастер шаблонов**). При необходимости, набор параметров следует пересмотреть после получения первых слепков приложения и во время отладки шаблона. В результате, выбранных параметров должно быть достаточно для однозначной идентификации формы целевого приложения.

Выбор способа матчинга приложения

Существует два основных способа матчинга исполняемого файла приложения:

- проверка пути до исполняемого файла.
- проверка crc и размера файла.

Эти способы можно использовать вместе или отдельно друг от друга. Рекомендуется использовать пути до исполняемых файлов, если исполняемых файлов и мест установки приложения немного и их все можно перечислить в шаблоне. Если необходимо обеспечить работу с приложением вне зависимости от расположения, то лучше использовать проверку сгс и размеров файла. Разбор параметров матчинга приложения описан в разделе **Матчинг приложения**.

Настройка быстрого запуска приложения

Быстрый запуск позволяет запускать приложения из меню быстрого запуска в SSO Areнте. Если приложение запускается через ярлык, то необходимо взять в качестве значения быстрого запуска командную строку из свойств ярлыка, если приложение запускается через исполняемый файл, то значение быстрого запуска должно быть командой запуска этого исполняемого файла. Если по клику на файл данных приложения – то это должна быть командная строка вида "<исполняемый_файл> <файл_данных>", т.е. файл данных передается как аргумент командной строки испоняемому файлу приложения.

Выбор алгоритма заполнения для Win32 приложения

В Indeed-Id Template Wizard позволяет использовать два алгоритма заполнения для win32-приложений: на основе **SetText**и **эмуляция ввода**.

Алгоритм с использованием **SetText** основан на стандартном методе контролов SetText. Основным требованием для использования данного метода является поддержка метода SetText контролами на форме. Самым простым способом проверить поддерживает ли контрол метод заполнения SetText служит создание шаблона с данным методом заполнения и проверка его на тестовом стенде.

По умолчанию Template Wizard создаёт команды с типом команды type="ctFillCtrl". Данный тип команд требует описания контрола, который необходимо заполнить в команде.

Пример алгоритма:

```
<fillingAlgorithm>
        <command type="ctFillCtrl">
          <description>Ввести имя пользователя</description>
          <dataFillCtrl fillingType="cftSetText" clearCtrlText="1">
           <ctrl className="Edit" ctrlId="1001" enabled="1" visible="1">
            <position x="159" y="15"/>
            <size cx="189" cy="23"/>
           </ctrl>
           <fieldValueInfo field="ffUserName"/>
          </dataFillCtrl>
        </command>
        <command type="ctFillCtrl">
          <description>Ввести пароль</description>
          <dataFillCtrl fillingType="cftSetText" clearCtrlText="1">
           <ctrl className="Edit" ctrlId="1002" enabled="1" visible="1">
            <position x="159" y="76"/>
            <size cx="189" cy="23"/>
           </ctrl>
           <fieldValueInfo field="ffPassword"/>
          </dataFillCtrl>
        </command>
        <command type="ctFillCtrl">
          <description>Нажать на OK</description>
          <dataFillCtrl fillingType="cftClickButton" clearCtrlText="1">
```

```
<ctrl className="Edit" ctrlId="1005" enabled="1" visible="1">
<position x="0" y="125"/>
<size cx="159" cy="23"/>
</ctrl>
<fieldValueInfo field="ffNoData"/>
</dataFillCtrl>
</command>
```

</fillingAlgorithm>

В примере приведены три команды заполнения контролов. Кратко рассмотрим теги содержащиеся в описании команд:

description – словесное описание команды, необходимо для упрощения редактирования шаблона и отображается в сообщения об ошибке, облегчая отладку шаблона.

dataFillCtrl – содержит данные целевого контрола формы.

fillingType – код алгоритма заполнения.

clearCtrlText – флаг очистки контрола перед заполнением. Если 1 – текст контрола будет очищен, если 0 – к тексту контрола добавится текст заполнения. Работает только для кода алгоритма заполнения cftSetText

ctrl – содержит параметры целевого контрола – в ходе выполнения данной команды SSO Agent будет искать на форме контрол, соответствующий данному описанию. Если ни один контрол не будет найден, то произойдет ошибка заполнения. Данный тип команды подходит для тех контролов, которые могут быть точно описаны в шаблоне (описание которых не совпадает с описание любого другого контрола на форме).

fieldValueInfo::field – код значения для заполнения поля формы (например field="ffUserName" – значение имени пользователя)

В случае, если контрол поддерживает использование SetText, но его нельзя точно описать с помощью его параметров, то можно использовать тип команды **ctFillCurrentCtrl**, заполняющий контрол находящийся в фокусе. В данном случае необходимо установить фокус в данный контрол посредством других команд (описание этих команд вы найдете ниже, в описании алгоритма эмуляци ввода).

Пример команды:

```
<command type="ctFillCurrentCtrl">
<description>Ввести имя пользователя</description>
<dataFillCurrentCtrl clearCtrlText="1" fillingType="cftSetText">
<fieldValueInfo field="ffUserName"/>
</dataFillCurrentCtrl>
```

Если метод SetText не работает в целевой форме, то необходимо использовать **эмуляцию ввода**. Данный способ использует эмуляцию нажатия и отпускания клавиш клавиатуры. Команды данного алгоритма обладают типом **ctFillCurrentCtrl**, т.е. заполняют только контрол, находящийся в фокусе.

Пример алгоритма заполнения:

```
<fillingAlgorithm>
<command type="ctWait">
<description>Подождать</description>
<dataWait periodMs="50"/>
</command>
<command type="ctPosition">
<description>Перейти на контрол имени пользователя</description>
<dataPosition waitIntervalMs="50">
<dataPosition >
<dataPosition waitIntervalMs="50">
<ctrl className="Edit" ctrlId="1001" enabled="1" visible="1">
<position x="159" y="15"/>
<size cx="189" cy="23"/>
</ctrl>
</dataPosition>
```

```
</command>
 <command type="ctWait">
  <description>Подождать</description>
  <dataWait periodMs="50"/>
 </command>
 <command type="ctFillCurrentCtrl">
  <description>Ввести имя пользователя</description>
  <dataFillCurrentCtrl clearCtrlText="1">
   <fieldValueInfo field="ffUserName"/>
  </dataFillCurrentCtrl>
 </command>
 <command type="ctWait">
  <description>Подождать</description>
  <dataWait periodMs="50"/>
 </command>
 <command type="ctControl">
  <description>Перейти на следующий контрол</description>
  <dataControl type="cctTab" waitIntervalMs="50"/>
 </command>
 <command type="ctWait">
  <description>Подождать</description>
  <dataWait periodMs="50"/>
 </command>
 <command type="ctFillCurrentCtrl">
  <description>Ввести пароль</description>
  <dataFillCurrentCtrl clearCtrlText="1">
   <fieldValueInfo field="ffPassword"/>
  </dataFillCurrentCtrl>
 </command>
 <command type="ctWait">
  <description>Подождать</description>
  <dataWait periodMs="50"/>
 </command>
 <command type="ctControl">
  <description>Перейти на следующий контрол</description>
  <dataControl type="cctTab" waitIntervalMs="50"/>
 </command>
 <command type="ctWait">
  <description>Подождать</description>
  <dataWait periodMs="50"/>
 </command>
 <command type="ctControl">
  <description>Нажать на OK</description>
  <dataControl type="cctEnter" waitIntervalMs="50"/>
 </command>
</fillingAlgorithm>
```

В данном примере содержится последовательность команд, среди которых:

- команда с параметром type="ctWait" добавляет паузу в алгоритм заполнения. Паузы позволяют избежать ошибок при переходе между полями и заполнением контролов. Параметр periodMs определяет продолжительность паузы в миллисекундах.
- команда с параметром type="ctPosition" переводит фокус на конкретный контрол, параметры которого описаны в команде. Шаблонизатор в данной команде использует описание контрола, указанного как поле имени пользователя.
- команда с параметром type="ctControl" осуществляет перевод фокуса. Перевод осуществляется за счет эмуляции нажатия клавиши Tab.

Команды паузы и перевода контрола можно добавлять в алгоритм заполнения методом SetText по необходимости.

Работа с Indeed-Id ESSO Template Wizard

Win32 мастер шаблонов

Генерация слепка шаблона

Запустите приложение Enterprise SSO – Win 32 Мастер шаблонов. Стартовое окно будет иметь вид, как на Рисунке 1а. Откройте приложение, шаблон для которого необходимо создать (например, 1С Предприятие), и затем перейдите на его форму входа (Рисунок 1b).

| ESSO Мастер шаблонов | |
|-----------------------------------------------|----------|
| Определение окна формы | |
| Пожалуйста, перейдите на целев | вое окно |
| и нажмите Ctrl+Shift+T | |
| | Выход |
| Рисунок 1а – Стартовое окно Win32 мастера шаб | лонов. |

| 1С:Предприятие. | Доступ к информ | пационной базе | ¢ |
|-----------------|-----------------|----------------|---|
| 1@ | Бухгалтерия г | редприятия | |
| | Пользователь: | | |
| | Пароль: | | |
| | | Отмена | |

Рисунок 1b – Форма входа целевого win32-приложения.

Нажмите комбинацию клавиш **Ctr+Shift+T** для обработки формы целевого приложения. Укажите необходимые параметры в окне настроек генерации шаблона (Рисунок 1с). Ниже приведено описание всех перечисленных настроек приложения.

Indeed-Id ESSO Template Wizard Руководство по эксплуатации

| ESSO Macte | ер шаблонов | |
|--------------------------|---------------------------|--------------------------|
| Настройки генера | ции шаблона | |
| Глубина просмотра контро | пов формы: глубина контро | олов выбранной формы: 2 |
| Для формы: | Для контролов формы: | Алгоритм заполнения: |
| V заголовок | Ctrlid | нет |
| 🔽 имя класса окна | имя класса окна | 🔘 на основе settext |
| 🔽 размеры | 📝 текст | 💿 эмуляция ввода |
| | 🔽 размеры | |
| | 🔽 позиция | |
| | 📝 флаг Visible | |
| | 📝 флаг Enabled | |
| | Перепривязаты | к контролам Генерировать |
| | | Выход |

Рисунок 1с – Настройки генерации шаблона win32 приложения.

Глубина контролов и глубина просмотра контролов

Глубина просмотра контролов формы

С помощью данного параметра можно задать глубину вложенности элементов управления формы, на основе которой Мастер шаблонов построит шаблон. Данный параметр задаётся вручную при создании шаблона.

Если параметр равен 0, то Мастер шаблонов соберёт информацию только о невложенных элементах управления. Если параметр равен 1, то Мастер шаблонов соберёт данные о невложенных элементах управления и о контрэлементах управления первого уровня вложенности и т.д.

Глубина контролов выбранной формы

Данный параметр отображает максимальную глубину вложенности элементов управления на выбранной форме (сколько уровней вложенности элементов управления существует на форме). При выборе значения параметра "Глубина просмотра контролов формы" рекомендуется ориентироваться на данный параметр. Значение глубины контролов для выбранной формы Мастер шаблонов вычисляет индивидуально для каждой формы в момент определения целевой формы приложения.

В таблицах 1 и 2 описаны параметры построения шаблона для формы и контролов формы соответственно. Таблица 3 содержит описание параметров алгоритма заполнения.

Настройки формы целевого приложения

Данный набор настроек позволяет задать совокупность признаков формы приложения, которые будут учитываться Агентом SSO при поиске целевого окна среди множества окон целевого приложения.

Параметр Описание При включении данной настройки значение заголовка формы будет Заголовок учитываться в процессе отслеживания целевой формы приложения SSO. В шаблон будет добавлено фиксированное значение заголовка для сравнения. Если настройка выключена, значение заголовка формы при определении целевой формы приложения учитываться не будет. Рекомендуется использовать данный параметр в шаблоне, кроме случаев, когда заголовок динамически изменяется. При включении данной настройки в процессе отслеживания целевой Имя класса окна формы приложения SSO будет учитываться имя класса окна (задается на этапе разработки приложения) для данной формы. В шаблон будет добавлено фиксированное значение имени класса окна для сравнения. Если настройка выключена, имя класса окна при определении целевой формы приложения учитываться не будет. Рекомендуется использовать данный параметр в шаблоне, кроме случаев, когда имя класса изменяется динамически. При включении данной настройки в процессе отслеживания целевой Размеры формы приложения SSO будут учитываться жестко зафиксированные размеры окна (ширина и высота, задаются в пикселях). В шаблон будет добавлено фиксированное значение размеров формы для сравнения. Если настройка выключена, размеры окна при определении целевой формы приложения учитываться не будут. Использовать данный параметр следует только в тех случаях, когда остальные параметры не позволяют достаточно точно описать целевую форму приложения.

Таблица 1 – Настройки формы целевого wein32-приложения.

Настройки контролов формы целевого приложения

Данный набор настроек позволяет задать совокупность признаков элементов управления, присутствующих в окне приложения, которые будут учитываться Агентом SSO при поиске целевого окна среди множества окон целевого приложения.

Таблица 2 – Настройки контролов формы целевого win32-приложения.

| Параметр | Описание |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CtrId | Включение данной настройки говорит о том, что в процессе отслеживания целевой формы приложения SSO будут учитываться идентификаторы (задаются на этапе разработки приложения) всех элементов управления, которые входят в состав окна. Значения всех идентификаторов элементов управления будут зафиксированы в шаблоне. Если настройка выключена, идентификаторы элементов управления при определении целевой формы приложения учитываться не будут. Рекомендуется использовать данный параметр, кроме случаев, когда Id контролов в целевой форме изменяется динамически. |
| Имя класса окна | При включении данной настройки в процессе отслеживания целевой формы приложения SSO будет учитываться имя класса окна (задается на этапе разработки приложения) для для каждого элемента управления, входящего в состав формы. В шаблон будет добавлено фиксированное значение имени класса окна для сравнения. Если настройка выключена, имя класса окна для элементов управления при определении целевой формы приложения учитываться не будет. Рекомендуется использовать данный параметр в шаблоне, кроме |
| | случаев, когда имя класса изменяется динамически. |
| Текст | Включение данной настройки позволяет учитывать текст, содержащийся в элементах управления в окне приложения (поля ввода, выпадающие списки и т.д.), в процессе отслеживания целевой формы приложения SSO. Параметр учитывается для всех элементов управления, которые входят в состав окна. Значения всех текстовых строк для элементов управления будут зафиксированы в шаблоне. |
| | Если настройка выключена, текст элементов управления при определении целевой формы приложения учитываться не будет. Внимание: |
| | При создании шаблона обращайте внимание на изменение текста элементов управления. При различных обстоятельствах (смена языка в системе, смена языка в приложении, изменение версии приложение, смена пользователя и т.д.) текст в элементе управления может измениться, что может помешать процедуре отслеживания приложения. Не используйте динамически изменяющийся текст в шаблоне, либо учитывайте все возможные варианты данного параметра. |

| Размеры | При включении данной настройки в процессе отслеживания целевой формы приложения SSO будут учитываться жестко зафиксированные размеры (ширина и высота, задаются в пикселях) для всех элементов управления в окне. В шаблон будет добавлено фиксированное значение размеров для сравнения. Если настройка выключена, размеры элементов управления при определении целевой формы приложения учитываться не будут. Использовать данный параметр следует только в тех случаях, когда остальные параметры не позволяют достаточно точно описать свойства элементов управления целевой формы приложения. |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Позиция | При включении данной настройки в процессе отслеживания целевой формы приложения SSO будет учитываться жестко зафиксированная позиция (позиция левого верхнего угла элемента управления относительно левого верхнего угла контейнера, содержащего этот элемент управления, в пикселях) для всех элементов управления в окне. В шаблон будет добавлено фиксированное значение позиции для сравнения. Если настройка выключена, позиция элементов управления при определении целевой формы приложения учитываться не будет. Использовать данный параметр следует только в тех случаях, когда остальные параметры не позволяют достаточно точно описать свойства элементов управления целевой формы приложения. |
| Visible | Включение данной настройки позволяет в процессе отслеживания целевой формы приложения SSO учитывать признак видимости (виден ли присутствующий в форме элемент управления для пользователя) всех элементов управления, которые входят в состав окна. Значения свойства видимости для элементов управления будут зафиксированы в шаблоне. Если настройка выключена, видимость элементов управления при определении целевой формы приложения учитываться не будет. |
| Enable | Включение данной настройки позволяет в процессе отслеживания целевой формы приложения SSO учитывать признак доступности (доступен ли элемент управления для действий пользователя) всех элементов управления, которые входят в состав окна. Значения свойства доступности для элементов управления будут зафиксированы в шаблоне. Если настройка выключена, доступность элементов управления при определении целевой формы приложения учитываться не будет. |

Настройки алгоритма заполнения

В данном разделе производится выбор алгоритма заполнения, который будет использован в процессе подстановки имени пользователя и пароля в целевую форму приложения SSO.

| | Таблица 2 – Настройка | алгоритма заполнени | ия для win32-приложения |
|--|-----------------------|---------------------|-------------------------|
|--|-----------------------|---------------------|-------------------------|

| Параметр | Описание |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Нет | Алгоритм заполнения не будет сгенерирован в шаблоне. Автоматическое заполнение формы происходить не будет. |
| На основе SetText | После выбора данной настройки Мастер шаблонов сгенерирует команды заполнения поля ввода имени пользователя, заполнения поля ввода пароля и нажатия кнопки "ОК". Каждая команда будет содержать данные элемента управления, к которому она должна быть применена. Заполнения элементов управления формы будет производиться стандартным методом SetText. Рекомендуется использовать данный алгоритм заполнения. |
| Эмуляция ввода | После выбора данной настройки Мастер шаблонов сгенерирует команды установки фокуса в поле ввода имени пользователя, заполнения поля ввода имени пользователя, заполнения поля ввода пароля, нажатия кнопки "ОК", команды перехода между элементами управления и команды паузы. Само заполнение будет происходить с помощью эмуляции нажатия клавиш клавиатуры. Переход фокуса между элементами управления будет осуществляться эмуляцией нажатия клавиши Tab. В алгоритме заполнения также будут присутствовать команды паузы в 50 миллисекунд командами заполнения и перехода. Рекомендуется к использованию в шаблонах для приложений, в которых использование метода settext невозможно или затруднено. |



При создании шаблона для конкретного приложения необходимо задать только тот набор настроек, который будет отличать это приложение от других. Перед генерацией шаблона важно понять, по каким признакам будет определяться форма входа в приложение, и какой алгоритм заполнения необходимо применить.

| ESSO Macre | ер шаблонов | |
|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| Настройки генера | ции шаблона | |
| Глубина просмотра контро | лов формы: 2 глубина контро | олов выбранной формы:2 |
| Для формы: ☑ заголовок ☑ имя класса окна □ размеры | Для контролов формы: ✓ ctrlid ✓ имя класса окна □ текст □ размеры □ позиция ✓ флаг Visible ✓ флаг Enabled | Алгоритм заполнения: О нет О на основе settext ⊙ эмуляция ввода |
| | <u>Перепривязать</u> | к контролам Генерировать |
| | | Выход |

Рисунок 2а – Пример настроек для создания шаблона win32-приложения.

Для определения контролов выполните следующие действия:

- в целевом окне перейти на контрол Имя пользователя и нажать Ctrl+Shift+T (Рисунок 2b)
- в целевом окне перейти на контрол Пароль и нажать Ctrl+Shift+T (Рисунок 2с)
- в целевом окне перейти на контрол кнопки **ОК** и нажать **Ctrl+Shift+T** (Рисунок 2d)

После определения контролов будет создан макет шаблона (Рисунок 2е).



Рисунок 2b – Определение контрола имени пользователя формы win32-приложения.



Рисунок 2с – Определение контрола пароля формы win32-приложения.



Рисунок 2d – Определение контрола кнопки ОК пароля формы win32-приложения.

| ESSO Мастер шаблонов | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| Шаблон | |
| <indeedidssodata></indeedidssodata> | |
| <applications> <application forcereauthentication<br="" ssopermissions="spSsoOnly">55909F9523F9}"> <matchingdetails> <components> <component 534832"="" componentid="Win3 <componentWin32> <paths> <paths> <paths> <crcs> <crc fileSizeBytes=" type="CFormData::mtWin32">2192561669 <filedescription>1cv8c</filedescription></component></components></matchingdetails></application></applications> | n="1" id="{47491552-D8A0-4887-82DC- 32Component1"> c.exe |
| Сохранять в UTF-16 | <u>К настройкам</u> <u>Сохранить</u> |
| | Выход |

Рисунок 2е – Макет шаблона win32-приложения.

Для сохранения макета шаблона нажмите **Сохранить** (Рисунок 2е). При сохранении шаблона обязательно установите опцию **Сохранять в UTF-16**. Шаблон будет сохранен по указанному пути с расширением .xml. Готовый макет шаблона в дальнейшем подвергается ручной доработке.

Следует обратить внимание на следующие моменты:

- Прописать в тэге application/userdescription название приложения. Именно это название будет видеть пользователь в окне ESSO Агент на своей рабочей станции.
- Если в приложении есть дополнительные поля, которые должны быть заполнены при входе в приложение, то эти поля необходимо описать самостоятельно в тэге application/customFields.
- Если при генерации макета шаблона был выбран «Алгоритм заполнения» *эмуляция ввода*, то необходимо внимательно исследовать в каком поле изначально установлен курсор и в какой последовательности он перемещается по остальным контролам. По результатам этого исследования отредактировать алгоритм заполнения формы в шаблоне, это тэг орXXX/forms/formdata/fillingAlgorithm.

Исправленный шаблон обязательно сохраните с расширением .app.

Web мастер шаблонов

Генерация шаблона

Запустите приложение Enterprise SSO – Web Мастер шаблонов. Стартовое окно будет иметь вид, как на Рисунке За.



Рисунок За – Стартовое окно Web мастера шаблонов.

Запустите приложение, шаблон для которого необходимо создать в браузере Internet Explorer. Находясь на целевой странице приложения нажмите **Ctrl+Shift+T**. В этот момент Мастер определит целевое окно и затем отобразит следующий диалог (Рисунок 3b):

| ESSO Мастер шаблонов |
|-----------------------------------------------------------------------------------------------------------------------------|
| Определение особенностей формы |
| Пожалуйста, выделите часть целевой веб-страницы, которая отличает ее от других страниц, и нажмите Ctrl+Shift+T |
| Определено узлов HTML DOM дерева страницы: 0 <u>Генерировать шаблон</u> |
| Выход |

Рисунок 3b – Определение структуры web-формы.

Перейдите на целевую страницу, выделите небольшой участок, на котором располагаются поля ввода имени пользователя, пароля, кнопки и нажмите **Ctrl+Shift+T**.

В этот момент Мастер определяет узлы HTML DOM дерева. По завершению операции число определенных узлов будет указано в нижней части окна в строке **Определено узлов HTML DOM дерева страницы**.



Этот шаг может занять немного времени. Длительность работы Мастера зависит от количества объектов обрабатываемого участка web-страницы. Не спешите закрывать страницу или окно Мастера.

После определения особенностей страницы нажмите **Генерировать шаблон** и укажите параметры его генерации (Рисунок 3с).

| ESSO Macte | ер шаблонов | |
|------------------------------|------------------------------------------------|-------------------------------------------------|
| Настройки генера | ции шаблона | |
| Для формы: 🔽 url страницы | Для контролов формы: 🗹 имя тега | Алгоритм заполнения: Нет |
| 🗹 заголовок страницы | текст атрибут id | С на основе fillWebCtrl |
| | 🗹 атрибут name | |
| | 🗋 другие атриоуты 🛄 🔲 стилевые атрибуты 🛄 | |
| | если есть id, подавлять все остально | be |
| | использовать текст, только когда нет других | c |
| | Перепривязать н | к контролам Генерировать |
| | | Выход |

Рисунок 3с – Настройки генерации шаблона web-приложения.

Настройки формы целевого приложения

Данный набор настроек позволяет задать совокупность признаков web-страницы приложения, которые будут учитываться Агентом SSO при поиске целевой страницы среди множества загруженных страниц браузера.

| Таблица 4 – Наст | гройки формы | целевого we | еb-приложения. |
|------------------|--------------|-------------|----------------|
|------------------|--------------|-------------|----------------|

| Параметр | Описание |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| URL страницы | При включении данной настройки в процессе отслеживания целевой web-страницы SSO будет учитываться url страницы. В шаблон будет добавлено фиксированное значение url для сравнения. Если настройка выключена, url при определении целевой web-страницы учитываться не будет. Рекомендуется использовать данный параметр в шаблоне, кроме случаев, когда url динамически изменяется (для таких страниц необходимо либо отказаться от использования данного параметра, либо использовать регулярное выражение). |
| Заголовок страницы | При включении данной настройки значение заголовка страницы будет учитываться в процессе отслеживания целевой web-страницы SSO. В шаблон будет добавлено фиксированное значение заголовка для сравнения. |

| Если настройка выключена, значение заголовка при определении целевой web–страницы учитываться не будет. |
|---------------------------------------------------------------------------------------------------------|
| Рекомендуется использовать данный параметр в шаблоне, кроме |
| случаев, когда заголовок динамически изменяется (для таких страниц |
| необходимо либо отказаться от использования данного параметра, либо использовать регулярное выражение). |

Настройки контролов формы целевого приложения

Данный набор настроек позволяет задать совокупность признаков элементов управления web, присутствующих на странице, которые будут учитываться Агентом SSO при поиске целевой страницы среди множества загруженных страниц браузера.

Таблица 5 – Настройки контролов формы целевого win32-приложения.

| Параметр | Описание |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Имя тега | При включении данной настройки в процессе отслеживания целевой страницы SSO будет учитываться имя тега для каждого web-элемента, входящего в состав DOM-дерева фрагмента страницы, выделенного на этапе подготовки к генерации шаблона. В шаблон будет добавлено фиксированное значение имени тега для сравнения. Если настройка выключена, имя тега для web-элементов при определении целевой формы приложения учитываться не будет. Рекомендуется использовать данный параметр в шаблоне, кроме случаев, когда имя тега изменяется динамически. |
| Текст | При включении данной настройки в процессе отслеживания целевой страницы SSO будет учитываться текст для каждого web-элемента, входящего в состав DOM-дерева фрагмента страницы, выделенного на этапе подготовки к генерации шаблона. В шаблон будет добавлено фиксированное значение текста для сравнения. Если настройка выключена, значение текстовой составляющей для web-элементов при определении целевой формы приложения учитываться не будет. |
| | При использовании данного параметра рекомендуется проверить получившийся шаблон – массив текста в шаблоне может превышать ожидаемый, что приведёт к проблеме с отслеживанием целевой web-страницы и усложнит редактирование шаблона. |
| Атрибут id | При включении данной настройки в процессе отслеживания целевой страницы SSO будет учитываться значение атрибута id для каждого web-элемента, входящего в состав DOM-дерева фрагмента страницы, выделенного на этапе подготовки к генерации шаблона. В шаблон будет добавлено фиксированное значение атрибутов id для сравнения. Если настройка выключена, значение атрибута id для web-элементов при определении целевой формы приложения учитываться не будет. Рекомендуется к использованию. |

| Атрибут name | При включении данной настройки в процессе отслеживания целевой страницы SSO будет учитываться значение атрибута name для каждого web-элемента, входящего в состав DOM-дерева фрагмента страницы, выделенного на этапе подготовки к генерации шаблона. В шаблон будет добавлено фиксированное значение атрибутов name для сравнения. Если настройка выключена, значение атрибута name для web-элементов при определении целевой формы приложения учитываться не будет. Рекомендуется к использованию. |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Другие атрибуты | При включении данной настройки в процессе отслеживания целевой страницы SSO будет учитываться набор значений для указанного списка атрибутов для каждого web-элемента, входящего в состав DOM-дерева фрагмента страницы, выделенного на этапе подготовки к генерации шаблона. Атрибуты для отслеживания необходимо выбрать из открывающегося списка (открывается нажатием). В шаблон будут добавлены фиксированные значения указанных атрибутов для сравнения. Данный параметр рекомендуется использовать в том случае, когда остальные параметры не могут однозначно описать элемент формы. |
| Стилевые атрибуты | При включении данной настройки в процессе отслеживания целевой страницы SSO будет учитываться набор значений для указанного списка стилевых атрибутов для каждого web-элемента, входящего в состав DOM-дерева фрагмента страницы, выделенного на этапе подготовки к генерации шаблона. Стилевые атрибуты для отслеживания необходимо выбрать из открывающегося списка (открывается нажатием). В шаблон будут добавлены фиксированные значения указанных стилевых атрибутов для сравнения. Данный параметр рекомендуется использовать в том случае, когда остальные параметры не могут однозначно описать элемент формы. |
| Если есть id, подставлять все остальное | Если у веб элемента присутствует атрибут id, все другие атрибуты для данного элемента не будут использованы в шаблоне. Данную настройку можно применять, если атрибут id – основной параметр, по которому будет происходить определение целевой страницы. |
| Использовать текст, только когда нет других | Атрибут text будет использован, если никаких других указанных атрибутов для данного элемента не будет найдено. В противном случае, атрибут text не будет использован в шаблоне. Рекомендуется к использованию. |

Настройки алгоритма заполнения

В данном разделе производится выбор алгоритма заполнения, который будет использован в процессе подстановки имени пользователя и пароля в целевую web-форму приложения SSO.

| Таблица 6 – Настройка алгоритма заг | полнения для web-приложения |
|-------------------------------------|-----------------------------|
|-------------------------------------|-----------------------------|

| Параметр | Описание |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Нет | Алгоритм заполнения не будет сгенерирован в шаблоне. Автоматическое заполнение формы происходить не будет. |
| На основе fillWebCtrl | После выбора данного пункта, Мастер шаблонов запросит у пользователя указать 3 контрола на web-форме с помощью сочетания клавиш Ctrl+Shift+T и сгенерирует 3 команды в алгоритме заполнения для данных элементов: первая команда – заполнение веб контрола значением логина пользователя; вторая команда – заполнение веб значением пароля учетной записи SSO; третья команда – нажатие на кнопку "ОК". В каждой команде содержится данные об элементе на форме, с которым данная команда работает. Аналог алгоритма SetText для Win32 приложений. |



При создании шаблона для конкретного приложения необходимо выбрать только тот набор настроек, который будет отличать это приложение от других.

Перед генерацией шаблона важно понять, какие контролы будут задействованы в определении формы входа web-приложения.

После определения параметров создания шаблона нажмите кнопку **Генерировать** (Рисунок 3с). Если был задан **Алгоритм заполнения**, то появится следующее окно (Рисунок 4а).



Рисунок 4а – Определение контрола имени пользователя формы web-приложения.

Для определения контролов выполните следующие действия:

- в целевом окне перейти на контрол Имя пользователя и нажать Ctrl+Shift+T (Рисунок 4а)
- в целевом окне перейти на контрол Пароль и нажать Ctrl+Shift+T (Рисунок 4b)
- в целевом окне перейти на контрол кнопки **ОК** и нажать **Ctrl+Shift+T** (Рисунок 4с)



Рисунок 4b – Определение контрола пароля формы web-приложения.



Рисунок 4с – Определение контрола кнопки ОК пароля формы web-приложения.

| ESSO Мастер шаблонов | Б И I | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|------------------|
| Шаблон | | |
| <indeedidssodata></indeedidssodata> | | _ |
| <applications> <application forcereauthenticatio<br="" ssopermissions="spSsoOnly">BF34C053B598}"> <matchingdetails> <components> <component <="" componentid="Web" td="" type="CEormData::mtWeb"><td>n="1" id="{E76E0FD2-6DE6-</td><td>4D7A-A8A6-</td></component></components></matchingdetails></application></applications> | n="1" id="{E76E0FD2-6DE6- | 4D7A-A8A6- |
| <componentweb> <url>http://iiddemoserver.indeed-id.local/IndeedID.SSO.De </url></componentweb> | mo.App/IndeedID.SSO.Dem | o.App.html≺/url≻ |
| | emptsLimit="2"> | _ |
| Сохранять в UTF-16 | <u>К настройкам</u> | Сохранить |
| | | Выход |

Рисунок 4d – Макет шаблона web-приложения.

Для сохранения макета шаблона нажмите **Сохранить** (Рисунок 4d). При сохранении шаблона обязательно установите опцию **Сохранять в UTF-16**. Шаблон будет сохранен по указанному пути с расширением .xml. Готовый макет шаблона в дальнейшем подвергается ручной доработке.

Шаблон будет сохранен по указанному пути с расширением .xml. Готовый макет шаблона в дальнейшем подвергается ручной доработке. Исправленный шаблон обязательно сохраните с расширением **.app**.

Редактирование шаблона приложения

После получения .app-файла шаблона его необходимо отредактировать для работы с целевым приложением в соответствии с требуемым сценарием доступа в приложение. Для дальнейшей работы с шаблоном обратитесь к следующим разделам данного руководства:

- Матчинг приложения
- Описание приложения
- Операции приложения
- Формы приложения
- <u>Матчинг Win32 формы</u>
- Матчинг Web формы
- Алгоритм заполнения
- Ограничения генерируемых паролей
- Параметры быстрого запуска
- Уникальные (кастомные) поля

Матчинг приложения

Матчингом приложения называется процесс определения целевого приложения ESSO Агентом. Механизм матчинга целевого приложения ESSO Агентом основан на сопоставлении данных, указанных в шаблоне, с данными приложения. После загрузки шаблона в систему ESSO, ESSO Агент из множества приложений, с которыми работает пользователь, обрабатывает только те приложения, параметры которых соответствуют заданным в шаблоне данным.

Параметрами матчинга для Win32-приложений могут быть пути, контрольные суммы, размер, описания исполняемых файлов приложения. Матчинг Web-приложений выполняется по URL-адресу страницы.



Внимание! Матчинг форм приложения выполняется отдельно от матчинга самого приложения, поэтому эти два понятия нельзя путать.

Синтаксис

Матчинг приложения в шаблоне выглядит следующим образом:

```
<matchingDetails>
 <components>
  <component type="тип компонента" componentId="ID компонента">
    <componentWeb>
      <url isRegex="флаг регулярного выражения">URL страницы</url>
     </componentWeb>
   </component>
   <component type="тип компонента" componentId="ID компонента">
    <componentWin32>
      <paths>
        <path>путь к файлу 1</path>
        <path>путь к файлу N</path>
      </paths>
      <crcs>
        <crc fileSizeBytes="paзмер файла 1 в байтах">CRC файла 1</crc>
        <crc fileSizeBytes="размер файла N в байтах" >CRC файла N</crc>
      </crcs>
      <fileDescription>Описание исполняемого файла</fileDescription>
     </componentWin32>
   </component>
 </components>
</matchingDetails>
```

Описание приведенной выше структуры:

<matchingDetails>. Определяет в шаблоне параметры матчинга. Включает в себя весь набор параметров, используемых для определения приложения.

<components>. Описывает компоненты приложения. Некоторые приложения имеют несколько представлений пользовательского интерфейса – как правило, это представления в виде Win32– и Web–приложений (например, веб–службы электронной почты). Представление пользовательского интерфейса в терминологии Indeed-Id называется компонентом приложения.

Компоненты приложения можно редактировать в Консоли управления ESSO (создавать и удалять компоненты через ESSO Консоль нельзя).

Включает в себя следующие атрибуты:

- component type. Тип компонента приложения приложения. Представление в виде Win32-приложения называется Win32-компонентом приложения. Представление в виде Web-приложения называется Web-компонентом приложения.
 Возможные значения:
 - о *CFormData::mtWin32* Win32-компонент приложения

- о *CFormData::mtWeb* Web-компонент приложения
- **componentId**. Идентификатор компонента приложения. Идентификатор компонента используется для привязки форм приложения к компоненту и представляет собой произвольную строку, уникальную в списке идентификаторов компонентов данного приложения. Для того, чтобы привязать форму к компоненту приложения, нужно в описании формы задать идентификатор соответствующего компонента:

```
<formData id="{74034A01-12DA-424F-B081-BB88F8010BE0}" filledWndMarkDeletionTime="fwmdtOnFillingFinished">
<matching matchingType="CFormData::mtWin32">
```

<matchingWin32 className="#32770" componentId="<Win32Component>">

```
//matchingWin32>
</matching>
```

</formData>

Идентификатор может содержать специальные символы, для задания которых используются сущности, определенные в формате XML, а именно:

| Символ | Сущность |
|--------|----------|
|--------|----------|

- < <
- > &qt;
- " "
 - '

Пример:

<component type="CFormData::mtWeb" componentId="<Web Component>">... </component>

Приведенное описание определяет компонент с имененем < Web Component>. Идентификатор компонента отображается в ESSO Консоли в настройках приложения.

<componentWeb>. Описывает web-компонент приложения. Включает в себя параметр url и его атрибут isRegex.

- **<url>** задает адрес, по которому будет выполняться матчинг Web-приложения. Значение может быть задано в виде текста или регулярного выражения (определяется флагом isRegex).
- **isRegex** флаг представления URL страницы в виде регулярного выражения. Возможные значения: 0 URL определяется текстом; 1 URL определяется регулярным выражением.

<componentWin32>. Описывает win32-компонент приложения. Включает в себя следующие параметры:

- <paths>. Структура задает пути исполняемых файлов компонентов приложения. Матчинг компонентов приложения выполняется по заданным путям. Параметр опциональный. Значение по умолчанию: пусто (нет путей).
 - о **<path>**. Путь исполняемого файла компонента приложения для матчинга.
- <crcs>. Структура задает контрольные суммы исполняемых файлов компонентов приложения.
 Значения структуры представлены парами <CRC, размер файла>. Параметр опциональный.
 Значение по умолчанию: пусто (нет пар <CRC, размер файла>).
 - <crc>. Контрольная сумма исполняемого файла компонента приложения. Содержит атрибут fileSizeBytes, задающий размер исполняемого файла компонента приложения.

<fileDescription>. Описание исполняемого файла, указанное в его свойствах (в Version Info). Значение параметра может быть пустым, если описание не задано в свойствах файла или выполнять проверку описания не требуется.

Ниже приведен пример заданных параметров матчинга Win32- и Web-компонентов приложения IndeedID.SSO.Test.App.exe:

```
<matchingDetails>
 <components>
  <component type="CFormData::mtWin32" componentId="&lt;Win32Component&gt;">
    <componentWin32>
      <paths>
        <path>%ProgramFiles%\Indeed-Id\Enterprise SSO\IndeedID.SSO.Test.App.exe</path>
        <path>C:\IndeedID.SSO.Test.App.exe</path>
      </paths>
      <crcs>
        <crc fileSizeBytes="270848">4135804682</crc>
      </crcs>
      <fileDescription>Indeed-Id SSO test application</fileDescription>
     </componentWin32>
   </component>
<component type="CFormData::mtWeb" componentId="WebComponent">
    <componentWeb>
      <url isRegex="1">(.*)\\IndeedID\.SSO\.web\.Test\.App\\(.*)\.htm.*</url>
    </componentWeb>
   </component>
 </components>
```

Описание приложения

Описание приложения необходимо для идентификации приложения и отображения информации о нем в системе ESSO. Обычно в качестве описания используют название приложения. Описание приложения должно быть уникальным (в противном случае при загрузке шаблона возникнет ошибка).

Указанное описание отображается в заголовке записи приложения в списке зарегистрированных приложений в ESSO Консоли и в списке приложений в окне быстрого запуска.

Синтаксис

Описание приложения в шаблоне задается параметром **<userDescription>**. Значение параметра представляет собой произвольную строку.

Пример:

<userDescription>Indeed-Id SSO test application</userDescription>

Операции приложения

Операцией называется процесс выполнения действий, необходимых для аутентификации (вход в приложение, смена пароля, разблокировка).

Шаблоны позволяют задать три операции:

- операция входа <opLogon>
- операция смены пароля <opPasswordChange>
- операция разблокировки <opUnlock>

Все операции являются опциональными. Описания всех операций одинаковы за исключением непосредственно названия операции.

В ходе выполнения операции происходят изменения в пользовательском интерфейсе приложения: в нем появляются и исчезают окна, изменяется их содержимое и т.д.

Синтаксис

Описание операций приложения в шаблоне выглядит следующим образом:

<[операция] timeoutMs="таймаут" failureReaction="реакция на ошибку" rematchAttemptsLimit="лимит попыток рематчинга">> <forms>

```
<formData>
    [данные формы 1]
   </formData>
  ...
   <formData>
    [данные формы N]
   </formData>
 </forms>
 <resultSigns>
   <formsOpOk>
    [данные формы]
   </formsOpOk>
   <formsOpFailed>
    [данные формы]
   </formsOpFailed>
 </resultSigns>
 <failureReactionAskingDetails isDefaultOptionDisabled="флаг">
   <disabledOptions>
     <failureReaction>[опция 1]</failureReaction>
     <failureReaction>[опция M]</failureReaction>
   </disabledOptions>
 </failureReactionAskingDetails>
</[операция]>
```

<[операция]>

Атрибуты операции:

timeoutMs. Таймаут операции (в миллисекундах). Параметр имеет смысл только для операций с заданной формой-признаком ошибки. Операция завершается ошибкой после матчинга формы-признака ошибки по истечении заданного времени.

Если таймаут операции не задан, операция завершается ошибкой сразу после матчинга формы-признака ошибки.

После завершения операции с ошибкой выполняется действие, заданное в качестве реакции на ошибку (атрибут failureReaction). Опциональный атрибут. Значение по умолчанию = 0.

failureReaction. Реакция на ошибку операции. Описывает действие, которое должно быть выполнено в том случае, если операция завершилась ошибкой.

Возможные значения:

- frRematchTargetForm выполнять рематчинг (повторный матчинг) формы. При завершении операции с ошибкой будет выполнен повторный матчинг формы (если она продолжает существовать) и начато выполнение новой операции.
 Количество попыток рематчинга определяется значением атрибута rematchAttemptsLimit. По истечению лимита попыток рематчинга агент запрашивает действие реакции на ошибку у юзера.
- frCancelTargetForm завершить матчинг формы
 При завершении операции с ошибкой процесс матчинга формы будет завершен и форма будет закрыта. Приложение при этом закрыто не будет.
- frTerminateApplication закрыть приложение
 При завершении операции с ошибкой приложение будет закрыто.

 frAskUser запросить у пользователя способ реакции на ошибку При завершении операции с ошибкой появляется запрос действия, которое необходимо выполнить в качестве реакции на возникшую ошибку. Доступность действий определяется состоянием приложения на момент появления запроса.

Опциональный атрибут. Значение по умолчанию: frAskUser.

<rematchAttemptsLimit>. Лимит попыток рематчинга формы. Определяет число повторных попыток матчинга формы. Значение атрибута имеет смысл только в том случае, если атрибут failureReaction имеет значение. В остальных случаях значение атрибута не учитывается.

<frRematchTargetForm>. После истечения лимита попыток рематчинга формы ESSO Агент запрашивает у пользователя способ реакции на ошибку.

Опциональный параметр. Значение по умолчанию: 0 (число повторных попыток не ограничено).

Пример описания операции входа:

<opPasswordChange failureReaction="frRematchTargetForm" rematchAttemptsLimit="3">

</opPasswordChange>

В приведенном примере допускается три попытки рематчинга формы смены пароля.

<forms>. Структура задает список целевых форм операции.

<formData>. Структура описывает целевую форму операции. Описание структуры приведено в разделе Формы приложения.

<resultSigns>. Структура задает данные для распознавания результата выполнения операции. Опциональный параметр. Значение по умолчанию: пусто (нет данных). В этом случае результат выполнения операции всегда считается успешным и данные операции сохраняются в базу данных SSO.

<formsOpOk>. Структура задает список форм-признаков успешного завершения операции. Описание структуры приведено в разделе "Формы приложения". Обязательный параметр (в рамках структуры <resultSigns>). Если в структуре

resultSigns формы-признаки успеха не описаны, данные операции не будут сохранены в базу данных SSO.

<formsOpFailed>. Структура задает список форм-признаков успешного завершения операции. Описание структуры приведено в разделе "Формы приложения". Обязательный параметр (в рамках структуры <resultSigns>).

<failureReactionAskingDetails>. Структура задает данные для окна ESSO Агента с запросом способа реакции на ошибку операции.

Атрибуты:

isDefaultOptionDisabled. Флаг запрета отображения действия, предусмотренного по умолчанию, для окна ESSO Агента с запросом способа реакции на ошибку операции. При включенном значении флага действие, предусмотренное по умолчанию для данной ситуации, выделяется в интерфейсе. В качестве действия по умолчанию установлены следующие реакции ESSO Агента:

- для приложений SSO-only (тип запуска приложения "Только Indeed-Id ESSO Агент") завершить приложение;
- о для остальных приложений игнорировать (ничего не делать).

Возможные значения:

- 0 флаг выключен (отображать действие по умолчанию)
- 1 флаг включен (не отображать действие по умолчанию)

Опциональный параметр. Значение по умолчанию: 0.

<disabledOptions>. Структура задает список запрещенных действий для окна ESSO Агента с запросом способа реакции на ошибку операции. Опциональный параметр. Значение по умолчанию: пусто (нет запрещенных опций).

<failureReaction>. Структура задает запрещенную опцию реакции на ошибку операции. Кнопка, предназначенная для выполнения соответствующего действия, в окне ESSO Агента с запросом способа реакции на ошибку операции будет неактивна.

Возможные значения:

- frRematchTargetForm запретить повторный рематчинг(повторный матчинг) формы.
- frCancelTargetForm запретить закрытие формы приложения.
- frTerminateApplication запретить завершение приложения. •
- frNone запретить игнорировать ошибку операции.

Ниже приведен пример описания операции.

Описание формы входа:

```
<opLogon failureReaction="frRematchTargetForm" rematchAttemptsLimit="2">
 <forms>
 </forms>
 <failureReactionAskingDetails isDefaultOptionDisabled="1">
   <disabledOptions>
     <failureReaction>frNone</failureReaction>
   </disabledOptions>
 </failureReactionAskingDetails>
</opLogon>
Описание операции смены пароля:
```

```
<opPasswordChange failureReaction="frRematchTargetForm" rematchAttemptsLimit="2">
 <forms>
```

</forms>

```
<failureReactionAskingDetails isDefaultOptionDisabled="1">
  <disabledOptions>
```

- <failureReaction>frNone</failureReaction>
- </disabledOptions>
- </failureReactionAskingDetails>
- </opPasswordChange>

Описание операции разблокировки:

```
<opUnlock failureReaction="frRematchTargetForm" rematchAttemptsLimit="2">
 <forms>
```

. . .

```
</forms>
```

```
<failureReactionAskingDetails isDefaultOptionDisabled="1">
```

```
<disabledOptions>
```

```
<failureReaction>frNone</failureReaction>
```

```
</disabledOptions>
```

```
</failureReactionAskingDetails>
```

```
</opUnlock>
```

Формы приложения

Формы в шаблонах служат для описания матчинга и алгоритма заполнения целевых окон приложения, таких как как окна входа, окна разблокировки, окна смены пароля. Обычные формы и формы-признаки успешности операции имеют идентичную структуру, поэтому при создании формы-признака можно

руководствоваться информацией из данного раздела. Каждая операция может содержать несколько форм для описания всех целевых окон для данной операции.

Синтаксис

Описание формы в шаблоне выглядит следующим образом:

<formData id="идентификатор формы" isForceRematching="флаг повторного матчинга" filledWndMarkDeletionTime="время удаления пометки "заполнено" isForReauthenticationOnly="флаг предназначения формы только для реаутентификации">

- <matching matchingType ="тип матчинга формы">
 - [данные для матчинга формы]
- </matching>
- <fillingAlgorithm>
- [данные алгоритма заполнения формы]
- </fillingAlgorithm>
- <cancellingAlgorithm>
- [данные алгоритма отмены формы]
- </cancellingAlgorithm>
- <targetPluginIds>

```
<pluginId value="значение идентификатора плагина 1"/>
```

```
...
```

```
<pluginId value="значение идентификатора плагина N"/>
```

</targetPluginIds>

</formData>

<formData>. Структура описывает форму приложения. Включает следующие атрибуты:

• id. Идентификатор формы (GUID). Идентификатор представляется в виде строки в формате: {00000000-0000-0000-0000-0000000000}}

Пример:

<formData id="{35CA5E88-3BB4-49E8-A453-E01F457653C6}" filledWndMarkDeletionTime="fwmdtOnFillingFinished"> ... </formData>

• **isForceRematching**. Флаг рематчинга (повторного матчинга) формы. Если для формы установлен флаг рематчинга, то после обработки этой формы (и снятия с нее пометки "заполнено"), ESSO Агент пытается повторно ее заматчить. Флаг следует устанавливать в том случае, если после заполнения формы, ESSO Агент не обнаруживает изменений в интерфейсе. Например, если сообщение об ошибке выводится в том же окне и после заполнения формы ESSO Агент не матчит сам форму как форму-признак ошибки.

Возможные значения:

- 0 рематчинг формы выключен
- 1 рематчинг формы включен

Опциональный параметр. Значение по умолчанию: 0.

 filledWndMarkDeletionTime. Время удаления пометки "заполнено" с окна целевой формы. После заполнения окно целевой формы помечается как "заполненное". До тех пор, пока отметка "заполнено" не будет снята, ESSO Агент не будет матчить окно как окно целевой формы. Соответственно, ESSO Агент не сможет начать выполнение новой операции, для которой это окно является целевой формой.

Возможные значения:

fwmdtOnFillingFinished – удалять пометку "заполнено" с окна целевой формы после завершения обработки заполненных данных (т.е. тогда, когда форма завершить обработку нажатия кнопки ОК).

fwmdtOnOpFinished – удалять пометку "заполнено" с окна целевой формы после завершения операции.

fwmdtNever – никогда не удалять пометку "заполнено" с окна целевой формы Опциональный параметр. Значение по умолчанию: *fwmdtNever*.

isForReauthenticationOnly. Флаг предназначения формы только для реаутентификации.

Если аутентификация при доступе в приложение выключена в настройках учетной записи приложения или пользователя, установка флага позволяет включить аутентификацию для отдельной формы приложения. Если флаг выключен, аутентификация для формы выполняется согласно настройкам приложения или пользователя.

Возможные значения:

0 – флаг выключен

1 – флаг включен (форма предназначена только для реаутентификации) Опциональный параметр. Значение по умолчанию: 0.

<matching>. Структура задает параметры матчинга формы. Описание структуры приведено в разделе Матчинг форм приложения. Включает следующие атрибуты:

• matchingType.Тип матчинга формы.

Возможные значения:

CFormData::mtWin32 – матчинг Win32–формы.

CFormData::mtWeb – матчинг Web-формы.

CFormData::mtDisplayedText – матчинг формы по способу отслеживания отображаемого текста. Используется для поддержки приложения Putty. В настоящее время для других приложений не используется.

CFormData::mtTextDataStream – матчинг формы по способу отслеживания потока текстовых данных. Используется для поддержки приложения Putty. В настоящее время для других приложений не используется.

CFormData::mtNative – матчинг формы по способу встроенного плагина. Используется для поддержки приложения Putty. В настоящее время для других приложений не используется.

<fillingAlgorithm>. Структура задает параметры матчинга формы. Описание структуры приведено в разделе Алгоритм заполнения.

<cancellingAlgorithm>. Структура задает алгоритм отмены формы. Если алгоритм отмены формы задан, то для его выполнения необходимо, чтобы в качестве реакции на ошибку операции была указана отмена формы (т.е. когда в описании операции тег *failureReaction* имеет значение *frCancelTargetForm*). Если алгоритм не задан, то при отмене формы она будет просто закрыта.

Отмена формы происходит в следующих случаях:

- пользователь отменил любой диалог SSO, связанный с выполнением операции, которой принадлежит данная форма;
- произошла ошибка при выполнении алгоритма заполнения формы.

Описание данной структуры аналогично описанию структуры **<fillingAlgorithm>**. Опциональный параметр. Значение по умолчанию: пусто (алгоритм отсутствует).

<targetPluginIds>. Структура задает множество идентификаторов плагинов, которым видна форма. Структура используется для поддержки приложения Putty. С Web- и Win32-шаблонами не применяется, с другими приложениями в настоящий момент не используется. Опциональный параметр. Значение по умолчанию: пусто.

<pluginId>. Структура задает описание идентификатора плагина. Структура используется для поддержки приложения Putty. С Web- и Win32-шаблонами не применяется, с другими приложениями в настоящий момент не используется. Включает в себя атрибут value, задающий значение

идентификатора плагина. Опциональный параметр. Отсутствие означает некорректность соответствующего описания идентификатора плагина.

Матчинг Win32 формы

Синтаксис

Матчинг формы win32 приложения выглядит следующим образом:

<matchingWin32 className="имя класса окна формы" componentId="ID компонента приложения" ignoreVisible="0 или 1" ignoreEnabled="0 или 1">

```
<caption isRegex="0- текст, 1 - regex">текст и regex заголовка формы</caption> <ctrls>
```

<ctrl>Данные контрола 1</ctrl>

<ctrl>Данные контрола C</ctrl>

</ctrls>

<size cx="ширина окна формы в пискелях" су="высота окна формы в пискелях"/> </matchingWin32>

Параметры

<matchingWin32>

Структура описывает матчинг win32 формы и содержит следующие атрибуты:

- className. Имя класса окна формы. Опциональный атрибут, по умолчанию не задан.
- componentId. ID компонента приложения, ссылка на соответствующий форме компонент, опциональный, по умолчанию не задан. Форма игнорируется при отсутствии компонента либо если его ID не указан.
- **ignoreVisible**. Флаг игнорирования невидимости формы при матчинге, по умолчанию 0. Возможные значения:
 - 1 Невидимая форма будет заматчена;
 - 0 Невидимая форма не будет заматчена
- **ignoreEnabled**. Флаг игнорирования состояния Disabled у формы при матчинге, по умолчанию 0. Возможные значения: 1 Отключенная форма будет заматчена;
 - 0 Отключенная форма не будет заматчена

<caption>. Структура задает заголовок окна формы. Является опциональной, по умолчанию не задана.

<ctris>. Структура задает описание параметров дочерних окон и контролов формы для матчинга.

<ctrl>. Структура задает параметры отдельного контрола. Содержание данной структуры см. в разделе **Настройки контролов формы целевого приложения** Win32 Мастера шаблонов.

Матчинг Web формы

Синтаксис

Матчинг формы web приложения выглядит следующим образом:

```
<matchingWeb componentId="ID компонента приложения" delay="период задержки матчинга в ms">
```

```
<title isRegex="0- текст, 1 - regex">текст и regex заголовка</title>
```

<formChangeScanSettings isScanOnFocusChange = "флаг поиска изменений при изменении фокуса" isSkip1stFocusChange = "флаг игнорирования 1-го события изменения фокуса на загруженной странице">

- <focusChangeEventIndexLimits>
- <min limit="минимальный предел индекса события изменения фокуса"/>
- <max limit="максимальный предел индекса события изменения фокуса"/>
- </focusChangeEventIndexLimits>
- </formChangeScanSettings>

<ctrls>

```
<webCtrl>Данные веб-контрола 1</webCtrl>
```

<webCtrl>Данные веб-контрола 2</webCtrl> </ctrls> </ matchingWeb>

Параметры

<Web>. Структура описывает матчинг web-формы. Включает следующие атрибуты:

- **componentId.** ID компонента приложения, ссылка на соответствующий форме компонент, опциональный, по умолчанию не задан. Матчинг формы по url игнорируется при отсутствии компонента либо если его ID не указан.
- delay период задержки матчинга в ms. Позволяет задать паузу перед началом матчинга.

<title>. Структура содержащая заголовок страницы формы, опциональная, по умолчанию не задана.

<formChangeScanSettings>. Структура, содержащая настройки поиска изменений страницы формы. По умолчанию не содержит настроек. Включает следующие атрибуты:

- **isScanOnFocusChange**. Флаг поиска изменений при изменении фокуса на странице формы. Опциональный атрибут. Значение по умолчанию 0.
- **isSkip1stFocusChange**. Флаг игнорирования первого события изменения фокуса на загруженной странице.

<focusChangeEventIndexLimits>. Структура, содержащая ограничения на индекс события изменения фокуса на странице, приводящего к повторному матчингу страницы на форму. Опциональный параметр. По умолчанию не нет ограничений (все события изменения фокуса приводят к повторному матчингу, если флаг isScanOnFocusChange выставлен).

<min> Структура, содержащая описание минимального предельного значения индекса события изменения фокуса, приводящего к повторному матчингу. Опциональный параметр. По умолчанию не задан (нет минимального предела). Включает следующие атрибуты:

• **limit.** Целое числовое значение минимального предельного значения индекса события изменения фокуса, приводящего к повторному матчингу. Опциональный атрибут. Значение по умолчанию 0.

max> – описание максимального предельного значения индекса события изменения фокуса, приводящего к повторному матчингу, опциональный, по умолчанию не задан (нет максимального предела). Включает атрибут limit. **limit.** – целое числовое значение максимального предельного значения индекса события изменения фокуса, приводящего к повторному матчингу. Опциональный атрибут. Значение по умолчанию 0

<ctrls> – параметры веб-контролов (html dom элементов) формы для матчинга. Подробно о одержании данной структуры см. в разделе Настройки контролов формы целевого приложения Web Мастера шаблонов.

Перехват и заполение окон Windows Security для Basic/Digest аутентификации в Internet Explorer

Окно **Безопасность Windows** (Windows Security, Рисунок 1) появляется при входе в web-приложение, если для доступа к нему настроена Basic/Digest проверка подлинности.

| Безопасность Wi | ndows | × |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|----|
| The server iiddemoserver.indeed-id.local is asking for your user name and password. The server reports that it is from iiddemoserver.indeed-id.local. | | |
| Warning: Your user name and password will be sent using basic authentication on a connection that isn't secure. | | |
| | Пользователь Пароль Запомнить учетные данные | |
| | ОК Отме | на |

Рисунок 1. – Окно Безопасность Windows.

В отличие от окна с запросом учетной записи для проверки подлинности Windows, для которого отрабатывает Credential Provider Indeed-Id, входящий в состав Indeed-Id Windows Logon, окно Basic/Digest аутентификации не позволяет использовать сторонний Credential Provider для аутентификации. В связи с этим перехват и заполнение данного окна должны быть организованы средствами Indeed-Id ESSO Агента.

Начиная с версии **5.4.21** в Агент SSO добавлен механизм, который позволяет корректно перехватывать и заполнять данный тип окон "Безопасность Windows".

Для того, чтобы Агент SSO корректно перехватывал и заполнял подобные окна необходимо в шаблоне добавить соответствующий Web-компонент с указанием URL целевой Web-страницы, а в разделе матчинга добавить тэг FormData, без описания матчинга элементов формы и без указания алгоритма заполнения:

```
...
```

```
<formData id="{F8DA6B54-9EB0-42cc-9CA4-AF78F7864F8A}"
filledWndMarkDeletionTime="fwmdtOnFillingFinished">
<matching matchingType="CFormData::mtWeb">
```

<matchingWeb componentId="Идентификатор компонента" />

</matching>

</formData>

...

Алгоритм заполнения

Алгоритм заполнения начинает работу после успешного матчинга формы. Все команды, описанные в алгоритме заполнения выполняются последовательно. Аалгоритм заполнения завершает работу после того, как все команды были выполнены или после возникновения ошибки в процессе выполнения текущей команды.

Синтаксис

```
<fillingAlgorithm>
<command type ="тип команды 1">
данные команды 1
</command>
...
<command type ="тип команды P">
данные команды P
</command>
</fillingAlgorithm>
```

Параметры

<fillingAlgorithm>. Структура, содержащая алгоритм заполнения формы.

<command>. Структура, содержащая описание отдельной команды. Включает в себя атрибут **type**. Подробнее в разделе **Команды в алгоритме заполнения**.

<description>. Описание команды. Не влияет на выполнение команды, отображается в сообщении об ошибке в команде. Описание необходимо задавать для лучшего понимания содержимого шаблона, при его редактировании и для отладки шаблона.

Команды в алгоритме заполнения

Типы команд:

ctControl

Тег команды <dataControl>

Команда эмулирует нажатие клавиш на клавиатуре, возможные значения: Атрибуты:

type. Тип управляющей команды. Возможные значения:

- cctTab эмуляция нажатия клавиши Tab
- cctShiftTab эмуляция нажатия сочетания клавиш Shift+Tab
- cctEnter эмуляция нажатия клавиши Enter
- cctEsc эмуляция нажатия клавиши Esc
- cctSpace эмуляция нажатия клавиши Пробел
- cctUp эмуляция нажатия клавиши Стрелка вверх
- cctDown эмуляция нажатия клавиши Стрелка вниз
- cctRight эмуляция нажатия клавиши Стрелка вправо
- cctLeft эмуляция нажатия клавиши Стрелка влево
- cctAltF4 эмуляция нажатия сочетания клавиш Alt+F4
- cctDel эмуляция нажатия клавиши Delete
- cctBackspace эмуляция нажатия клавиши Backspace
- cctCtrlA эмуляция нажатия сочетания клавиш Ctrl+A (выделить всё)

repetitionCount. Число повторений команды

waitIntervalMs. Интервал ожидания между повторениями команды. Имеет смысл указывать только если значение атрибута repetitionCount больше 1, т.к. он применяется для разделения повторений команды. Опциональный атрибут. Значение по умолчанию 0 (нет ожидания).

disableProcessingResultTracking. Флаг отмены слежения за результатом. Возможные значения:

1 – отменить слежение за результатом команды

0 – включить слежение за результатом команды. Значение по умолчанию – 0

ctFillCurrentCtrl

Тег команды <dataFillCurrentCtrl>

Команда выполняет подстановку данных в контрол в фокусе.

Параметры:

<fieldValueInfo> – структура, содержащая описание значения для заполнения поля формы

Атрибуты <fieldValueInfo>:

field – код значения для заполнения контрола, возможные значения:

- ffNoData нет данных. используется в командах нажатия на кнопку и установки фокуса на определённый контрол.
- ffUserName имя пользователя из учетной записи SSO.
- ffPassword пароль пользователя из учетной записи SSO
- ffNewPassword Новый пароль пользователя для учетной записи SSO. Используется в операции смены пароля для полльзователя.
- ffCustom Подставляет значение указанного кастомного поля.

customFieldId – ID кастомного поля, необходимо указывать только если значение атрибута field = ffCustom.

Атрибуты:

waitIntervalMs интервал ожидания меж эмуляцией нажатия отдельных символов, опциональный, по умолчанию 0 (нет ожидания)

fillingType – код алгоритма заполнения, возможные значения:

- cftSetText подстановка значения в контрол методом SetText
- cftSelectComboItem выбор значения в комбо-боксе.
- cftClickButton нажатие на контрол кнопки. Не требует данных для подстановки (**field=**ffNoData).
- cftEmulateInput подстановка значения в контрол методом эмуляции ввода
- cftSetFocus устанавливает фокус в указанный контрол.

clearCtrlText – флаг очистки контрола перед заполнением. возможные значения 1 – текст контрола будет очищен перед заполнением.

0 – к тексту контрола добавится текст заполнения

Работает только для кода алгоритма заполнения cftSetText.

ctFillCtrl

Тег команды <dataFillCtrl>

Команда выполняет подстановку данных в контрол, параметры которого указаны в команде

Атрибуты:

fillingType – код алгоритма заполнения, возможные значения:

- cftSetText подстановка значения в контрол методом SetText
- cftSelectComboItem выбор значения в комбо-боксе.
- cftClickButton нажатие на контрол кнопки. Не требует данных для подстановки (field=ffNoData).
- cftEmulateInput подстановка значения в контрол методом эмуляции ввода
- cftSetFocus устанавливает фокус в указанный контрол.

clearCtrlText – флаг очистки контрола перед заполнением. возможные значения 1 – текст контрола будет очищен перед заполнением.

0 – к тексту контрола добавится текст заполнения

Работает только для кода алгоритма заполнения cftSetText.

Параметры:

<**ctrl>** – данные целевого контрола.

<fieldValueInfo> – структура, содержащая описание значения для заполнения поля формы

Атрибуты <fieldValueInfo>

field – код значения для заполнения контрола, возможные значения:

- ffNoData нет данных. используется в командах нажатия на кнопку и установки фокуса на определённый контрол.
- ffUserName имя пользователя из учетной записи SSO.
- ffPassword пароль пользователя из учетной записи SSO

Indeed-Id ESSO Template Wizard Руководство по эксплуатации

- ffNewPassword Новый пароль пользователя для учетной записи SSO. Используется в операции смены пароля для полльзователя.
- ffCustom Подставляет значение указанного кастомного поля.

customFieldId – ID кастомного поля, необходимо указывать только если значение атрибута field = ffCustom.

ctWait

Тег команды <dataWait>

Команда выполняет ожидание, позволяющее создавать паузы между командами заполнениями

Атрибуты:

periodMs – атрибут задает период ожидания в миллисекундах.

ctPosition

Тег команды <dataPosition>

Команда выполняет перевод фокуса на контрол на форме, параметры которого указаны в команде.

Атрибуты:

waitIntervalMs – интервал ожидания между переходом к следующему контролу при во время поиска, по умолчанию 0 (нет ожидания).

disableProcessingResultTracking – флаг отмены слежения за результатом

возможные значения

1 – отменить слежение за результатом команды

0 – включить слежение за результатом команды.

Значение по умолчанию – 0

Параметры:

<ctrl> – данные целевого контрола.

ctFillWebCtrl

Тег команды <dataFillWebCtrl>

Команда выполняет заполнение веб-контрола, параметры которого указаны в команде

Атрибуты:

fillingType – код алгоритма заполнения, возможные значения для ctFillWebCtrl:

- cftSetText подстановка значения в контрол методом SetText
- cftSelectComboItem выбор значения в комбо-боксе.
- cftClickButton нажатие на контрол кнопки. Не требует данных для подстановки (**field=**ffNoData).

Параметры:

<webCtrl> – структура содержит данные целевого веб-контрола.

<fieldValueInfo> – структура, содержащая описание значения для заполнения поля формы

Атрибуты <fieldValueInfo>:

field – код значения для заполнения контрола, возможные значения:

- ffNoData нет данных. используется в командах нажатия на кнопку и установки фокуса на определённый контрол.
- ffUserName имя пользователя из учетной записи SSO.
- ffPassword пароль пользователя из учетной записи SSO
- ffNewPassword Новый пароль пользователя для учетной записи SSO. Используется в операции смены пароля для полльзователя.
- ffCustom Подставляет значение указанного кастомного поля.

customFieldId – ID кастомного поля, необходимо указывать только если значение атрибута field = ffCustom.

Ограничения генерируемых паролей

Шаблон может содержать параметры генерации паролей целевого приложения, такие как:

- длина пароля
- группы символов, допустимых для использования в пароле
- количество вхождений символов каждой группы

Синтаксис

<passwordRestrictions length="длина пароля" eachCharacterOccurenceMax="макс. количество вхождений">
<acceptableCharacters>

<characterGroup type="тип группы 1" customCharacters="кастомные символы группы 1"/>

<characterGroup type="тип группы N" customCharacters="кастомные символы группы N"/> </acceptableCharacters>

<characterOccurences>

<characterOccurence anyCharacterOccurenceMin="мин. количество вхождений из группы 1"> <characterGroup type="тип группы 1" customCharacters="кастомные символы группы 1"/> </characterOccurence>

...

```
<characterOccurence anyCharacterOccurenceMin="мин. количество вхождений из группы N">
<characterGroup type="тип группы N" customCharacters="кастомные символы группы N"/>
</characterOccurence>
```

</characterOccurences></passwordRestrictions>

Параметры

passwordRestrictions>. Структура задает ограничения и определяет требования к качеству генерируемых паролей для ESSO-приложений. Включает в себя следующие атрибуты:

- length. Длина генерируемого пароля в символах. Если также заданы требования вхождения символов (структура <characterOccurences>), то параметр определяет минимальную длину пароля. В этом случае при генерации пароля сначала выполняются требования вхождения символов. Если после выполнения требований вхождения длина сгенерированного пароля окажется больше заданной, пароль будет иметь длину, соответствующую выполненным требования вхождения (т.е. пароль не будет обрезан). Опциональный параметр. Значение по умолчанию: 0
- eachCharacterOccurenceMax. Максимальное количество вхождений каждого символа (независимо от того, в какую группу символов он входит) в сгенерированный пароль. Опциональный параметр. Значение по умолчанию: 0

Пример:

<passwordRestrictions length="8" eachCharacterOccurenceMax="2">

</passwordRestrictions>

Приведенное описание обозначает, что сгенерированный пароль в 8 символов будет содержать не более двух одинаковых символов (независимо от того, в какую группу символов они входят).

<acceptableCharacters>. Структура задает множество допустимых символов для генерации пароля.

Опциональный параметр. Значение по умолчанию: пусто. В этом случае при генерации пароля выполняются следующие условия:

пароль содержит символы из следующих групп:

- цифры (0...9);
- латинские строчные буквы (а...z);
- латинские прописные буквы (A...Z);
- специальные символы (`~!@#\$%^&*()_-+={}[]\|:;"'<>,.?/,пробел);

Количество вхождений символов в пароль определяется структурой <characterOccurences>.

<characterGroup>. Структура описывает допустимые группы символов, которые могут входить в состав пароля при генерации. Включает в себя следующие атрибуты:

- **type**. Тип группы допустимых символов. Возможные значения:
 - *cgtCustom* пользовательская группа символов (группа символов, определяемая пользователем)

cgtLatins – латинские буквы (А...Z, a...z)

cgtLatinsLower – латинские строчные буквы (а...z)

cgtLatinsUpper – латинские прописные буквы (А...Z)

cgtDigits – цифры (0...9)

cgtSpecial – неалфавитные символы (` ~ ! @ # \$ % ^ & * () _ - + = { } [] \ | : ; " '<> , . ? /, пробел и др.)

cgtAcceptableByDefault – латинские буквы (А...Z, а...z), цифры (0...9), специальные символы (` ~ ! @ # \$ % ^ & * () _ - + = { } [] \ | : ; " '<> , . ? /, пробел и др.)

Значение по умолчанию (если не задано ни одной группы допустимых символов): *cgtAcceptableByDefault.*

• **customCharacters**. Множество символов пользовательской группы. Опциональный параметр. Значение по умолчанию: пусто. Задание атрибута имеет смысл только для пользовательской группы символов, т.е. только в том случае, когда атрибут **type** имеет значение *cgtCustom*.

Пример: <characterGroup type="cgtCustom" customCharacters=",.<>/?[]{}=+-_\\|!@#\$%^&*() " />

Если тип группы отличается от **cgtCustom**, значение атрибута при обработке шаблона не учитывается.

Значение атрибута задается в виде строки, в которой перечислены все символы, входящие в пользовательскую группу, при этом необходимо соблюдать следующие правила перечисления пользовательских символов. Для задания специальных символов используются сущности, определенные в формате XML, а именно:

Символ Сущность

| < |
|---|
| |

> &qt;

& &

- " "
- ' '

Пример: <characterGroup type="cgtCustom" customCharacters="!@#\$%*() " />

Приведенное описание определяет пользовательскую группу со следующим множеством символов: ,.<>/?[]{}=+-_!@#\$%^&*() "

<characterOccurences>. Стуктура задает требования вхождения символов в пароль. Опциональный параметр. Значение по умолчанию: пусто. В этом случае при генерации пароля выполняются следующие условия:

- длина генерируемого пароля определяется значением атрибута **length** тега **<passwordRestrictions>**;
- если задано множество допустимых символов для генерации пароля (т.е. описана структура <acceptableCharacters>), пароль содержит символы из заданного множества, при этом количество вхождений символов из каждой группы определяется случайным образом.

<characterOccurence>. Структура задает требования вхождения в пароль символов заданной группы. Если для допустимой группы символов требования вхождения в пароль не определены, количество вхождений для символов этой группы определяется случайным образом с учетом заданных требований вхождений для других допустимых групп символов. Включает в себя атрибут аnyCharacterOccurenceMin, задающий минимальное число вхождений символов из заданной группы в генеририуемый пароль.

Пример:

```
<characterOccurences>
<characterOccurence anyCharacterOccurenceMin="3">
<characterOccurence anyCharacterOccurenceMin="3">
<characterOccurence="cgtLatinsLower" />
</characterOccurence>
```

</characterOccurences>

Приведенное описание обозначает, что сгенерированный пароль должен содержать не менее трех латинских строчных букв.

Параметры быстрого запуска

Синтаксис

```
<launcherDetails alias="Пользовательский псевдоним приложения">
<launcherApplications>
<applicationDetails alias="Пользовательский псевдоним компонента">
<crdLine>командная строка запуска приложения</crdLine>
<currentDir>текущий каталог </currentDir >
</applicationDetails>
</launcherApplications>
</launcherDetails>
```

Параметры

auncherdetails>. Параметры быстрого запуска. Должны быть заполнены, чтобы приложение появилось в окне быстрого запуска агента, опциональный. По умолчанию нет параметров быстрого запуска. Включают в себя атрибут alias – пользовательский псевдоним приложения. Если задан – замещает собой описание приложения в окне быстрого запуска.

<launcherApplications>. Структура задает контейнер для приложений, доступных для быстрого запуска. Включает атрибут alias – пользовательский псевдоним компонента приложения. Если задан – замещает собой командную строку в окне быстрого запуска.

<cmdline>. Структура содержит значение для командной строки для запуска процесса приложения.

<currentDir>. Структура задает текущий каталог для приложения. Опциональный параметр. Если не задан, будет использован каталог из пути, либо (если путь – неполный) – системный каталог.

Уникальные (кастомные) поля

В некоторых приложениях для авторизации, помимо логина и пароля, требуются дополнительные параметры (например, сервер, база данных и т.п.). Поля для задания дополнительных параметров авторизации в терминологии Indeed-Id называются **кастомными полями**.

Для обработки кастомного поля (обучение, заполнение поля и т.п.) в шаблоне приложения необходимо:

- описать кастомное поле;
- описать команду заполнения кастомного поля в алгоритме заполнения.

Синтаксис

<customFields>

<customField id="ID поля">

<field emptyStringValueForbidden="флаг пустого значения" editLengthSymbols="ширина поля" alternateLogin="флаг альтернативного логина">

<displayedName>имя поля</displayedName>

<value type="тип значения поля">

<str>значение поля</str>

</value>

</field>

</customField>

</customFields>

Indeed-Id ESSO Template Wizard

Руководство по эксплуатации

Параметры

<customFields>. Структура задает описание кастомных полей, которые могут использоваться для заполнения форм приложения. Опциональный параметр. Значение по умолчанию: пустой набор кастомных полей.

<customField>

Структура задает описание кастомного поля в виде набора пар <идентификатор поля; описание поля>. Включает атрибут **id** – идентификатор кастомного поля. Идентификатор используется при определении порядка кастомных полей для самообучения и при определении кастомного поля в алгоритме заполения. Значение: число, строка.

Идентификатор должен быть уникальным в пределах описания кастомных полей. В случае, если для нескольких полей заданы одинаковые идентификаторы, идентификатор будет присвоен тому полю, описание которого указано первым. Описания остальных полей с таким же индентификатором при обработке шаблона не учитываются.

<field>. Структура задает описание кастомного поля. Включает в себя следующие атрибуты:

• emptyStringValueForbidden. Флаг запрета пустых строковых значений поля.

Опциональный параметр. Значение по умолчанию: 0.

Возможные значения:

- 0 поле может иметь пустое значение;
- 1 поле не может иметь пустое значение.
- editLengthSymbols. Ширина поля для редактирования значения кастомного поля при самообучении. Атрибут определяет отображаемый размер поля для редактирования значения кастомного поля в окне самообучения и не определяет количество символов, которое можно задать в поле. Значение по умолчанию: 43 символа.

Значение атрибута имеет смысл только для кастомных полей с самообучением. Для других кастомных полей значение атрибута при обработке шаблона не учитывается.

 alternateLogin. Флаг использования значения кастомного поля в качестве альтернативного имени учетной записи SSO. Опциональный параметр. Значение по умолчанию 0. Возможные значения:

0 – значение поля не используется в качестве альтернативного имени учетной записи SSO;

1 – значение поля используется в качестве альтернативного имени учетной записи SSO.

В качестве альтернативного имени учетной записи SSO целесообразно использовать самообучаемое кастомное поле. Обучение кастомного поля происходит при обучении имени учетной записи SSO пользователем, при этом процесс обучения не отличается от стандартного обучения учетной записи.

Обученное значение кастомного поля при соответствующем описании алгоритма заполнения можно использовать для заполнения поля логина целевого приложения.

<displayedName>. Имя кастомного поля, отображаемое в ESSO Консоли в настройках учетной записи SSO. Значение: непустая строка.

<value>. Структура задает значение кастомного поля. Опциональный параметр. Значение по умолчанию: пусто. Включает в себя атрибут **type**, определяющий тип значения кастомного поля. Обязательный параметр. Возможное значение: *vtString*.

<str>. Строковое значение кастомного поля. Задание тега имеет смысл только в том случае, если для кастомного поля задан строковый тип (т.е. тогда, когда тег **type** имеет значение **vtString**). В остальных случаях значение тега при обработке шаблона не учитывается.

Заданное значение является значением по умолчанию и используется при добавлении новой записи SSO для пользователя. Для существующей записи пользователя текущее значение поля изменено не будет.

Если для поля задан строковый тип (тег **type** имеет значение **vtString**) при запрете пустых строковых значениях поля (атрибут **emptyStringValueForbidden** имеет значение 1), то значение тега **<str>** должно быть непустой строкой. Если тег **<str>** при указанных условиях имеет пустое значение, кастомное поле будет определено как самообучаемое.

Самообучаемые кастомные поля

Самообучаемым кастомным полем называется кастомное поле, значение которого задается пользователем при обучении учетной записи SSO для приложения. Для того, чтобы определить самообучаемое кастомное поле, необходимо:

- установить флаг запрета пустых строковых значений поля (т.е. установить значение атрибута emptyStringValueForbidden равным 1);
- удалить из описания значение поля (тег <value>)

Синтаксис

<customField id="ID поля">

<field emptyStringValueForbidden="*флаг пустого значения*" editLengthSymbols="*ширина поля*" alternateLogin="*флаг альтернативного логина*">

<displayedName>имя поля</displayedName>

</field>

</customField>

В ESSO Консоли самообучаемое кастомное поле отображается с соответствующим комментарием.

Порядок обучения кастомных полей

В некоторых случаях обучение кастомных полей необходимо выполнять в определенном порядке. Шаблоны ESSO позволяют задать этот порядок.

Синтаксис

<customFieldsSelfEnrollmentOrder>

<fieldId>идентификатор кастомного поля 1</fieldId>

<fieldId>идентификатор кастомного поля N</fieldId>

</customFieldsSelfEnrollmentOrder>

Параметры

<customFieldsSelfEnrollmentOrder>. Структура определяет порядок отображения кастомных полей в ESSO Консоли и порядок обучения кастомных полей.

Сначала отображаются и обучаются поля, для которых порядок явно указан, затем — остальные поля в порядке их перечисления в шаблоне.

Опциональный параметр. Если порядок обучения кастомных полей не задан, отображение в ESSO Консоли и обучение полей выполняется в порядке, соответствующем последовательности описания самообучаемых кастомных полей в структуре **<customFileds>**.

<fieldId>. Идентификатор кастомного поля. Кастомное поле с указанным идентификатором должно быть описано в структуре <customFileds>.

Сбор программных логов

Наличие программных логов Indeed-Id позволяет специалистам службы поддержки оперативно локализовать причины возможных проблемных ситуаций и принять меры к их устранению. Сбор программных логов осуществляется с помощью утилиты Indeed-Id GetLog, поставляемой в составе программного комплекса Indeed-Id Access Management. Для получения подробной информации обратитесь к документу Indeed-Id GetLog. Руководство пользователя.pdf.

Часто задаваемые вопросы

Ознакомиться со списком часто задаваемых вопросов и ответов на них можно в разделе **Технические** вопросы нашей базы знаний.