

AppChecker

Руководство администратора

Скачано с  техключи.рф

 **Эшелон**
комплексная безопасность

Содержание

1	О РУКОВОДСТВЕ	3
2	ОБЩАЯ ИНФОРМАЦИЯ	5
3	УСТАНОВКА И ЗАПУСК ПРОГРАММЫ	7
4	АДМИНИСТРИРОВАНИЕ	11
5	КОНСОЛЬНЫЙ КЛИЕНТ	27
6	ПОДДЕРЖКА	39
7	ГЛОССАРИЙ	41
8	О КОМПАНИИ	43
	ПРИЛОЖЕНИЕ	44

1 0 руководстве

Этот документ представляет собой Руководство Администратора AppChecker.

Руководство предназначено для следующих целей:

- ❖ помочь установить и запустить AppChecker;
- ❖ рассказать о ролях и группах пользователей;
- ❖ научить управлять сервером AppChecker;
- ❖ рассказать об интеграции с системами управления версиями.

Разделы документа

Этот документ содержит следующие разделы.

Общая информация ([стр. 4](#))

Раздел содержит описание функций AppChecker и перечень минимальных требований для компьютера, на который устанавливается AppChecker.

Установка и запуск программы ([стр. 6](#))

Раздел содержит инструкции по установке и запуску программы.

Администрирование ([стр. 10](#))

Раздел содержит информацию о работе с пользователями, об управлении сервером, об интеграции с системами управления версиями и о просмотре журналов работы.

Интеграция с системами Continuous Integration ([стр. 27](#))

В разделе содержатся сведения о добавлении функции проверки кода продуктом AppChecker в процесс сборки проектов в CI-системах Jenkins и Teamcity. Раздел рассчитан на пользователей уже знакомых с данными CI-системами.

Консольный клиент ([стр. 36](#))

Раздел содержит информацию о работе консольного клиента AppChecker.

Поддержка ([стр. 39](#))

Раздел содержит информацию о технической поддержке AppChecker.

Глоссарий ([стр. 41](#))

Раздел содержит определение терминов, встречающихся в документе.

О компании ([стр. 43](#))

Раздел содержит информацию о АО «НПО «Эшелон».

2 Общая информация

AppChecker – статический анализатор кода, предназначенный для поиска дефектов в исходном коде приложений, разработанных на C/C ++, C#, Java, PHP.

Функции

- ❖ поиск дефектов кодирования с помощью постоянно обновляемой базы правил и технологии анализа потоков данных;
- ❖ поиск программных закладок.

Преимущества

- ❖ поиск свыше 100 типов дефектов кодирования;
- ❖ постоянное пополнение базы правил поиска дефектов;
- ❖ поддержка классификации CWE;
- ❖ применение технологии анализа потока данных;
- ❖ возможность проводить совместный аудит кода несколькими экспертами;
- ❖ гибкая конфигурация анализируемых проектов.

Системные требования

Процессор	64-битный Intel-совместимый
Оперативная память	4 Гб
Жесткий диск (Свободно)	30 Гб (Рекомендуется 100 Гб)
Порты	USB порт для ключа безопасности
Поддерживаемые браузеры	Internet Explorer 10+
	Google Chrome 30+
	Mozilla Firefox 25+
Операционная система	Linux Ubuntu 14.04 Server LTS x64
	Microsoft Windows 7 SP1 x64
	Microsoft Windows 8 x64
	Microsoft Windows 10 x64
	Microsoft Windows Server 2008 x64
Разрешение экрана	1024x768 и выше
Среда виртуализации	VMWare
	Virtual Box
Дополнительное ПО	Software Platform .Net Framework 4.0 (только для ОС Windows)
	Java Runtime Environment 8 x64
	CodeMeter Runtime 4.5

3 Установка и запуск программы

AppChecker поставляется с аппаратным или программным ключом.

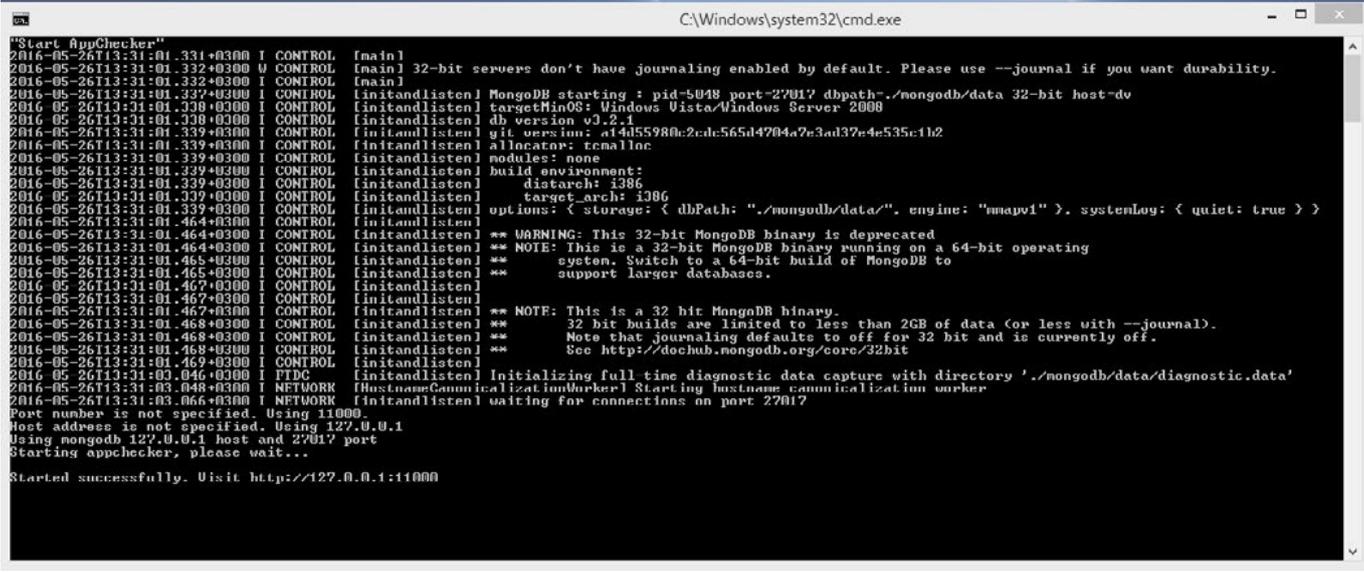
При наличии аппаратного ключа вставьте его в USB-порт перед установкой. При исполнении программного ключа активируйте его после установки (см. [приложение](#)).

3.1 Установка и запуск на ОС Windows

Установить (если не установлено) Software Platform .Net Framework 4.0

Запустите установочный файл appchecker_setup.exe. Следуйте дальнейшим инструкциям, чтобы установить AppChecker.

Запустите файл start.bat, который находится в директории установки.



```
C:\Windows\system32\cmd.exe

*Start AppChecker*
2016-05-26T13:31:01.331+0300 I CONTROL [main]
2016-05-26T13:31:01.332+0300 V CONTROL [main] 32-bit servers don't have journaling enabled by default. Please use --journal if you want durability.
2016-05-26T13:31:01.332+0300 I CONTROL [main]
2016-05-26T13:31:01.337+0300 I CONTROL [initandlisten] MongoDB starting : pid=5048 port=27017 dbpath=./mongodb/data 32-bit host=dv
2016-05-26T13:31:01.338+0300 I CONTROL [initandlisten] targetMinOS: Windows Vista/Windows Server 2008
2016-05-26T13:31:01.338+0300 I CONTROL [initandlisten] db version v3.2.1
2016-05-26T13:31:01.339+0300 I CONTROL [initandlisten] git version: a14d55980c2cd8565d4704a7e3ad37e4e535c1b2
2016-05-26T13:31:01.339+0300 I CONTROL [initandlisten] allocator: tcmalloc
2016-05-26T13:31:01.339+0300 I CONTROL [initandlisten] modules: none
2016-05-26T13:31:01.339+0300 I CONTROL [initandlisten] build environment:
2016-05-26T13:31:01.339+0300 I CONTROL [initandlisten] distarch: i386
2016-05-26T13:31:01.339+0300 I CONTROL [initandlisten] target_arch: i386
2016-05-26T13:31:01.339+0300 I CONTROL [initandlisten] options: { storage: { dbPath: './mongodb/data/', engine: 'mmapv1' }, systemLog: { quiet: true } }
2016-05-26T13:31:01.464+0300 I CONTROL [initandlisten]
2016-05-26T13:31:01.464+0300 I CONTROL [initandlisten] ** WARNING: This 32-bit MongoDB binary is deprecated
2016-05-26T13:31:01.464+0300 I CONTROL [initandlisten] ** NOTE: This is a 32-bit MongoDB binary running on a 64-bit operating
2016-05-26T13:31:01.465+0300 I CONTROL [initandlisten] ** system. Switch to a 64-bit build of MongoDB to
2016-05-26T13:31:01.465+0300 I CONTROL [initandlisten] ** support larger databases.
2016-05-26T13:31:01.467+0300 I CONTROL [initandlisten]
2016-05-26T13:31:01.467+0300 I CONTROL [initandlisten]
2016-05-26T13:31:01.467+0300 I CONTROL [initandlisten] ** NOTE: This is a 32 bit MongoDB binary.
2016-05-26T13:31:01.467+0300 I CONTROL [initandlisten] ** 32 bit builds are limited to less than 2GB of data (or less with --journal).
2016-05-26T13:31:01.468+0300 I CONTROL [initandlisten] ** Note that journaling defaults to off for 32 bit and is currently off.
2016-05-26T13:31:01.468+0300 I CONTROL [initandlisten] ** See http://dochub.mongodb.org/core/32bit
2016-05-26T13:31:01.468+0300 I CONTROL [initandlisten]
2016-05-26T13:31:01.469+0300 I CONTROL [initandlisten] Initializing full time diagnostic data capture with directory './mongodb/data/diagnostic.data'
2016-05-26T13:31:03.046+0300 I PIDG [HostnameCommunicationWorker] Starting hostname communication worker
2016-05-26T13:31:03.048+0300 I NETWORK [initandlisten]
2016-05-26T13:31:03.066+0300 I NETWORK [initandlisten] waiting for connections on port 27017
Port number is not specified. Using 11000.
Host address is not specified. Using 127.0.0.1
Using mongodb 127.0.0.1 host and 27017 port
Starting appchecker, please wait...
Started successfully. Visit http://127.0.0.1:11000
```

Сервер AppChecker будет запущен.

Введите в адресной строке браузера адрес веб-интерфейса AppChecker, указанный в консоли.

3.2 Установка и запуск на ОС Linux

Распакуйте архив с программой, используя следующую команду:

```
tar -zxvf appchecker.tgz -C имя_директории
```

Установите пакеты openjdk-8-jre или oracle-java8-installer, libqt4-core, clang-3.5, mono-runtime, mono-devel, mongodb-server и gsettings-desktop-schemas, используя команду:

```
sudo apt-get install имя_пакета
```

Загрузите codemeter с сайта разработчика. Для установки пакета следуйте инструкциям, приведенным на сайте.

Запустите файл start.sh, который находится в директории установки. Сервер AppChecker будет запущен.

Введите в адресной строке браузера адрес веб-интерфейса AppChecker, указанный в консоли.

```
echelon@appchecker:~$ ./start.sh
echelon@appchecker:~$ Port number is not specified. Using 11000.
Host address is not specified. Using 127.0.0.1
Using mongodb 127.0.0.1 host and 27017 port
Starting appchecker, please wait...

Started successfully. Visit http://localhost:11000
echelon@appchecker:~$
```

3.3 Установка и запуск в среде виртуализации

AppChecker может быть запущен в средах виртуализации VMWare и VitrualBox.

Импортируйте образ AppChecker формата OFV в среду виртуализации. **Запустите виртуальную машину** и выполните вход в гостевую ОС.

Логин — appchecker. Пароль — appchecker.

```
Ubuntu 14.04.2 LTS appchecker tty1
appchecker login: appchecker
Password: _
```

Для системы виртуализации VMware необходимо пробросить порт 11000 виртуальной машины, чтобы иметь доступ к AppChecker. Для этого:

1. Узнайте IP-адрес виртуальной машины с помощью команды «ifconfig».
2. Откройте Virtual Network Editor через пункт меню «Edit».
3. Выберите сеть типа «NAT» из предложенного списка и нажмите на кнопку «NAT Settings...».
4. Нажмите кнопку «Add» в разделе «Port Forwarding».
5. Введите значение «11000» в поле «Host port».
6. Введите IP-адрес виртуальной машины, полученные на шаге 1, в поле «Virtual machine IP address».
7. Введите значение «11000» в поле «Virtual machine port». Нажмите «Применить все изменения».

Для системы виртуализации Virtual Box дополнительная настройка не требуется.

В корне домашней директории расположен файл `start.sh` для запуска сервера AppChecker.

Введите команду `ls` для просмотра домашней директории.

```
appchecker@appchecker:~$ ls
appchecker start.sh stop.sh
appchecker@appchecker:~$ _
```

Запустите `start.sh` для запуска сервера AppChecker.

Войдите в браузер адрес веб-интерфейса AppChecker, указанный в КОНСОЛИ.

```
Ubuntu 14.04.4 LTS appchecker tty1

appchecker login: Port number is not specified. Using 11000.
Using server host 0.0.0.0
Using mongodb 127.0.0.1 host and 27017 port
Starting appchecker, please wait...

Started successfully. Visit http://0.0.0.0:11000
```

4 Администрирование

Введите логин и пароль для входа в AppChecker.

По умолчанию логин – admin, пароль – admin.

AppChecker поддерживает назначение ролей пользователям.

Администрирование доступно только пользователям с системными ролями.

Пользователь «admin» имеет роли «Системный администратор» и «Администратор проекта». Пользователя «admin» нельзя изменить или удалить.

За более подробной информацией по работе с пользователями и ролями обратитесь к пунктам [4.1](#), [4.2](#), [4.3](#), [4.4](#), настоящего руководства.

4.1 Пользователи

Для добавления, редактирования и удаления пользователей воспользуйтесь вкладкой «Пользователи».

Имя пользователя может содержать только латинские буквы, цифры, дефис, нижнее подчеркивание.

Пользователи

Добавить пользователя

Имя	Действие
admin	управление группами управление ролями сменить пароль удалить

4.2 Роли пользователей

AppChecker позволяет создавать роли и назначать их пользователям.

Роль – это набор разрешений, соответствующих тому, какие действия может выполнять пользователь. Роли делятся на системные и проектные. Различия между системными и проектными ролями описаны далее в настоящем руководстве.

Для добавления, редактирования и удаления ролей воспользуйтесь вкладкой «Управление ролями».

Системные роли

Добавить системную роль	
Название роли	Действие
Системный администратор	изменить удалить
Создание проектов	изменить удалить

Проектные роли

Добавить проектную роль	
Название роли	Действие
Администратор проекта	изменить удалить
Аудитор	изменить удалить
Наблюдатель	изменить удалить
Разработчик	изменить удалить

Для назначения роли пользователю нажмите «Управление ролями» во вкладке «Пользователи».

Права пользователя "user1"

Системные роли

У пользователя нет системных ролей

Добавить новую роль: [добавить](#)

Проектные роли

У пользователя нет проектных ролей

Добавить новую роль: для проекта: [добавить](#)

Системные роли

Системные роли содержат следующий набор разрешений: «Управление сервером», «Управление пользователями» и «Создание проекта».

Разрешение «Управление сервером» позволяет выключать сервер, создавать и редактировать шаблоны систем управления версиями (СУВ), а также просматривать журналы работы сервиса и ошибок сервиса. Подробнее про сервер AppChecker, про интеграцию с шаблонами СУВ и про журналы работы смотрите в пунктах [4.6](#) – [4.10](#) данного руководства.

Разрешение «Управление пользователями» позволяет добавлять, редактировать и удалять пользователей, их роли и группы.

Разрешение «Создание проекта» позволяет создавать новый проект в системе. Подробнее про добавление проекта смотрите пункт 4.5. данного руководства.

В AppChecker предустановлены две системные роли. Роль «Создание проектов» имеет разрешение «Создание проекта». Роль «Системный администратор» имеет максимальный набор разрешений. Ее нельзя изменить или удалить.

Добавить системную роль

Название роли

Разрешения

- Управление сервером
- Управление пользователями
- Создание проекта

Отменить

Сохранить

Проектные роли

Проектные роли одержат следующий набор разрешений: «Управление проектом», «Запуск проекта», «Работа с дефектами» и «Просмотр проекта». **Укажите проект** при назначении проектной роли. Так же можно создать проектную роль, которая распространяется на все проекты в системе.

Добавить проектную роль

Название роли

Разрешения Управление проектом
 Запуск проекта
 Работа с дефектами
 Просмотр проекта

Разрешение «Управление проектом» позволяет удалять проекты.

Разрешение «Запуск проекта» позволяет запускать проект на новое сканирование и загружать новые версии исходного кода.

Разрешение «Работа с дефектами» позволяет изменять статус дефекта.

Разрешение «Просмотр проекта» позволяет просматривать отчеты проекта.

Разрешения «Запуск проекта», «Работа с дефектами», и «Управление проектом» доступны только при выборе разрешения «Просмотр проекта».

Действия, доступные пользователям с проектными ролями, подробно описаны в Руководстве Пользователя.

В AppChecker предустановлены четыре проектные роли.

Роль «Наблюдатель» имеет разрешение «Просмотр проекта». Роль «Аудитор» имеет разрешение «Просмотр проекта» и «Работа с дефектами». Роль «Разработчик» имеет разрешения «Просмотр проекта», «Работа с дефектами» и «Запуск проекта». Роль «Администратор проекта» имеет максимальный набор разрешений. Ее нельзя изменить или удалить.

4.3 Группы пользователей

AppChecker позволяет объединять пользователей в группы. Роли пользователя складываются из собственных ролей. А также из ролей групп. В которых он состоит.

Для создания группы нажмите «Добавить группу» во вкладке «Группы пользователей».

Имя группы может содержать только латинские буквы, цифры, дефис, и нижнее подчеркивание

Для добавления пользователя в группу нажмите «Управление группами» во вкладке «Пользователи».

Группы пользователей

Добавить группу	
Имя	Действие
group_1	управление ролями удалить
group_2	управление ролями удалить

Для назначения роли группе нажмите «Управление ролями» во вкладке «Группы пользователей».

Права группы пользователей "prj1"

Системные роли

У группы нет системных ролей

Добавить новую роль: System administrator [добавить](#)

Проектные роли

У группы нет проектных ролей

Добавить новую роль: Projects administrator для проекта: Все проекты [добавить](#)

4.4 Пример работы с пользователями, ролями и группами

Предположим, в компании работают следующие сотрудники:

- ❖ Системный администратор;
- ❖ Администратор проектов;
- ❖ Два аудитора;
- ❖ Два разработчика.

Ведется работа над двумя проектами. Администратор проектов создает проекты и контролирует работу разработчиков. Аудиторы проверяют проекты. Первый разработчик работает над первым проектом, второй – над вторым. Рассмотрим действия системного администратора в данной ситуации.

Создайте пользователя «Project_Manager» для администратора проектов. Для этого во вкладке «Пользователи» нажмите «Добавить пользователя». Введите имя пользователя и пароль. Нажмите «ОК».

Добавление пользователя

Имя пользователя	<input type="text" value="Project_manager"/>
Пароль	<input type="password" value="...."/>
<input type="button" value="Отменить"/> <input type="button" value="Добавить"/>	

Назначьте пользователю «Project_Manager» роли. Для менеджера рекомендуется назначить роль «Администратор проекта», а также назначить системную роль «Создание проектов».

Чтобы назначить роль «Администратор проекта» во вкладке «Пользователи» нажмите «Управление ролями» напротив пользователя «Project_Manager». В разделе «Проектные роли» выберите роль «Администратор проекта» для всех проектов и нажмите добавить.

Чтобы назначить роль «Создание проектов» во вкладке «Пользователи» нажмите «Управление ролями» напротив пользователя «Project_Manager». В разделе «Системные роли» выберите роль «Создание проектов» и нажмите «Добавить».

Права пользователя "Project_manager"

Системные роли

Название роли	Действие
Создание проектов	удалить

Добавить новую роль: Системный администратор

Проектные роли

Название роли	Название проекта	Действие
Администратор проекта	Все проекты	удалить

Добавить новую роль: Разработчик для проекта: Все проекты

Создайте пользователей «Auditor_1» и «Auditor_2» для двух аудиторов. Создание пользователей «Auditor_1» и «Auditor_2» происходит аналогично созданию пользователя «Project_Manager».

Создайте группу «Audit» для двух аудиторов. Для этого во вкладке «Группы пользователей» нажмите «Добавить группу».

Группы пользователей

Группа успешно добавлена

Имя	Действие
Audit	<input type="button" value="управление ролями"/> <input type="button" value="удалить"/>

Добавьте пользователей в группу «Audit». В группы рекомендуется объединять пользователей, работающих над одним проектом или обладающих одинаковым набором разрешений. Для группы «Audit» рекомендуется добавить пользователей «Auditor_1» и «Auditor_2».

Чтобы добавить пользователей в группу «Audit» во вкладке «Пользователи» нажмите «Управление группами» напротив нужного пользователя, выберите группу «Audit» и нажмите «Включить».

Управление группами пользователя "Auditor_1"

Название группы	Действие
Audit	исключить из группы

Включить пользователя в группу:

Назначьте роли группе пользователей «Audit». При назначении роли группе пользователей роль присвоится каждому пользователю, состоящему в группе. Для группы аудиторов рекомендуется назначить роль «Аудитор» для всех проектов.

Чтобы назначить роль «Аудитор» во вкладке «Группы пользователей» нажмите «Управление ролями» напротив группы «Audit». В разделе «Проектные роли» выберите роль «Аудитор» для всех проектов и нажмите «Добавить».

Права группы пользователей "Audit"

Системные роли

У группы нет системных ролей

Добавить новую роль:

Проектные роли

У группы нет проектных ролей

Добавить новую роль: для проекта:

Создайте пользователей «Developer_1» и «Developer_2» для двух разработчиков. Создание пользователей «Developer_1» и «Developer_2» происходит аналогично созданию пользователей «Project_manager», «Auditor_1» и «Auditor_2».

Назначьте роли пользователям «Developer_1» и «Developer_2». Для пользователя «Developer_1» рекомендуется назначить роль «Разработчик» для первого проекта, а пользователю «Developer_2» — роль «Разработчик» для второго проекта.

Чтобы назначить роль «Разработчик» для конкретного проекта во вкладке «Пользователи» нажмите «Управление ролями» напротив нужного пользователя. В разделе «Проектные роли» выберите роль

«Разработчик», выберите нужный проект и нажмите «Добавить».

Пользователи

[Добавить пользователя](#)

Имя	Действие
admin	управление группами управление ролями сменить пароль удалить
Project_manager	управление группами управление ролями сменить пароль удалить
Developer_1	управление группами управление ролями сменить пароль удалить
Developer_2	управление группами управление ролями сменить пароль удалить
Auditor_1	управление группами управление ролями сменить пароль удалить
Auditor_2	управление группами управление ролями сменить пароль удалить

После действий, приведенных выше, администратор проекта имеет разрешения «Создание проекта», «Управление проектом», «Просмотр проекта», «Работа с дефектами» и «Запуск проекта».

Аудиторы объединены в группу и имеют разрешения «Просмотр проекта» и «Работа с дефектами».

Первый разработчик имеет разрешения «Просмотр проекта», «Работа с дефектами» и «Запуск проекта» для работы с первым проектом, второй разработчик имеет аналогичные разрешения для работы со вторым проектом.

Права пользователя "Developer_1"

Системные роли

У пользователя нет системных ролей

Добавить новую роль: [добавить](#)

Проектные роли

Название роли	Название проекта	Действие
Разработчик	Project1	удалить

Добавить новую роль: для проекта: [добавить](#)

4.5 Добавление проектов

Нажмите «Добавить проект» в верхнем левом углу. В появившемся окне введите имя проекта.

Имя проекта может содержать только латинские буквы, цифры, дефис, нижнее подчеркивание и точку

Нажмите «Добавить».

Добавление проекта

Название

Загрузите проект в AppChecker, нажав «Выберите файл». Загрузка проекта начнется автоматически. Дождитесь окончания загрузки.

Загрузка исходных текстов проекта "Project1"

Для C/C++ проектов требуется дополнительная конфигурация.
Перейти на страницу загрузки утилит и инструкции по конфигурированию.

Загрузите zip-архив с исходными текстами либо архив, полученный в результате конфигурирования проекта утилитами.

или переместите файл на форму

Для проектов, написанных на C/C++, требуется дополнительная конфигурация для улучшения качества анализа. За более подробной информацией нажмите «Перейти на страницу загрузки утилит и инструкции по конфигурированию».

Для удаления проекта **нажмите кнопку «Удалить проект»** в правом нижнем углу.

Запустите анализ проекта, нажав «Запустить анализ».

Статический анализ будет запущен.

Статический анализ проекта "Test1"

Выполнение статического анализа 

4.6 Управление сервером

AppChecker по умолчанию использует порт 11000 для запуска веб-сервера.

При необходимости настройки сервера можно изменить (см. пункт [4.7](#) данного руководства).

Работать с сервером можно через интерфейс или через консольное приложение.

Общую информацию о сервере можно посмотреть во вкладке «Управление сервером» в веб-интерфейсе. Так же в этой вкладке можно выключить сервер.

Для выключения сервера нажмите «Выключить сервер» в разделе «Управление сервером».

Для включения сервера запустите файл **start.bat** для ОС Windows и **start.sh** для ОС Linux и для виртуальной машины.

Управление сервером

Выключить сервер

Информация о сервере

Версия: 1.9.20
Операционная система: Linux (3.16.0-30-generic amd64)
Время старта сервера: 9 июня 2016 г., 15:39:41

4.7 Настройка сервера

Настройте сервер **AppChecker** через конфигурационный файл `default.ini`, расположенный в директории `config`.

Описание параметров

Группа параметров [analyzer] описывает опции java-машины при запуске анализатора.

Параметр	Описание	Значение по умолчанию
<code>javaopts</code>	дополнительные параметры для JVM	отсутствует

Группа параметров [https] описывает настройки https-протокола.

Параметр	Описание	Значение по умолчанию
<code>enabled</code>	дополнительные параметры для JVM	<code>false</code>
<code>keystoreFile</code>	абсолютный путь к файлу-ключу	отсутствует
<code>keystorePass</code>	пароль	отсутствует
<code>keystoreType</code>	тип файла-ключа	отсутствует
<code>keystoreProvider</code>	наименование провайдера	отсутствует
<code>keystoreAlias</code>	алиас сервера	отсутствует

Группа параметров [mongo] описывает настройки доступа к базе данных.

Параметр	Описание	Значение по умолчанию
<code>login</code>	логин. Может отсутствовать. В этом случае авторизация не используется.	отсутствует
<code>password</code>	пароль. Должен присутствовать, если есть логин.	отсутствует
<code>dbName</code>	имя базы данных	<code>appsec</code>
<code>server</code>	ip-адрес сервера	<code>127.0.0.1</code>
<code>port</code>	Порт сервера	<code>27017</code>

Группа параметров [rest] описывает параметры работы сервиса

Параметр	Описание	Значение по умолчанию
<code>host</code>	адрес сервиса	<code>127.0.0.1</code>
<code>port</code>	порт сервиса	<code>11000</code>

Пример файла

```
[rest]
host=127.0.0.1
port=11000

[analyzer]
javaopts=-Xmx2g

[https]
enabled=true
keystoreFile=/home/user/ssl/keystore.p12
keystorePass=Sec1ab1234
keystoreType=PKCS12
keystoreProvider=SunJSSE
keystoreAlias=tomcat

[mongo]
login=dbuser
password=12345
dbName=rest
server=127.0.0.1
port=27017
```

4.8 Подключение https

Купите или сгенерируйте ключи для использования https.

Укажите в конфигурационном файле группу параметров https.

Обратите внимание, что для keystoreFile необходимо указать абсолютный путь.

Перезапустите сервис.

4.9 Интеграция с системами управления версиями

AppChecker поддерживает загрузку исходных текстов с помощью СУВ.

По умолчанию установлены шаблоны СУВ git, svn и mercurial.

Администратор может добавлять, редактировать и удалять шаблоны.

Для работы с шаблонами воспользуйтесь вкладкой «Шаблоны команд СУВ».

Шаблоны команд СУВ

[Создать новый шаблон](#)[Протестировать шаблоны команд](#)

Команды успешно протестированы

Название шаблона	Шаблоны команд	Действие
mercurial	Команда для инициализации: hg clone {url} . Команда для обновления: hg pull && hg update Команда для проверки: hg version	изменить удалить
Команда доступна Показать результат выполнения команды для проверки		
git	Команда для инициализации: git clone {url} --branch {branch} . Команда для обновления: git pull origin {branch} Команда для проверки: git version	изменить удалить
Команда доступна Показать результат выполнения команды для проверки		
svn	Команда для инициализации: svn checkout {url} Команда для обновления: svn update Команда для проверки: svn --version	изменить удалить
Команда доступна Показать результат выполнения команды для проверки		

Для создания шаблона нажмите «Создать новый шаблон».

Для редактирования шаблона нажмите «Изменить».

Для удаления шаблона нажмите «Удалить».

Каждый шаблон системы управления версиями включает в себя три команды:

1) Команда для инициализации.

Данная команда выполняется единожды после создания проекта.

Примеры команды: *git clone*, *svn checkout*.

2) Команда для обновления исходных текстов.

Данная команда служит для получения новых версий исходных файлов. Примеры команды: *git pull*, *svn update*.

3) Команда для проверки доступности.

Данная команда проверяет доступность СУБ. Примеры команды: *hg version*, *svn -version*.

Команда для проверки доступности выполняется при загрузке системы, при изменении команды или с помощью функции «Протестировать шаблоны команд» во вкладке «Шаблоны команд СУБ».

Если команда выполнена успешно, то данный шаблон можно будет использовать при создании проекта.

Если команда выполнена неудачно, то данный шаблон будет считаться неактивным и не будет отображаться при выборе метода загрузки исходных файлов.

Команда для проверки необязательна — в случае ее отсутствия шаблон будет считаться активным.

Для команд инициализации и обновления исходных текстов можно **здать переменные**, значения которых пользователь может ввести при создании проекта. Примеры таких переменных — адрес сервера, название ветки и т.д. Переменная оборачивается фигурными скобками.

Пример команды: *git clone {url} --branch {branch}*.

В данном случае заданы две переменные - *url* и *branch*, их значение указывает пользователь при создании проекта.

4.10 Журнал работы

Для просмотра журнала обращений к серверу и журнала ошибок воспользуйтесь вкладкой «Журналы работы». Логи хранятся в директории data/logs.

Журнал обращений к серверу регистрирует входящие запросы.

Журнал обращений к сервису

```
Access log
192.168.2.20 [28/Jul/2006:10:27:10 0300] "GET /cgi bin/try/ HTTP/1.0" 200 3395
127.0.0.1 - - [28/Jul/2006:10:22:04 -0300] "GET / HTTP/1.0" 200 2216
```

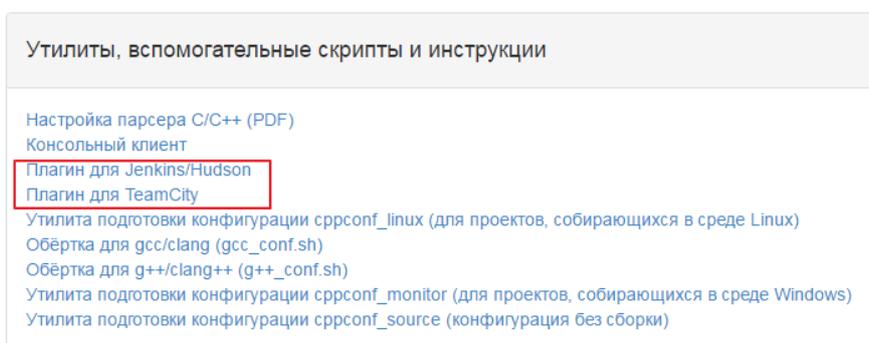
Журнал ошибок регистрирует сбои в работе системы.

Журнал ошибок

```
Error log
[Sat Aug 12 04:05:49 2006] [notice] SIGHUP received. Attempting to restart
```

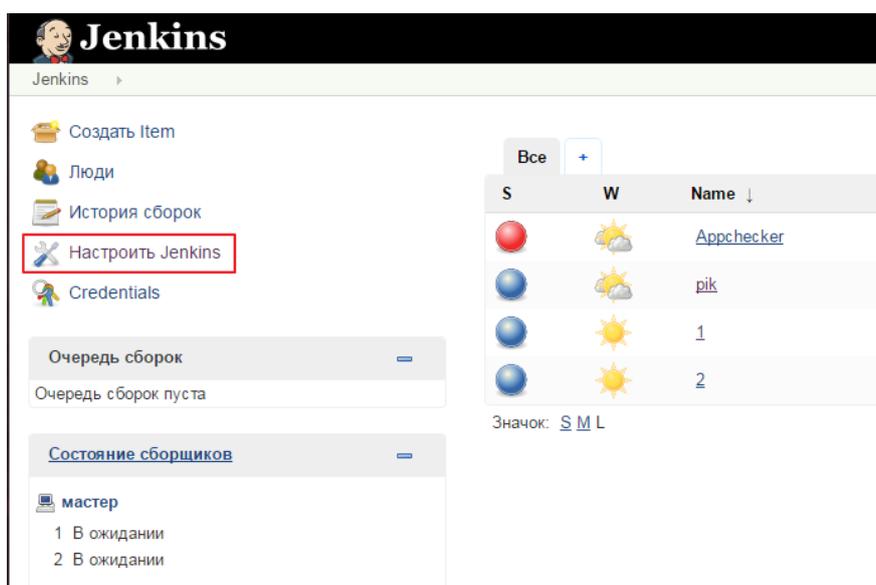
5 Интеграция с системами Continuous Integration

Скачайте установочные плагины для интеграции AppChecker с CI-системами Jenkins и Teamcity. Для этого **запустите AppChecker** и **зайдите на вкладку «О программе»**. Выберите одну из ссылок, выделенных на рисунке. Скачивание плагина начнется автоматически. При необходимости повторите действия со второй ссылкой.



5.1 Интеграция с CI Jenkins

Запустите CI Jenkins. В открывшемся окне браузера **выберите** пункт меню «**Настроить Jenkins**».



На новой странице **выберите «Управление плагинами»**

-  [Конфигурирование системы](#)
Конфигурировать глобальные настройки и пути.
-  [Configure Global Security](#)
Secure Jenkins; define who is allowed to access/use the system.
-  [Configure Credentials](#)
Configure the credential providers and types
-  [Перечитать настройки из файла](#)
Отменить все загруженные данные в памяти и перечитать все из файловой системы. Полезно в случае, если вы модифицировали файлы настроек вручную.
-  [Управление плагинами](#)
Добавить, удалить, выключить или включить плагины, расширяющие функциональные возможности Jenkins. **(Доступны обновления)**
-  [Системная информация](#)
Показывает всевозможную информацию о среде окружения для помощи в решении проблем.

В текущей вкладке браузера отобразится новая страница. На этой странице **перейдите на вкладку «Дополнительно»** и **найдите раздел «Загрузить плагин»**. **Нажмите** кнопку **«Выберите файл»**. Появится диалоговое окно открытия файла. **Найдите** ранее скаченный **плагин** и **загрузите** его на сайт, нажав на кнопку **«Загрузить»**.

Загрузить плагин

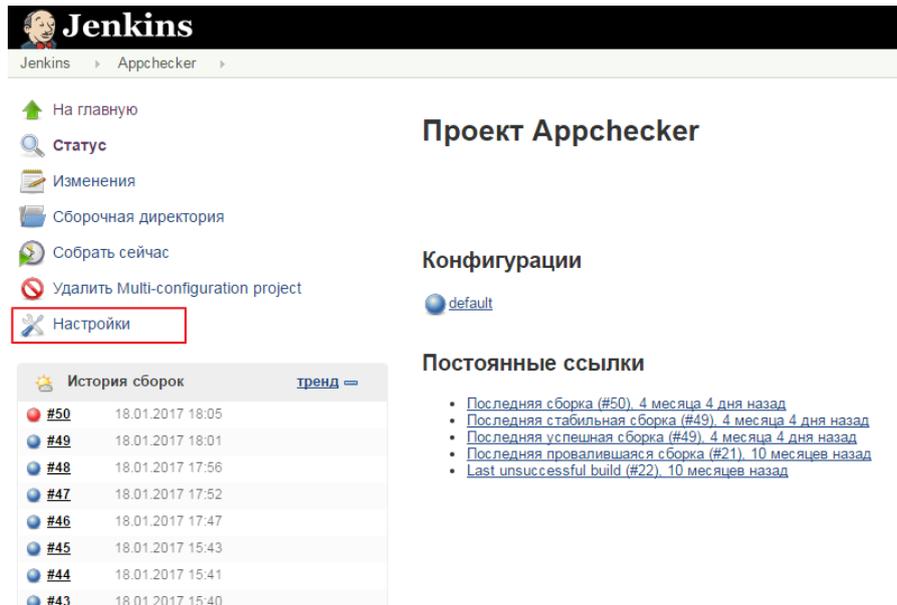
Здесь Вы можете установить плагин не из центрального репозитория, загрузив .hpi-файл

Файл: Файл не выбран

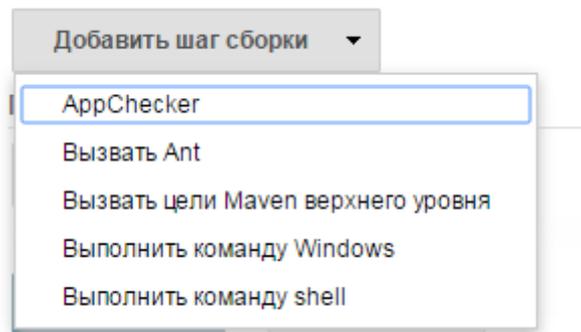
Перезапустите CI Jenkins в браузере с помощью команды

http://localhost:8080/jenkins/restart.

Откройте любой существующий **проект** или создайте новый.
Откройте страницу проекта. В меню проекта **щелкните** левой кнопкой мыши **по пункту «Настройки»**.



На открывшейся странице **найдите кнопку** выпадающего меню **«Добавить шаг сборки»**. **Щелкните на нее**. В появившемся списке **выберите пункт AppChecker**.



На странице проекта появятся поля для ввода настроек доступа к AppChecker.

AppChecker	
URL	<input type="text" value="127.0.0.1"/>
Port	<input type="text" value="11000"/>
Login	<input type="text" value="admin"/>
Password	<input type="password" value="....."/>
Project name	<input type="text" value="234"/>
Excluding extensions *	<input type="text"/>
Excluding directories *	<input type="text"/>
Source directory (empty for jenkins default)	<input type="text" value="/t/mc dist"/>

* - can contain multiple parameters, divided by comma

Заполните строки в соответствии со следующими требованиями:

- ❖ в поля URL и Port введите IP-адрес и номер TCP-порта веб-интерфейса, по которому доступен AppChecker;
- ❖ в поля Login и Password введите логин и пароль пользователя AppChecker;
- ❖ в поле Project name введите имя проекта. Данное имя будет отображаться в AppChecker как имя проекта;
- ❖ в поле Excluding extensions введите расширения файлов, которые будут исключены из проверки. Расширения указываются без масок ("doc", а не "*.doc").
- ❖ в поле Excluding directories введите пути к папкам, исключаемым из проверки, относительно папки с исходными текстами;
- ❖ поле Source directory заполняется в том случае, если в проекте существуют сценарии сборки, использующие исходные тексты из источников, не указанных в основных настройках Jenkins. Если такие сценарии присутствуют - введите путь к исходным текстам, в ином случае оставьте поле пустым.

Для настройки конфигурации проектов на C/ C++, нажмите на кнопку «Расширенные».

Configuration options
Make command

- none
- execute command with cppconf_linux.py
- execute cppconf_monitor
- execute only command

В строке Make command укажите команду сборки проекта. Ниже расположен переключатель выбора способов исполнения команды сборки:

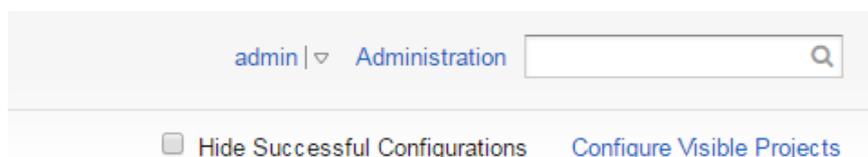
- ❖ не использовать;
- ❖ использовать с подключением утилит для среды Linux;
- ❖ использовать с подключением утилит для среды Windows;
- ❖ использовать с подключением утилит конфигурирования на основе исходного текста.

Выберите один из пунктов и нажмите кнопку внизу страницы:

- «Применить», чтобы сохранить настройки и остаться на странице;
- «Сохранить», чтобы сохранить настройки и покинуть страницу.

5.2 Интеграция с CI Teamcity

Запустите CI Teamcity. В открывшемся окне браузера в правом верхнем углу **выберите «Administration»**.



На открывшейся странице выберите пункт меню Plugins List, щелкните по нему левой кнопкой мыши.

Project-related Settings

- Projects **+ Create project** ▾
- Build Time
- Disk Usage
- Server Health
- Audit

You have 2 active projects with 1 build configuration. You can have a maximum of 20 build configurations (active and archived). ⓘ

Filter: Filter Show archived

⬆ ⬇

⌘ <Root project> Contains all other projects

▶ ⌘ qwe asd

User Management

- Users
- Groups

Integrations

- Tools
- NuGet Feed

Server Administration

- Global Settings
- Authentication
- Nodes Configuration
- Email Notifier
- Jabber Notifier
- Agent Cloud
- Diagnostics
- Backup
- Projects Import
- Licenses
- Clean-up Settings
- Usage Statistics
- Plugins List**

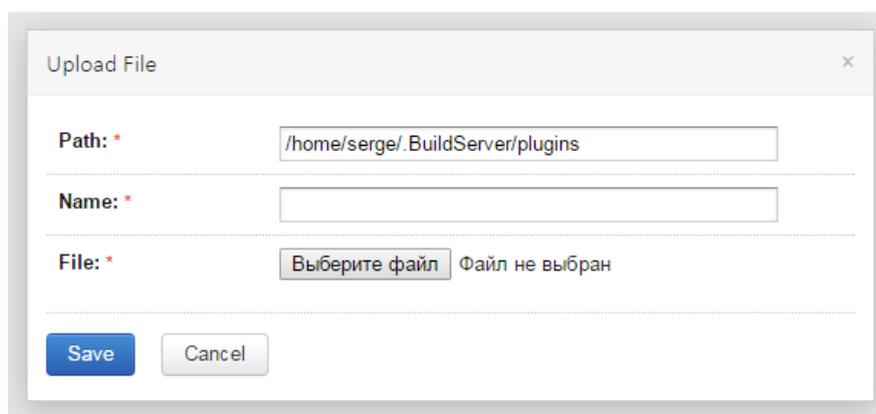
На новой странице нажмите Upload plugin zip.

This TeamCity installation has 95 plugins (including 1 external)
[Upload plugin zip](#) | [Available plugins](#)

External plugins

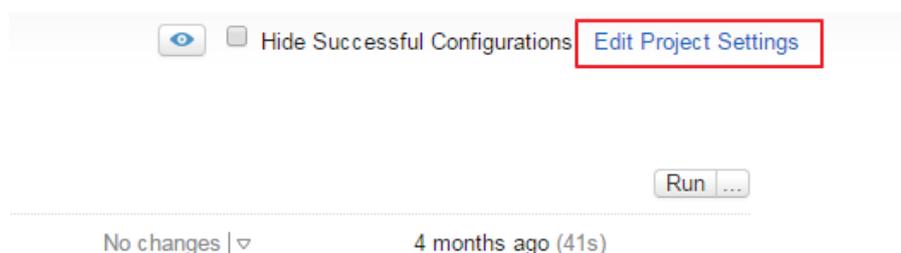
Plugin Name	Version
AppChecker launcher	0.1
Allows to check source codes by AppChecker	

В открывшемся диалоговом окне нажмите кнопку «Выберите файл». Появится диалоговое окно открытия файла. Найдите ранее загруженный плагин и загрузите его на сайт, нажав на кнопку “Save”.



Перезапустите агент сборки Teamcity.

Зайдите в любой существующий проект или создайте новый. Откройте страницу проекта. В меню проекта щелкните левой кнопкой мыши по пункту Edit Project Settings в правом верхнем углу экрана.



На открывшейся странице выберите конфигурацию из таблицы и щелкните по ней левой кнопкой мыши.

Build Configurations (19 left)

Build configurations define how to retrieve and build sources of a project. ⓘ

+ Create build configuration ▾

Name	Build Steps
Build1	Build AKVS, AppChecker plugin

На новой странице в меню слева выберите пункт Build Steps.

General Settings
Version Control Settings
Build Steps 2
Triggers
Failure Conditions
Build Features
Dependencies
Parameters
Agent Requirements

Нажмите на кнопку “Add build step” в центральной части экрана.

Build Steps

In this section you can configure the sequence of build step.

+ Add build step

Reorder build steps

Из выпадающего меню выберите AppChecker plugin.

New Build Step: | ▾

Runner type:

AppChecker plugin ▾

Когда установка параметра завершится, в текущей вкладке браузера отобразится страница с настройками AppChecker.

Build Step (2 of 2): AppChecker plugin | ▾

Make command using type

Runner type:

AppChecker plugin ▾

Allows to check source codes by AppChecker

Step name:

Optional, specify to distinguish this build step from other steps.

Execute step: 

If all previous steps finished successfully ▾

Specify the step execution policy.

Runner Parameters

AppChecker service address:



Service url or ip

Service port



Service port

Username:



AppChecker username

Password:



AppChecker user password

Project name:



Project name in AppChecker

Source folder (empty for default):



Source folder on build machine

Excluding extensions:



Extensions for excluding, divided by ";". Ex: "exe,class,zip"

Excluding directories:



Directory names for excluding, divided by ";". Ex: "bin" will exclude paths "/project/bin/", "/project/bin/lib/", "/project2/bin" etc.

Make command (if used):



Enter here make command if it is used

Make directory:



Directory, where make command will be executed

How to use make command:

execute command with cppconf_linux.py ▾

 Hide advanced options

Save

Cancel

Заполните строки в соответствии со следующими требованиями:

- ❖ в поля AppChecker service address и Service port введите IP-адрес и номер TCP-порта веб-интерфейса, по которому доступен AppChecker;
- ❖ в поля Username и Password введите логин и пароль пользователя AppChecker;

- ❖ в поле Project name введите имя проекта. Данное имя будет отображаться в AppChecker как имя проекта;
- ❖ в поле Excluding extensions введите расширения, которые будут исключены из проверки. Расширения указываются без масок ("doc", а не "*.doc").
- ❖ в поле Excluding directories введите пути к папкам, исключаемым из проверки, относительно папки с исходными текстами;
- ❖ поле Source folder заполняется в том случае, если в проекте существуют сценарии сборки, использующие исходные тексты из источников, не указанных в основных настройках Jenkins. Если такие сценарии присутствуют - введите путь к исходным текстам, в ином случае оставьте поле пустым.
- ❖ в строке Make command укажите команду сборки проекта;
- ❖ в строке Make directory укажите директорию, в которой будет выполняться команда сборки;
- ❖ В выпадающем меню How to use make command выберите способ исполнения команды сборки.

Нажмите "Save" для сохранения результата.

После интеграции проекты CI-систем, подключенные к AppChecker, появятся в среде AppChecker. О работе с проектами в среде AppChecker см. «Руководство оператора» главы 4 - 8.

6 Консольный клиент

Консольный клиент `appchecker-rest-client.jar` расположен в директории сервиса. Так же консольный клиент доступен для скачивания во вкладке «О программе».

Запустите консольный клиент, используя следующую команду:

```
java -jar appchecker-rest-client.jar [параметры]
```

Все доступные параметры приведены в конце данного раздела.

Запуск анализа проекта

Запустите анализ проекта, задав следующие параметры:

```
-i, --input <входная_директория>
```

Параметром могут быть директория, файл с расширением `tre` или `zip`-архив. Если в качестве параметра задана директория, то она будет упакована в `zip`-архив. Если в качестве параметра задан файл с расширением `tre`, то в `zip`-архив будут упакованы те файлы, на который он указывает.

```
-n, --name <имя_проекта>
```

Имя проекта указывается в кавычках. Имя проекта по умолчанию — `Project`.

```
-o, --output <выходная_директория>
```

Директория для отчетов о проведенном анализе. Пример запуска анализа проекта с именем Project1 приведен ниже.

```
java -jar appchecker-rest-client.jar -i C:\Projects\  
Project1\source_code.zip  
-n "Project1" -o C:\Projects\ Project1\Results
```

Удаление проекта

Удалите проект, указав следующие параметры:

```
-d, --delete
```

Удаление проекта.

```
-n, --name <имя_проекта>
```

Имя проекта указывается в кавычках. Имя проекта по умолчанию — Project.

Пример удаления проекта с именем Project1 приведен ниже.

```
java -jar appchecker-rest-client.jar -d -n "Project1"
```

Справка

Для получения справки введите следующий параметр:

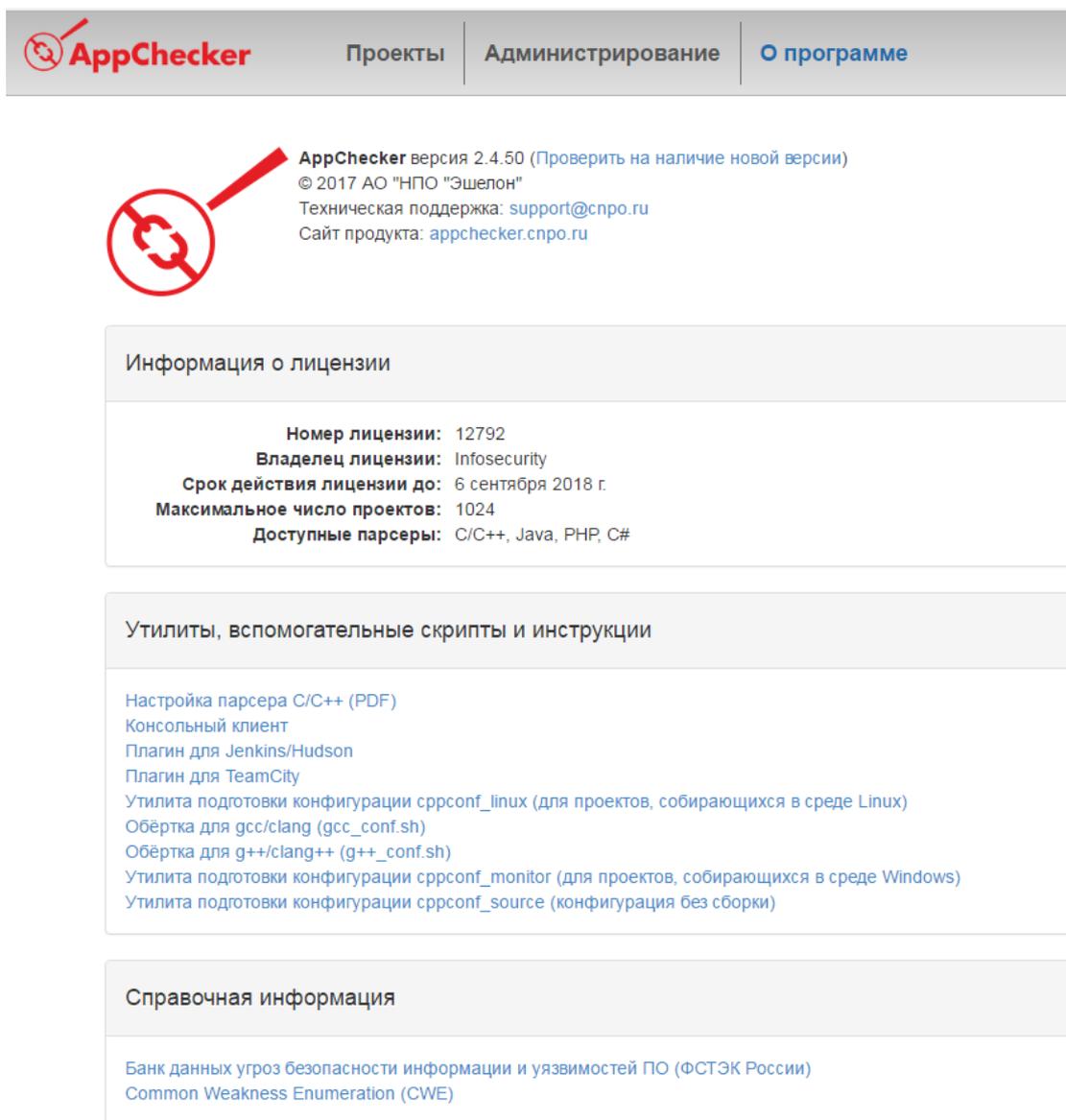
```
-h, --help
```

Все доступные параметры консольного клиента приведены в таблице ниже.

Параметр	Описание
-i,--input <входная_директория>	Входная директория.
-n,--name <имя_проекта>	Имя проекта. По умолчанию – Project.
-o,--output <выходная_директория>	Выходная директория.
-d,--delete	Удаление проекта.
-h,--help	Вызов справки.
-p,--port <порт_сервиса>	Порт сервиса. По умолчанию – 11000.
-s,--server <адрес_сервиса>	IP или доменный адрес сервиса. По умолчанию - http://127.0.0.1
-to,--timeout <время_таймаута>	Время таймаута в секундах. По умолчанию – 30.
-u,--user <имя_пользователя>	Имя пользователя. По умолчанию – admin.
-pw,--password <пароль>	Порт сервиса. По умолчанию – 11000.

7 Поддержка

Вкладка «О программе» содержит информацию о версии AppChecker, информацию о лицензии, ссылку для загрузки консольного клиента AppChecker, ссылки для загрузки утилит конфигурации и ссылку на банк данных угроз безопасности информации и уязвимостей ПО (ФСТЭК).



The screenshot shows the AppChecker web interface. At the top, there is a navigation bar with the AppChecker logo and three tabs: "Проекты", "Администрирование", and "О программе". The "О программе" tab is active. Below the navigation bar, there is a red magnifying glass icon over a broken link icon, pointing to the version information. The version information includes the current version (2.4.50), a link to check for updates, copyright information (© 2017 АО "НПО "Эшелон"), technical support email (support@cnpo.ru), and the product website (appchecker.cnpo.ru). Below this, there are three sections: "Информация о лицензии" (License Information), "Утилиты, вспомогательные скрипты и инструкции" (Utilities, scripts, and instructions), and "Справочная информация" (Reference information). The license section lists license number 12792, owner Infosecurity, expiration date September 6, 2018, maximum projects 1024, and supported parsers C/C++, Java, PHP, C#. The utilities section lists various tools and scripts for configuration and parsing. The reference information section lists the FSTEC Russia database and CWE.

AppChecker версия 2.4.50 ([Проверить на наличие новой версии](#))
© 2017 АО "НПО "Эшелон"
Техническая поддержка: support@cnpo.ru
Сайт продукта: appchecker.cnpo.ru

Информация о лицензии

Номер лицензии: 12792
Владелец лицензии: Infosecurity
Срок действия лицензии до: 6 сентября 2018 г.
Максимальное число проектов: 1024
Доступные парсеры: C/C++, Java, PHP, C#

Утилиты, вспомогательные скрипты и инструкции

- [Настройка парсера C/C++ \(PDF\)](#)
- [Консольный клиент](#)
- [Плагин для Jenkins/Hudson](#)
- [Плагин для TeamCity](#)
- [Утилита подготовки конфигурации srpconf_linux \(для проектов, собирающихся в среде Linux\)](#)
- [Обёртка для gcc/clang \(gcc_conf.sh\)](#)
- [Обёртка для g++/clang++ \(g++_conf.sh\)](#)
- [Утилита подготовки конфигурации srpconf_monitor \(для проектов, собирающихся в среде Windows\)](#)
- [Утилита подготовки конфигурации srpconf_source \(конфигурация без сборки\)](#)

Справочная информация

- [Банк данных угроз безопасности информации и уязвимостей ПО \(ФСТЭК России\)](#)
- [Common Weakness Enumeration \(CWE\)](#)

Для проверки актуальности версии нажмите на ссылку «Проверить на наличие новой версии».

Для получения технической поддержки отправьте электронное письмо на адрес support@cnpo.ru.

8 Глоссарий

Виртуальная машина

Модель вычислительной машины, созданной путем виртуализации вычислительных ресурсов: процессора, оперативной памяти, устройств хранения и ввода/вывода информации.

Гостевая ОС

ОС, устанавливаемая на виртуальную машину.

Дефект кода

Ошибка в исходном тексте программы, которая может привести к уязвимости.

Ключ программы

Ключ, который подтверждает право пользоваться AppChecker. Ключ бывает программным с расширением webb или аппаратным в виде usb-флешки.

Конфигурация проекта

Настройка проекта для повышения качества анализа.

Программная закладка

Скрытно внедренный в систему код, позволяющий злоумышленнику осуществить несанкционированный доступ.

Роль

Набор разрешений, соответствующий тому, какие действия может выполнять пользователь или группа пользователей.

Система управления версиями

Программное обеспечение, позволяющее хранить несколько версий одного документа.

Статический анализ

Анализ программного обеспечения, производимый без реального выполнения исследуемых программ.

CWE

Common Weakness Enumeration. Представляет собой список дефектов безопасности программного обеспечения. CWE охватывает более 15000 дефектов безопасности и включает в себя детальную структуру классификации из различных научных источников и примеров.

CWE – совместимость

Позволяет продукту быть зарегистрированным в качестве официального «CWE-совместимого». CWE-совместимый продукт должен представлять пользователям возможность поиска дефектов безопасности в CWE-идентификаторам и получения связанных с ними CWE-идентификаторов.

9 0 компании

АО «НПО «Эшелон» специализируется на комплексном обеспечении информационной безопасности.

Основными направлениями деятельности являются:

- ❖ проектирование, внедрение и сопровождение комплексных систем обеспечения информационной безопасности;
- ❖ сертификация, аттестация и лицензирование в области защиты информации;
- ❖ аудит информационной безопасности организаций;
- ❖ проектирование и аудит систем управления информационной безопасностью;
- ❖ разработка и поставка оборудования и средств защиты информации, в том числе в защищенном исполнении;
- ❖ обеспечение технической поддержки и сопровождение поставляемых решений, систем и продуктов;
- ❖ испытания, экспертизы, исследования в области безопасности информации.

Более детальную информацию о компании вы сможете найти на сайте npo-echelon.ru.

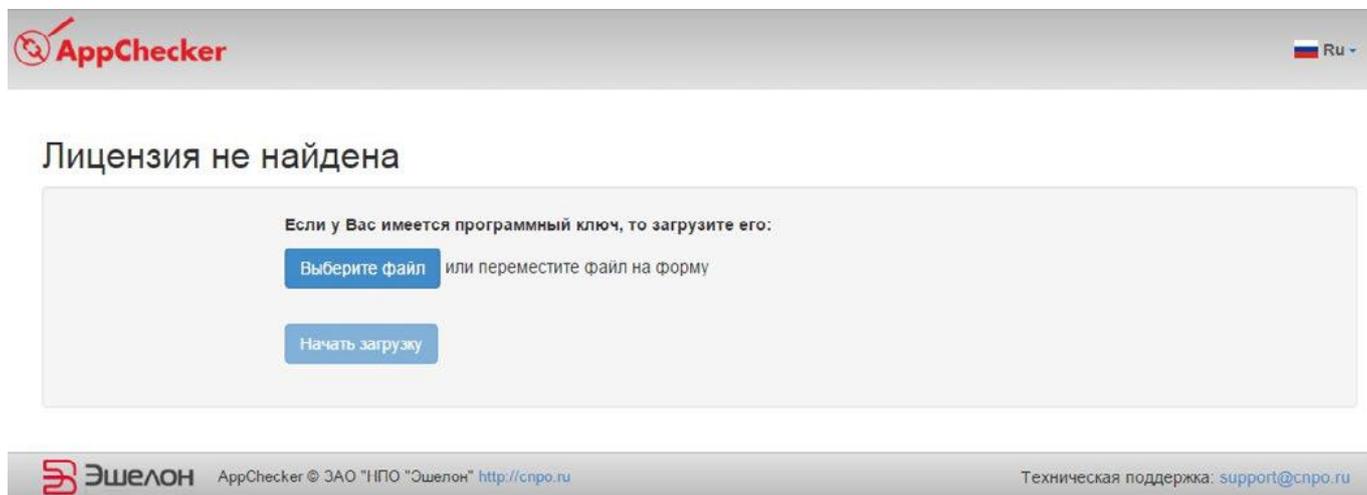
Приложение

Активация лицензии

Установите и запустите AppChecker. Более подробная информация содержится в главе 3 настоящего руководства.

Загрузите программный ключ с расширением wbb. После загрузки в веб-интерфейсе вам будет показан ключ инсталляции.

Отправьте ключ инсталляции на support.appc@npo-echelon.ru.



AppChecker Ru

Лицензия не найдена

Если у Вас имеется программный ключ, то загрузите его:

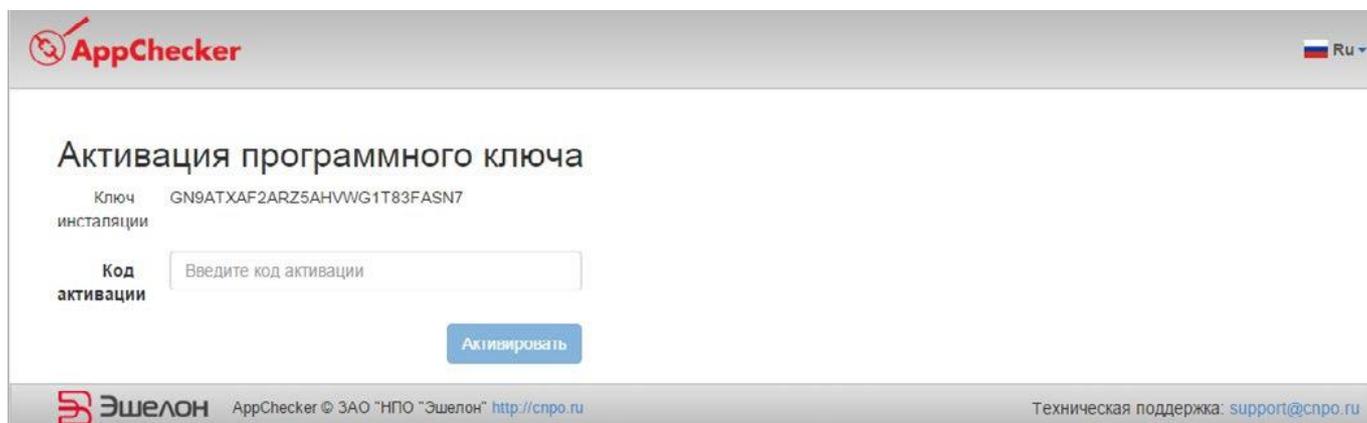
[Выберите файл](#) или переместите файл на форму

[Начать загрузку](#)

Эшелон AppChecker © ЗАО "НПО "Эшелон" <http://cnpo.ru> Техническая поддержка: support@cnpo.ru

В ответ вы получите код активации.

Введите код активации и нажмите «Активировать».



AppChecker Ru

Активация программного ключа

Ключ инсталляции: GN9ATXAF2ARZ5AHVWG1T83FASN7

Код активации:

[Активировать](#)

Эшелон AppChecker © ЗАО "НПО "Эшелон" <http://cnpo.ru> Техническая поддержка: support@cnpo.ru