

**AXIOM JDK**

# **Сервер приложений Libercat Руководство администратора**

Axiom Libercat | Май 2025

Copyright © 2019-2025 Все права защищены АО "АКСИОМ" (АКСИОМ)

Программное обеспечение АКСИОМ содержит программное обеспечение с открытым исходным кодом. Дополнительная информация о коде сторонних разработчиков доступна на сайте [https://axiomjdk.ru/third\\_party\\_licenses](https://axiomjdk.ru/third_party_licenses). Для дополнительной информации о том, как получить копию исходного кода, можно обратиться по адресу [info@axiomjdk.ru](mailto:info@axiomjdk.ru).

ДАННАЯ ИНФОРМАЦИЯ МОЖЕТ ИЗМЕНЯТЬСЯ БЕЗ ПРЕДВАРИТЕЛЬНОГО УВЕДОМЛЕНИЯ. АКСИОМ ПРЕДОСТАВЛЯЕТ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ "КАК ЕСТЬ" БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, АКСИОМ ПРЯМО ОТКАЗЫВАЕТСЯ ОТ ВСЕХ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ПОДРАЗУМЕВАЕМЫМИ ГАРАНТИЯМИ ТОВАРНОЙ ПРИГОДНОСТИ И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ.

АКСИОМ НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ЗА ЛЮБЫЕ КОСВЕННЫЕ, СЛУЧАЙНЫЕ, СПЕЦИАЛЬНЫЕ, ШТРАФНЫЕ ИЛИ КОСВЕННЫЕ УБЫТКИ, ИЛИ УБЫТКИ ОТ ПОТЕРИ ПРИБЫЛИ, ДОХОДА, ДАННЫХ ИЛИ ИСПОЛЬЗОВАНИЯ ДАННЫХ, ПОНЕСЕННЫЕ ВАМИ ИЛИ ЛЮБОЙ ТРЕТЬЕЙ СТОРОНОЙ, БУДЬ ТО В РЕЗУЛЬТАТЕ ДЕЙСТВИЯ ДОГОВОРА ИЛИ ДЕЛИКТА, ДАЖЕ ЕСЛИ АКСИОМ БЫЛО ПРЕДУПРЕЖДЕНО О ВОЗМОЖНОСТИ ТАКИХ УБЫТКОВ.

Использование любого программного продукта АКСИОМ регулируется соответствующим лицензионным соглашением, которое никоим образом не изменяется условиями данного уведомления. Программные продукты и фирменные наименования: Axiom JDK, Axiom JDK Pro, Axiom Runtime Container Pro, Axiom Linux, Libercat, Libercat Certified и АКСИОМ принадлежат АКСИОМ и их использование допускается только с разрешения правообладателя.

Товарный знак Linux® используется в соответствии с сублицензией от Linux Foundation, эксклюзивного лицензиата Линуса Торвальдса, владельца знака на всемирной основе. Java и OpenJDK являются товарными знаками или зарегистрированными товарными знаками компании Oracle и/или ее аффилированных лиц. Другие торговые марки являются собственностью их соответствующих владельцев и используются только в целях идентификации.

# Содержание

|                                                                        |    |
|------------------------------------------------------------------------|----|
| 1. Введение .....                                                      | 10 |
| 2. Установка Axiom Libercat .....                                      | 11 |
| Предварительные условия .....                                          | 11 |
| Microsoft Windows .....                                                | 11 |
| Установка Axiom Libercat с помощью MSI-инсталлятора .....              | 11 |
| Установка Axiom Libercat с помощью PowerShell .....                    | 12 |
| Настройка переменных окружения для Axiom Libercat в Windows .....      | 12 |
| Другие опциональные переменные .....                                   | 13 |
| Проверка корректности установки Axiom Libercat в Windows .....         | 13 |
| Деинсталляция Axiom Libercat в Windows .....                           | 14 |
| Linux .....                                                            | 14 |
| Ubuntu и другие ОС на базе .deb .....                                  | 14 |
| Red Hat Linux и другие ОС на основе .rpm .....                         | 14 |
| Ручная установка пакета RPM .....                                      | 14 |
| Alpine Linux .....                                                     | 15 |
| Установка отдельного пакета Axiom Libercat из архива в GNU/Linux ..... | 15 |
| Добавление пути к Axiom Libercat в переменные окружения .....          | 15 |
| Другие опциональные переменные .....                                   | 16 |

|                                                        |    |
|--------------------------------------------------------|----|
| Проверка корректности установки Axiom Libercat .....   | 17 |
| Деинсталляция Axiom Libercat .....                     | 17 |
| Особые случаи установки RPM и APK пакетов. ....        | 17 |
| Unix демон .....                                       | 19 |
| Проверка загруженных файлов .....                      | 20 |
| Windows .....                                          | 21 |
| Unix .....                                             | 21 |
| Устранение неполадок. ....                             | 21 |
| Проверка системных требований .....                    | 21 |
| Предупреждения от Microsoft Defender SmartScreen ..... | 21 |
| Загрузка Axiom Libercat с помощью браузера .....       | 23 |

### 3. Базовые настройки Axiom Libercat .....

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| Каталоги и файлы .....                                                | 24 |
| Переменные CATALINA_HOME и CATALINA_BASE .....                        | 24 |
| Преимущества использования CATALINA_BASE .....                        | 25 |
| Содержимое CATALINA_BASE .....                                        | 25 |
| Установка CATALINA_BASE .....                                         | 26 |
| Использование скрипта setenv для настройки переменных окружения ..... | 26 |
| Использование logj4 версии 2 .....                                    | 27 |

### 4. Файлы запуска и конфигурации .....

|                              |    |
|------------------------------|----|
| Скрипты запуска .....        | 28 |
| Конфигурационные файлы ..... | 34 |

## 5. Host Manager – Текстовый интерфейс .....

|                                                     |    |
|-----------------------------------------------------|----|
| Введение .....                                      | 36 |
| Конфигурирование доступа к приложению Manager ..... | 36 |
| Список команд .....                                 | 37 |
| Команда List .....                                  | 37 |
| Команда Add .....                                   | 38 |
| Команда Remove .....                                | 39 |
| Команда Start .....                                 | 39 |
| Команда Stop .....                                  | 39 |
| Команда Persist .....                               | 40 |

## 6. Руководство по приложению Manager .....

|                                                              |    |
|--------------------------------------------------------------|----|
| Введение .....                                               | 41 |
| Конфигурирование доступа к приложению Manager .....          | 42 |
| Интерфейс HTML .....                                         | 44 |
| Поддерживаемые команды диспетчера .....                      | 45 |
| Общие параметры .....                                        | 45 |
| Удаленное развертывание нового архива приложения (WAR) ..... | 46 |
| Развертывание нового приложения из локального пути .....     | 47 |

|                                                                     |    |
|---------------------------------------------------------------------|----|
| Развертывание ранее развернутого веб-приложения .....               | 48 |
| Развертывание каталога или WAR по URL-адресу .....                  | 48 |
| Развертывание каталога или WAR из базы приложений хоста .....       | 48 |
| Развертывание с использованием XML-файла конфигурации Context ..... | 49 |
| Примечания к развертыванию .....                                    | 49 |
| Ответ на развертывание .....                                        | 50 |
| Текущий перечень развернутых приложений .....                       | 51 |
| Перезагрузка существующего приложения .....                         | 51 |
| Перечень свойств ОС и JVM .....                                     | 52 |
| Перечень доступных глобальных ресурсов JNDI .....                   | 53 |
| Статистика сессии .....                                             | 54 |
| Истекшие сессии .....                                               | 54 |
| Запуск существующего приложения .....                               | 54 |
| Остановка существующего приложения .....                            | 55 |
| Отмена развертывания существующего приложения .....                 | 56 |
| Обнаружение утечек памяти .....                                     | 57 |
| Информация шифра SSL/TLS коннектора .....                           | 58 |
| Информация цепочки сертификатов SSL/TLS коннектора .....            | 58 |
| Информация доверенного сертификата SSL/TLS коннектора .....         | 59 |
| Перезагрузка конфигурации TLS .....                                 | 59 |
| Дамп потоков .....                                                  | 60 |
| Информация о виртуальной машине .....                               | 60 |

|                                                        |        |
|--------------------------------------------------------|--------|
| Сохранение конфигурации .....                          | 61     |
| Статус сервера .....                                   | 61     |
| Использование прокси-сервлета JMX .....                | 63     |
| Что такое прокси-сервлет JMX. ....                     | 63     |
| Команда JMX Query .....                                | 63     |
| Команда JMX Get .....                                  | 64     |
| Команда JMX Set .....                                  | 64     |
| Команда JMX Invoke .....                               | 65     |
| Выполнение команд диспетчера с помощью Ant .....       | 65     |
| Отражение результатов выполнения задачи .....          | 67     |
| <br>7. Обновление .....                                | <br>73 |
| Обновление с помощью программ менеджеров пакетов ..... | 73     |
| Обновление в ручном режиме .....                       | 74     |
| <br>8. Развертывание J2EE приложения .....             | <br>76 |
| Введение .....                                         | 76     |
| Варианты упаковки приложения .....                     | 76     |
| Развёртывание с помощью maven плагина .....            | 77     |
| <br>9. Развертывание веб-приложения .....              | <br>79 |

|                                                              |    |
|--------------------------------------------------------------|----|
| Введение .....                                               | 79 |
| Установка .....                                              | 79 |
| Несколько слов о Контексте .....                             | 80 |
| Развертывание при запуске Axiom Libercat .....               | 80 |
| Развертывание на действующем сервере Axiom Libercat .....    | 81 |
| Развертывание с использованием Axiom Libercat Manager .....  | 82 |
| Развертывание с использованием пакета Deployer клиента ..... | 82 |

## 10. Конфигурация менеджера транзакций..... 84

|                                            |    |
|--------------------------------------------|----|
| Введение .....                             | 84 |
| Конфигурация .....                         | 84 |
| Восстановление транзакций .....            | 85 |
| Установка уникального идентификатора ..... | 88 |

## 11. Как работает lookup..... 90

|                                                             |    |
|-------------------------------------------------------------|----|
| Клиент по умолчанию .....                                   | 90 |
| OpenEJBInitialContextFactory .....                          | 90 |
| RemoteInitialContextFactory .....                           | 90 |
| Безопасность .....                                          | 91 |
| Конфигурирование JNDI провайдера .....                      | 91 |
| Конфигурирование JNDI провайдера на уровне сервера .....    | 92 |
| Конфигурирование JNDI провайдера на уровне EJB модуля ..... | 92 |



|                                                                                                         |    |
|---------------------------------------------------------------------------------------------------------|----|
| Формат JNDI имен бинов .....                                                                            | 93 |
| Переопределение формата JNDI имен .....                                                                 | 96 |
| Переопределение через системные свойства .....                                                          | 97 |
| Переопределение через свойства файла дескриптора <code>openejb-jar.xml</code> .....                     | 97 |
| Переопределение с помощью тэга <code>jndi</code> в файле дескриптора <code>openejb-jar.xml</code> ..... | 97 |

# 1. Введение

Axiom Libercat является сервером приложений, исполняющим программы написанные на языке программирования Java в соответствии с набором спецификаций Servlets и JSP из Java EE. Программное изделие поставляется в четырех комплектах поставки (версиях):

- Сервер приложений Libercat версии 9, 10;
- Набор библиотек Axiom Libercat Embedded версии 9, 10;

Axiom Libercat реализует спецификации Servlet и Java Server Pages из набора спецификаций Servlets, JSP из Java EE. Комплект поставки Axiom Libercat Embedded реализует спецификации Servlet, Java Server Pages и Websocket из набора спецификаций Servlets, JSP из Java EE 8, используется как библиотека для приложения пользователя.

Вы можете загрузить Axiom Libercat по ссылке на [портале поддержки](#). Доступ к этому portalу предоставляется клиентам с активным договором поддержки. После входа на портал поддержки, если у вас есть несколько активных проектов, выберите нужный проект из выпадающего списка **"Выберите проект"** вверху страницы под именем пользователя. В зависимости от выбранного проекта вам будет доступна прямая ссылка на запрошенную вами версию продукта в разделе "Технологии". После завершения загрузки вы можете [проверить](#) загруженный файл, сравнив его размер на вашем диске с размером на странице загрузок.

Вы также можете использовать командную строку для загрузки необходимых пакетов установки, как описано далее в этом документе.

## 2. Установка Axiom Libercat

### Предварительные условия

Поскольку Изделие Axiom Libercat является приложением Java, перед его установкой нужно убедиться, что у вас установлена Среда разработки и исполнения Axiom JRE или Axiom JDK версии 8 или выше. Для установки пакета Axiom JRE или Axiom JDK выполните следующие действия:

1. Скачайте файл инсталлятора (по ссылке на [портале поддержки](#)) или найдите его на поставляемом компакт-диске.
2. Найдите и дважды кликните по файлу.
3. Следуйте инструкциям на экране окна инсталлятора.

Для подробной информации по установке и настройке пакетов Axiom JRE и Axiom JDK смотрите *Руководство администратора изделия Программное обеспечение «Axiom JDK»*.

### Microsoft Windows

#### Установка Axiom Libercat с помощью MSI-инсталлятора

Для установки Axiom Libercat с помощью файла инсталлятора выполните следующие действия:

1. Скачайте exe/msi-файл инсталлятора (по ссылке на [портале поддержки](#)) или найдите его на поставляемом компакт-диске.
2. Найдите и дважды кликните по файлу.
3. Следуйте инструкциям на экране окна инсталлятора. Его интерфейс и функциональность аналогичны другим мастерам установки и содержит следующие шаги:
  - Настройка каталога для установки
  - Установка Axiom Libercat как службы Windows (включено по умолчанию)

См. главу Руководство по службе Windows в документе «Руководство по эксплуатации» для получения информации по тому, как работать с Axiom Libercat в качестве службы Windows.

## Установка Axiom Libercat с помощью PowerShell

В случае, если нет возможности установить Axiom Libercat через графический интерфейс, существует способ установки посредством PowerShell:

1. Скачайте архив zip (по ссылке на [портале поддержки](#)) или найдите его на поставляемом компакт-диске.
2. Выполните следующую команду:

```
Expand-Archive [Сервер приложений Axiom Libercat].zip -DestinationPath
```

Данная команда распакует Сервер приложений Axiom Libercat в текущий каталог.

## Настройка переменных окружения для Axiom Libercat в Windows

Axiom Libercat является приложением Java и не использует переменные окружения напрямую. Переменные окружения используются сценариями (скриптами) запуска Axiom Libercat. Скрипты используют переменные окружения для подготовки команды, запускающей Axiom Libercat.

Необходимо добавить и настроить следующие переменные:

- CATALINA\_HOME (обязательная) – Данная переменная должна указывать на местоположение корневого каталога бинарного дистрибутива Axiom Libercat.
- CATALINA\_BASE (рекомендуемая) – Данная переменная указывает на расположение каталога «активной конфигурации» Axiom Libercat. По умолчанию используется путь указанный в CATALINA\_HOME.
- JRE\_HOME or JAVA\_HOME (обязательные и обычно устанавливаются автоматически при установке Java) – Данные переменные указывают на расположение каталога установки Java.

Мы рекомендуем использовать разные значения для переменных CATALINA\_HOME и CATALINA\_BASE для упрощения дальнейших обновлений и обслуживания, а также для использования нескольких экземпляров Axiom Libercat.

Чтобы добавить необходимые пути Сервера приложений Axiom Libercat в переменные окружения Windows выполните следующие операции:

1. В поисковой строке меню «Пуск» введите env и выберите пункт **Edit the system environment variables**.
2. Кликните по кнопке "Environment Variables".
3. В нижней части окна под секцией "System Variables", найдите и выделите строку со значением

Path в первом столбце. Затем нажмите кнопку "Edit".

4. В окне "Edit environment variable" нажмите кнопку "New" и введите требуемый путь. В данном окне можно также отредактировать или упорядочить пути.
5. В появившихся диалоговых окнах подтверждения операции нажмите "OK".

После применения изменений в настройке переменных окружения, возможно, потребуется перезапуск приложений. Рекомендуется перезагрузить всю операционную систему, чтобы гарантировать, что все приложения запустились с новым значением переменной окружения PATH.

Для проверки текущего значения переменной окружения PATH следует выполнить следующую команду в PowerShell:

```
$env:PATH
```

## Другие опциональные переменные

Помимо четырех, описанных выше, существуют и другие переменные окружения. См. описание каждой из них в комментариях в верхней части сценариев catalina.bat или catalina.sh.

Одной из часто используемых переменных является CATALINA\_OPTS. Она позволяет уточнить дополнительные параметры для команды java, запускающей Axiom Libercat.

Аналогичная переменная — JAVA\_OPTS. Он используется реже и позволяет установить настройки для запуска и остановки Axiom Libercat, а также для других команд.



### Примечание:

Не используйте JAVA\_OPTS для указания предельного размера используемой памяти. Не нужно выделять много памяти для небольшого процесса, который используется для остановки Axiom Libercat. Эти настройки содержатся в CATALINA\_OPTS.

## Проверка корректности установки Axiom Libercat в Windows

Для проверки корректности установки изделия Axiom Libercat, откройте следующий адрес в адресной строке браузера.

```
http://localhost:8080
```

Вы должны увидеть экран приветствия Axiom Libercat.

## Деинсталляция Axiom Libercat в Windows

Для деинсталляции Axiom Libercat удалите продукт через стандартный механизм деинсталляции Windows. Если использовалась установка из zip, удалите директорию её размещения и измените значение связанных переменных окружения.

## Linux

### Ubuntu и другие ОС на базе .deb

1. Для установки Axiom Libercat, загрузите пакет `.deb` по ссылке на [портале поддержки](#) или найдите его на поставляемом компакт-диске.
2. Установите пакет с помощью инструмента `apt`.

```
sudo apt install [Сервер приложений Axiom Libercat].deb
```

### Red Hat Linux и другие ОС на основе .rpm

1. Для установки Axiom Libercat, загрузите пакет `.rpm` по ссылке на [портале поддержки](#) или найдите его на поставляемом компакт-диске.
2. Установите пакет с помощью инструмента `yum`.

```
sudo yum install [Сервер приложений Axiom Libercat].rpm
```

### Ручная установка пакета RPM

1. Получите официальный GPG-ключ Axiom, а затем импортируйте его, как показано в следующем примере. Добавление ключа GPG требуется только один раз на компьютере.

```
wget -q -O - https://download.axiomjdk.ru/pki/GPG-KEY-axiom  
sudo rpm --import GPG-KEY-axiom
```

2. Загрузите пакет по ссылке на [портале поддержки](#) или найдите его на поставляемом компакт-диске. Установите пакет с помощью следующей команды.

```
sudo rpm --install [Сервер приложений Axiom Libercat].rpm
```

## Alpine Linux

Чтобы установить Axiom Libercat:

1. Загрузите пакет `.apk` по ссылке на [портале поддержки](#) или найдите его на поставляемом компакт-диске.
2. Установите официальный ключ BELLSOFT, как показано в следующем примере.

```
sudo wget -P /etc/apk/keys/ https://apk.axiomjdk.ru/info@bell-  
sw.com5fea454e.rsa.pub
```

3. Установите пакет как показано ниже.

```
apk add [Сервер приложений Axiom Libercat].apk
```

## Установка отдельного пакета Axiom Libercat из архива в GNU/Linux

Для установки отдельного пакета Axiom Libercat из архива, скачайте архив `tar.gz` (по ссылке на [портале поддержки](#)) и распакуйте его при помощи следующей команды:

```
tar -zxvf [Сервер приложений Axiom Libercat].tar.gz
```

Данная команда распакует Axiom Libercat в текущий каталог.

## Добавление пути к Axiom Libercat в переменные окружения

Axiom Libercat является приложением Java и не использует переменные окружения напрямую. Переменные окружения используются сценариями (скриптами) запуска Axiom Libercat. Скрипты используют переменные окружения для подготовки команды, запускающей Axiom Libercat.

Необходимо добавить и настроить следующие переменные:

- CATALINA\_HOME (обязательная) – Данная переменная должна указывать на местоположение корневого каталога бинарного дистрибутива Axiom Libercat.
- CATALINA\_BASE (рекомендуемая) – Данная переменная указывает на расположение каталога «активной конфигурации» Axiom Libercat. По умолчанию используется путь указанный в CATALINA\_HOME.

- JRE\_HOME or JAVA\_HOME (обязательные и обычно устанавливаются автоматически при установке Java) – Данная переменная указывает на расположение каталога установки Java.

Мы рекомендуем использовать разные значения для переменных CATALINA\_HOME и CATALINA\_BASE для упрощения дальнейших обновлений и обслуживания, а также для использования нескольких экземпляров Axiom Libercat.

Чтобы добавить необходимые пути Сервера приложений Axiom Libercat в переменные окружения в операционных системах на базе Debian требуется выполнить следующие операции:

1. Открыть конфигурационный файл `/etc/profile` в текстовом редакторе.
2. Установить переменные окружения PATH следующей командой:

```
export PATH=$PATH:<переменная>=<путь>
```

3. Сохранить изменения в файле и закрыть редактор. Для незамедлительного применения изменений следует выполнить следующую команду: `source /etc/profile`

## Другие опциональные переменные

Помимо четырех, описанных выше, существуют и другие переменные окружения. См. описание каждой из них в комментариях в верхней части сценариев `catalina.bat` или `catalina.sh`.

Одной из часто используемых переменных является CATALINA\_OPTS. Она позволяет уточнить дополнительные параметры для команды `java`, запускающей Axiom Libercat.

Аналогичная переменная – JAVA\_OPTS. Он используется реже и позволяет установить настройки для запуска и остановки Axiom Libercat, а также для других команд.



### Примечание:

Не используйте JAVA\_OPTS для указания предельного размера используемой памяти. Не нужно выделять много памяти для небольшого процесса, который используется для остановки Axiom Libercat. Эти настройки содержатся в CATALINA\_OPTS.

Другая часто используемая переменная – это CATALINA\_PID. Она указывает на расположение файла, в котором находится идентификатор разветвленного процесса Libercat java. Этот параметр является необязательным. Он, активирует следующие особенности:

- Защиту от дублирующих попыток запуска.
- Разрешает принудительное завершение процесса Axiom Libercat, когда он не реагирует на



стандартную команду выключения.

## Проверка корректности установки Axiom Libercat

Для проверки корректности установки изделия Axiom Libercat, откройте следующий адрес в адресной строке браузера.

```
http://localhost:8080
```

Вы должны увидеть экран приветствия Axiom Libercat.

## Деинсталляция Axiom Libercat

Для деинсталляции Axiom Libercat в Debian и других операционных системах, основанных на Debian выполните следующую команду:

```
sudo apt remove <имя_пакета>
```

Для деинсталляции Axiom Libercat в Red Hat Linux и других ОС на основе RPM выполните следующую команду:

```
sudo yum remove <имя_пакета>
```

Для деинсталляции Axiom Libercat в Alpine Linux выполните следующую команду:

```
sudo apk del <имя_пакета>
```

Если использовалась установка из архива, удалите директорию размещения Axiom Libercat и измените значение связанных переменных окружения.

## Особые случаи установки RPM и APK пакетов

При возникновении ошибок установки сертифицированных RPM или APK пакетов Axiom Libercat, связанных с неразрешимыми зависимостями, используйте следующие пути решения, в зависимости от используемого дистрибутива.

### RPM

При использовании RPM пакетов, используйте команду `rpm`, игнорирующую данные неразрешимые зависимости, как показано ниже:

```
rpm -Uvh --nodeps ./axiom-libercat9.1.3-2-noarch-linux.rpm
```

## APK

Если при установке Axiom Libercat 8 возникла ошибка связанная с неразрешимыми зависимостями, установите необходимую зависимость вручную.

Пример ошибки:

```
ERROR: unable to select packages:  
  axiomjdk8-jre (no such package):  
    required by: axiom-libercat8.0.16-7[axiomjdk8-jre]
```

Данная ошибка показывает, что пакету Axiom Libercat 8 требуется пакет, предоставляющий зависимость axiomjdk8-jre. Установите пакет Axiom JRE следующим образом:

```
apk add -t axiomjdk8-jre ./axiomjdk-jdk8u412+9-linux-x64-musl.apk
```

Если при установке Axiom Libercat 9 возникла ошибка связанная с неразрешимыми зависимостями, установите необходимую зависимость вручную.

Пример ошибки:

```
ERROR: unable to select packages:  
  axiomjdk11-jre (no such package):  
    required by: axiom-libercat9.1.3-2[axiomjdk11-jre]
```

Для Axiom Libercat 9 требуется зависимость axiomjdk11-jre, которая может быть предоставлена следующим образом:

```
apk add -t axiomjdk11-jre ./axiomjdk-jdk11.0.23+10-linux-x64-musl.apk
```



### Примечание:

Вы можете использовать установку Axiom Libercat из файлов `tar.gz` вместо использования менеджера пакетов, чтобы избежать ошибок, связанных с неразрешимыми зависимостями.

## Unix демон

Axiom Libercat может работать в качестве демона с использованием инструмента `jsvc` из проекта `commons-daemon`. Исходные tar-архивы входят в бинарные файлы Axiom Libercat и нуждаются в компилировании. Сборка `jsvc` требует наличия компилятора C ANSI (например, GCC), GNU Autoconf и JDK.

До запуска скрипта должна быть установлена переменная окружения `JAVA_HOME`. Так же путь к JDK можно указать при вызове скрипта `./configure` с помощью параметра `--with-java`, например:

```
./configure --with-java=/usr/java
```

Используя следующие команды, можно скомпилировать бинарный файл `jsvc`, расположенный в папке `$CATALINA_HOME/bin`. Для этого необходим GNU TAR и установленная переменная окружения `CATALINA_HOME`.

Обратите внимание, что нужно использовать GNU make (`gmake`), а не BSD make, установленный на системах FreeBSD.

```
cd $CATALINA_HOME/bin
tar xvfz commons-daemon-native.tar.gz
cd commons-daemon-1.1.x-native-src/unix
./configure
make
cp jsvc ../..
cd ../..
```

Теперь можно запустить Axiom Libercat как демон:

```
CATALINA_BASE=$CATALINA_HOME
cd $CATALINA_HOME
./bin/jsvc \
  -classpath $CATALINA_HOME/bin/bootstrap.jar:$CATALINA_HOME/bin/tomcat-
juli.jar \
  -outfile $CATALINA_BASE/logs/catalina.out \
  -errfile $CATALINA_BASE/logs/catalina.err \
  -Dcatalina.home=$CATALINA_HOME \
  -Dcatalina.base=$CATALINA_BASE \
  -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager \
  -Djava.util.logging.config.file=$CATALINA_BASE/conf/logging.properties \
  org.apache.catalina.startup.Bootstrap
```

При работе на Java 9 потребуется дополнительно указать следующие параметры при запуске `jsvc`, чтобы избежать предупреждений при остановке.

```
...
--add-opens=java.base/java.lang=ALL-UNNAMED \
--add-opens=java.base/java.io=ALL-UNNAMED \
--add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED \
...
```

Может потребоваться установить параметр `-jvm server`, если JVM по умолчанию использует ClientVM. Такое наблюдается на системах OSX.

`jsvc` имеет и другие полезные параметры, например, `-user`. Он позволяет сменить пользователя после завершения инициализации демона. Таким образом Axiom Libercat может работать от имени пользователя без администраторских прав, при этом с использованием администраторских портов. Обратите внимание, что, если вы хотите использовать эту опцию и запускать Axiom Libercat от имени администратора, необходимо отключить `org.apache.catalina.security.SecurityListener`, т.к. он не дает Axiom Libercat запускаться от имени администратора.

`jsvc --help` выдаст всю информацию по использованию `jsvc`. В частности, опция `-debug` полезна для отладки проблем, связанных с работой `jsvc`.

Файл `$CATALINA_HOME/bin/daemon.sh` можно использовать в качестве шаблона для запуска Axiom Libercat автоматически при загрузке системы из `/etc/init.d` `jsvc`.

Обратите внимание, что JAR-файл Commons-Daemon должен находиться в classpath для такой работы Axiom Libercat. JAR-файл Commons-Daemon находится в Class-Path манифеста `bootstrap.jar`, но, если вы получаете ошибки `ClassNotFoundException` или `NoClassDefFoundError` для класса Commons-Daemon, добавьте JAR-файл Commons-Daemon к аргументу `-cp` при запуске `jsvc`.

## Проверка загруженных файлов

Мы рекомендуем всегда проверять загруженный установочный файл, сравнив его размер на вашем диске с размером на [портале поддержки](#).

Более продвинутый подход — получить контрольную сумму загруженного файла в интерфейсе командной строки и сравнить ее с той, которую вы можете найти рядом со ссылкой на установочный файл. Ниже приведены команды для получения контрольной суммы для каждой ОС.

Для проверки воспользуйтесь программой для расчёта контрольной суммы: «Средство фиксации и контроля исходного состояния программного комплекса «ФИКС»».

## Windows

После скачивания пакета выполните в командной строке Windows команду для вычисления контрольной суммы ГОСТ Р 34.11-94. Например:

```
Isx_Sost.exe "#1" "&f1=[Путь к установщику Axiom Libercat].msi &f2=[Full path to out.txt] &u4"
```

Сравните контрольные суммы, полученные в «ФИКС» в файле out.txt и указанные на [портале поддержки](#). Если они отличаются, то снова повторите попытку скачивания пакета Axiom Libercat.

## Unix

Выполните следующую команду:

```
ufix [Пакет установки Axiom Libercat].deb
```

Сравните контрольные суммы, полученные в «ФИКС» и указанные на [портале поддержки](#). Если они отличаются, то снова повторите попытку скачивания пакета Сервер приложений Axiom Libercat.

## Устранение неполадок

В этом разделе содержатся инструкции по устранению некоторых наиболее распространенных проблем при установке.

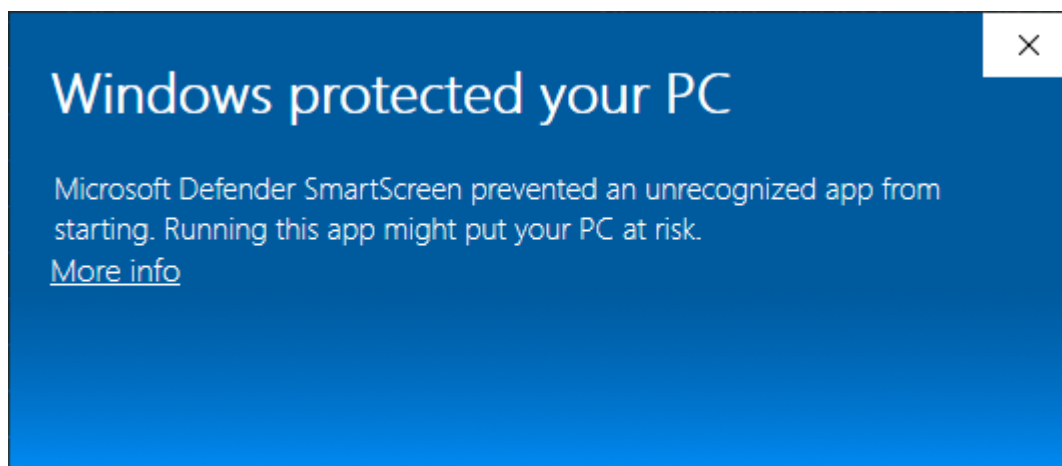
### Проверка системных требований

Axiom Libercat может работать на ряде платформ. Мы рекомендуем проверить совместимость вашей системы и требования на странице [Поддерживаемые системные конфигурации](#) перед установкой продукта Axiom Libercat.

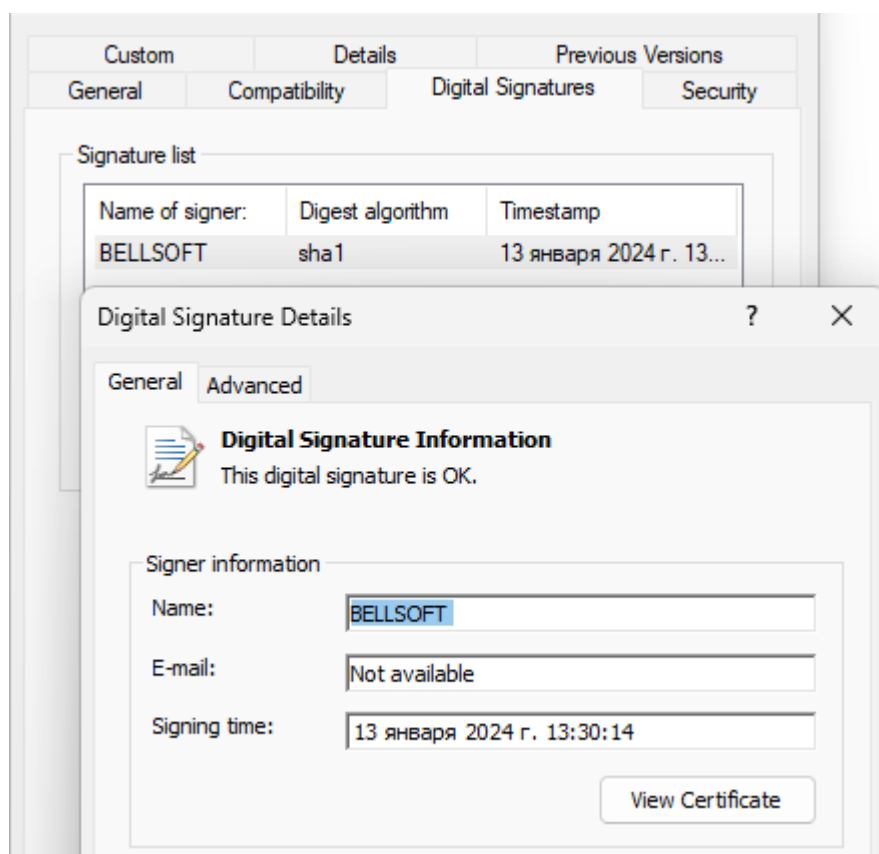
### Предупреждения от Microsoft Defender SmartScreen

Для обеспечения безопасности компьютера пользователя фильтр SmartScreen от Microsoft Defender иногда предотвращает запуск загруженных файлов и выдает предупреждения при установке программного обеспечения.

Если фильтр SmartScreen не позволяет установить продукт Axiom Libercat и отображает сообщение, подобное приведенному ниже, следуйте инструкциям в этом разделе.



1. Проверьте загруженный установочный файл Libercat, как описано в главе [Проверка загруженных файлов](#).
2. Убедитесь, что установочный файл подписан Axiom, проверив сведения о **цифровых подписях** в **свойствах** установочного файла в разделе **Digital Signatures**.



3. В зависимости от результатов проверки в пунктах 1 и 2 выполните следующие действия:
  - Если размер и контрольная сумма загруженного файла совпадают с указанными на вашей

странице на [портале поддержки](#) и файл подписан Axiom, нажмите **More info** в окне Microsoft Defender SmartScreen, а затем нажмите **Run anyway**.

- Если какая-либо информация неверна, удалите файл и снова загрузите установочный файл.

## Загрузка Axiom Libercat с помощью браузера

Если ваш браузер не позволяет загрузить установочный файл Axiom Libercat, нажмите **More actions** (три точки или стрелка) рядом с загруженным файлом и, в зависимости от браузера, выберите **Keep** или **Download**, чтобы сохранить файл.

## 3. Базовые настройки Axiom Libercat

### Каталоги и файлы

- `/bin` - Запуск, выключение и другие сценарии. Файлы с расширением `*.sh` (для систем Unix) являются функциональными копиями файлов с расширением `*.bat` (для систем Windows). Так как командная строка Win32 не имеет достаточного функционала, здесь также присутствуют дополнительные файлы.
- `/conf` - Файлы конфигурации и связанные с ними DTD. Самый важный файл – это `server.xml`. Это главный файл конфигурации контейнера.
- `/logs` - По умолчанию здесь расположены файлы журнала. `/webapps` - Место для установки веб-приложений.

### Переменные CATALINA\_HOME и CATALINA\_BASE

Вам необходимо знать и понимать значение как минимум двух переменных: CATALINA\_HOME и CATALINA\_BASE.

- CATALINA\_HOME содержит ссылку на корневую папку установки Axiom Libercat, например:

```
/home/Libercat/libercat-9.1.3
```

или для установки на системе Windows

```
C:\Program Files\libercat-9.1.3
```

- CATALINA\_BASE содержит ссылку на конфигурацию определенного экземпляра Axiom Libercat. Если вы хотите иметь несколько экземпляров Axiom Libercat на одной машине – используйте переменную CATALINA\_BASE.

Если вы используете различные экземпляры Axiom Libercat, то CATALINA\_HOME содержит статические данные, такие как файлы `.jar` или бинарные файлы. CATALINA\_BASE содержит файлы вашей конфигурации, журнала, развернутые приложения и другие необходимые ресурсы, то есть то, что необходимо сохранить при обновлении приложения.



## Преимущества использования CATALINA\_BASE

По умолчанию CATALINA\_HOME и CATALINA\_BASE указывают на один и тот же каталог. Вручную укажите в CATALINA\_BASE путь к экземпляру конфигурации Axiom Libercat, если вы хотите запускать несколько экземпляров Axiom Libercat на одной машине. Благодаря этому вы получаете следующие преимущества:

- Более легкий переход на новую версию Axiom Libercat. Так как все экземпляры с одним местоположением CATALINA\_HOME используют один набор файлов `.jar` и двоичных файлов, вы можете с легкостью обновить файлы до новой версии и применять изменения ко всем экземплярам Axiom Libercat, используя один каталог CATALINA\_HOME. См. раздел Обновление ниже.
- Отсутствие дублирования одних и тех же статических файлов `.jar`.
- Возможность совместного использования некоторых настроек, например, `setenv` или файла сценария `bat` (в зависимости от операционной системы).

## Содержимое CATALINA\_BASE

Перед началом использования CATALINA\_BASE создайте дерево каталогов, используемое CATALINA\_BASE.



### Примечание:

Если вы не создадите все рекомендуемые каталоги, Axiom Libercat создаст их автоматически. Если автоматически создать каталог не удастся, например из-за отсутствия разрешений, Axiom Libercat либо не запустится, либо будет работать неправильно.

Список каталогов:

- Каталог `bin` с файлами `setenv.sh`, `setenv.bat` и `tomcat-juli.jar`.

Рекомендуемый: Нет.

Порядок поиска: Сначала проверяется CATALINA\_BASE; если не найден, то используется CATALINA\_HOME.

- Каталог `lib` с ресурсами для добавления в `classpath`.

Рекомендуемый: Да, если ваше приложение зависит от внешних библиотек.

Порядок поиска: Сначала проверяется CATALINA\_BASE; затем загружается CATALINA\_HOME.

- Каталог logs для файлов журнала экземпляра.

Рекомендуемый: Да.

- Каталог webapps для автоматически загружаемых веб-приложений.

Рекомендуемый: Да, если вы хотите развертывать приложения.

Порядок поиска: Только CATALINA\_BASE.

- Каталог work, содержащий временные рабочие каталоги для развернутых веб-приложений.

Рекомендуемый: Да.

- Каталог temp, используемый JVM для временных файлов.

Рекомендуемый: Да.

Рекомендуется не изменять файл `tomcat-juli.jar`. Однако, если вы хотите использовать собственную систему записи в журнал работы, вы можете заменить файл `tomcat-juli.jar` в CATALINA\_BASE для определенного экземпляра Axiom Libercat.

Также рекомендуем скопировать все файлы конфигурации из каталога `CATALINA_HOME/conf` в каталог `CATALINA_BASE/conf`. Если файлы конфигурации отсутствуют в CATALINA\_BASE, они не будут прочитаны из CATALINA\_HOME. В дальнейшем это может вызвать отказ в работе.

## Установка CATALINA\_BASE

CATALINA\_BASE – это переменная окружения. Необходимо установить ее до запуска Axiom Libercat, например:

- В Unix: `CATALINA_BASE=/tmp/tomcat_base1 bin/catalina.sh start`
- В Windows: `CATALINA_BASE=C:\tomcat_base1 bin\catalina.bat start`

См главу «Установка Axiom Libercat».

## Использование скрипта setenv для настройки переменных окружения

Кроме CATALINA\_HOME и CATALINA\_BASE, все переменные окружения могут быть установлены с помощью скрипта "setenv". Скрипт помещается либо в `CATALINA_BASE/bin`, либо в

`CATALINA_HOME/bin` и называется `setenv.bat` (в Windows) или `setenv.sh` (в Unix). Файл должен быть читаемый. Переменные `CATALINA_HOME` и `CATALINA_BASE` нельзя настроить в `setenv`, потому что они используются для поиска этого файла.

По умолчанию файл сценария `setenv` отсутствует после установки Axiom Libercat. Если вы скопировали файл сценария как в `CATALINA_BASE`, так и в `CATALINA_HOME`, используется тот, что находится в `CATALINA_BASE`.

Например, чтобы настроить переменные `JRE_HOME` и `CATALINA_PID`, создайте следующий файл сценария:

В Windows `%CATALINA_BASE%\bin\setenv.bat`:

```
set "JRE_HOME=%ProgramFiles%\Java\jre8"
exit /b 0
```

В Unix, `$CATALINA_BASE/bin/setenv.sh`:

```
JRE_HOME=/usr/java/latest
CATALINA_PID="/run/tomcat.pid"
```

Все описанные здесь переменные окружения и скрипт `setenv` используется только в том случае, если вы используете стандартные сценарии для запуска Axiom Libercat. Например, если вы установили Axiom Libercat в качестве службы в Windows, оболочка службы запускает Java напрямую и не использует файлы сценариев.

## Использование logj4 версии 2

В поставку Libercat включены скрипты `libercat.setenv.bat`, `libercat.setenv.sh`, в которых есть команды для использования `log4j` вместо `JULI`. Для перехода на `log4j`, раскомментируйте строки, содержащие исполняемые команды и запустите скрипт. Вы можете продолжать использовать интерфейс Java `logger`, но журналирование будет делегировано в `log4j`. Также можно использовать классы `log4j` напрямую. Они находятся в пакете `ru.axiomjdk.org.apache.logging.log4j`. Возможность использования `log4j` второй версии, позволяет, в частности, устанавливать права на создаваемые лог файлы, отправлять лог в различные сборщики логов, отправлять лог в `syslog`, делать настройку логирования на уровне приложений Libercat EE, разделять лог старта сервера и лог от приложений по разным файлам.

## 4. Файлы запуска и конфигурации

Axiom Libercat, будучи основанным на Libercat содержит те же файлы запуска и конфигурации, что и Libercat а также дополняет их собственными.

### Скрипты запуска

Файлы с расширением `*.sh` (для систем Unix) являются функциональными копиями файлов с расширением `*.bat` (для систем Windows) содержатся в каталоге `bin`.

Список всех скриптов в каталоге `bin`:

| Скрипт                            | Описание                                                                                                                                  |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <code>catalina.sh(bat)</code>     | Скрипт управления Axiom Libercat сервером.                                                                                                |
| <code>ciphers.sh(bat)</code>      | Выводит список доступных алгоритмов шифрования.                                                                                           |
| <code>configtest.sh(bat)</code>   | Тест конфигурации Axiom Libercat сервера.                                                                                                 |
| <code>digest.sh(bat)</code>       | Генерирует строку хранимых учётных данных.                                                                                                |
| <code>makebase.sh</code>          | Создаёт <code>CATALINA_BASE</code> каталог (для конфигурации, когда <code>CATALINA_BASE</code> отличается от <code>CATALINA_HOME</code> ) |
| <code>setclasspath.sh(bat)</code> | Служебный скрипт, устанавливает переменные окружения для запуска команды <code>java</code> .                                              |
| <code>shutdown.sh(bat)</code>     | Останавливает Axiom Libercat сервер. Аналог <code>catalina.sh start</code> .                                                              |

| Скрипт               | Описание                                                                                                                    |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------|
| startup.sh(bat)      | Запускает Axiom Libercat сервер. Аналог <code>catalina.sh stop</code> .                                                     |
| tomee.sh(bat)        | Даёт возможность делать deploy и undeploy приложений, а также шифровать пароли (например для задания из в конфигурации БД). |
| tool-wrapper.sh(bat) | Служебный скрипт-оболочка для инструментов командной строки.                                                                |
| version.sh(bat)      | Выводит информацию о версиях Java, ОС, Axiom Libercat.                                                                      |

С помощью созданного скрипта (см. [Использование скрипта setenv для настройки переменных окружения](#)) можно задавать переменные среды, которые будут использованы скриптом `catalina.sh(bat)`.

Для запуска и остановки Axiom Libercat используется скрипт `catalina.sh(bat)` (или `shutdown.sh(bat)`).

Список всех команд скрипта `catalina.sh(bat)`:

| Команда         | Описание                                                                                                      |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| debug           | Запускает Axiom Libercat сервер в режиме отладки (в командной строке, через команду <code>jdb</code> ).       |
| debug -security | Запускает Axiom Libercat сервер в режиме отладки с менеджером безопасности ( <code>security manager</code> ). |

| Команда         | Описание                                                                                                                                            |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| jpda start      | Запускает Axiom Libercat сервер в режиме JPDA отладки (отладчик подключается удалённо).                                                             |
| run             | Запускает Axiom Libercat сервер в текущем окне (не фоновым процессом)                                                                               |
| run -security   | Запускает Axiom Libercat сервер в текущем окне с менеджером безопасности.                                                                           |
| start           | Запускает Axiom Libercat сервер в отдельном окне (фоновым процессом)                                                                                |
| start -security | Запускает Axiom Libercat сервер в отдельном окне с менеджером безопасности.                                                                         |
| stop            | Останавливает Axiom Libercat и ждёт 5 секунд до завершения процесса                                                                                 |
| stop            | Останавливает Axiom Libercat и ждёт n секунд до завершения процесса. (Не для Windows систем)                                                        |
| stop -force     | Останавливает Axiom Libercat, ждёт 5 секунд до завершения процесса и использует "kill -KILL" если процесс всё ещё работает. (Не для Windows систем) |
| stop n -force   | Останавливает Axiom Libercat, ждёт n секунд до завершения процесса и использует "kill -KILL" если процесс всё ещё работает. (Не для Windows систем) |

| Команда    | Описание                                                                                |
|------------|-----------------------------------------------------------------------------------------|
| configtest | Запускает базовую проверку синтаксиса конфигурационного файла <code>server.xml</code> . |
| version    | Выводит информацию о версиях Java, ОС, Axiom Libercat.                                  |
| help       | Выводит возможные аргументы скрипта и их назначение.                                    |

- Для запуска сервера можно использовать команды `catalina.sh(bat) start` / `catalina.sh(bat) run`.
- Для останова сервера можно использовать команды `catalina.sh(bat) stop` / `shutdown.sh(bat)`.

Скрипт `ciphers.sh(bat)` выводит список доступных алгоритмов шифрования. Список всех опций скрипта `ciphers.sh(bat)`:

| Опция     | Альтернативная команда | Описание                                                         |
|-----------|------------------------|------------------------------------------------------------------|
| --help    | -h                     | Выводит возможные опции скрипта и их назначение.                 |
| --openssl |                        | Показывает OpenSSL имена вместо IANA имён алгоритмов шифрования. |
| --verbose | -v                     | Показывает детальный листинг алгоритмов шифрования.              |

Скрипт `digest.sh(bat)` генерирует строку хранимых учётных данных. Список всех опций скрипта `digest.sh(bat)`:

| Опция | Описание                                                                                                                                                                                                                                                                                                     |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a    | Задаёт алгоритм шифрования. Если не задан, то используется SHA-512.                                                                                                                                                                                                                                          |
| -e    | Задаёт кодировку для преобразований (по необходимости) байт в символ или наоборот. Если не задана, то используется <code>Charset.defaultCharset()</code> .                                                                                                                                                   |
| -i    | Число итераций во время генерирования. Если не задан, то используется значение по умолчанию класса, реализующего интерфейс <code>CredentialHandler</code> (см. опцию -h).                                                                                                                                    |
| -s    | Размер массива байт соли для генерации. Если не задана, то используется значение по умолчанию класса, реализующего интерфейс <code>CredentialHandler</code> (см. опцию -h).                                                                                                                                  |
| -k    | Количество бит ключа (ключей), создаваемых во время генерации. Если не задано, то используется значение по умолчанию класса, реализующего интерфейс <code>CredentialHandler</code> (см. опцию -h).                                                                                                           |
| -h    | Полное имя (FQN) класса, реализующего <code>CredentialHandler</code> . Если не задано встроенные реализации ( <code>MessageDigestCredentialHandler</code> и <code>SecretKeyCredentialHandler</code> ) будут проверены по очереди и первый пригодный для обработки выбранного алгоритма будет использован (). |



| Опция | Описание                                                                                                                                                                                                     |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -f    | Имя файла, содержащего пароль для шифрования. Каждая строка в файле должна содержать только один пароль. При использовании этой опции игнорируются аргументы скрипта (которые задают пароли для шифрования). |

Скрипт `makebase.sh(bat)` создаёт каталог `CATALINA_BASE`. Скрипт запускается следующим образом:

```
makebase <path-to-target-directory> [-w | --webapps]
```

Здесь:

- Первый аргумент `path-to-target-directory` задаёт каталог где будет создана структура `CATALINA_BASE`.
- Второй аргумент `-w` определяет нужно ли копировать `webapps`.

Скрипт `tomee.sh(bat)` расширяет функционал скриптов Libercat, обращаясь к реализации EE части Axiom Libercat сервера.

Список всех аргументов скрипта `digest.sh(bat)`:

- `cipher` - Шифрует/дешифрует пароль, указанный аргументом, следующим за аргументом `cipher` и выводит его на консоль. По умолчанию используется шифр `Static3DES`, но шифр можно указать с помощью опции `-c`.
  - `-c/--cipher` Указывает шифр
  - `-d/--decrypt` Дешифрует пароль.
  - `-e/--encrypt` Шифрует пароль (по умолчанию)
  - `-h/--help` Выводит список аргументов и их значение
- `deploy` - Разворачивает EJB jar или EAR на Axiom Libercat сервере.
  - `-d/--debug` Повышает уровень детализации ошибок валидации и отчёта развёртывания
  - `--dir` Устанавливает целевой каталог развёртывания приложения. По умолчанию это

`{CATALINA_BASE}/apps/`. Если этот каталог изменён, то, если он не прописан в `tomee.xml` (`<Deployments dir=""/>`) при рестарте приложение не будет развёрнуто.

- `-h/--help` Выводит информацию по аргументам и их описание.
- `-o/--offline` Разворачивает приложение на незапущенный сервер с помощью копирования архива приложения в целевой каталог. Приложение будет полноценно развёрнуто только при старте сервера.
- `-q/--quiet` Понижает уровень детализации валидации и не выводит отчёт развёртывания.
- `-s/--server-url <url>` Устанавливает URL Axiom Libercat сервера на который происходит развёртывание приложения. Значение должно быть таким же как URL JNDI провайдера, используемого для поиска (lookup) компонент. По умолчанию `ejbd://localhost:4201`.
- `-u/--undeploy` Удаляет приложение с сервера и из директории развёртывания.
- `-v/--version` Выводит OpenEJB (реализацию функциональности J2EE в Axiom Libercat) версию.
- `effectivetomee` - Показывает свойства, которые Axiom Libercat будет использовать для ресурсов.
- `properties` - Преобразовывает и распечатывает конфигурацию Axiom Libercat в виде свойств.
- `setters` - Выводит список свойств, доступных для класса.
  - `-c/--class` Задаёт имя класса в котором будут искажаться свойства.
- `undeploy` - Удаляет EJB jar или EAR с Axiom Libercat сервера.
  - `-h/--help` Выводит информацию по аргументам и их описание.
  - `-s/--server-url <url>` Устанавливает URL Axiom Libercat сервера с которого удаляется приложение. Значение должно быть таким же как URL JNDI провайдера, используемого для поиска (lookup) компонент. По умолчанию `ejbd://localhost:4201`.
  - `-v/--version` Выводит OpenEJB (реализацию функциональности J2EE в Axiom Libercat) версию.

## Конфигурационные файлы

Конфигурационные файлы находятся в каталоге `conf`.

Axiom Libercat добавляет свои конфигурационные файлы к файлам Libercat.

- `system.properties` - содержит свойства специфичные для Axiom Libercat (в основном это настройки функционала `openejb`). Все эти свойства могут быть выставлены (или переопределены) и помощью установки системных Java свойств через командную строку (`-D<property-name>=<property-value>`). Так как это аргументы команды `java` их нужно задавать через механизмы установки аргументов для Java (не для скрипта запуска), например с помощью переменной окружения `CATALINA_OPTS`, которую в свою очередь можно задать в файле `setenv.sh(bat)`.
- `tomee.xml` - конфигурационный файл Axiom Libercat в XML формате, позволяющий описывать настройки сущностей подобно настройкам в файле `server.xml` для Libercat. В этом файле можно сконфигурировать ресурсы (например `DataSource`), контейнеры, расположение каталогов и файлов развёртывания приложений.

Файл `system.properties` имеет обычный формат файла свойств Java. В файле `tomee.xml` пользовательские свойства прописываются в текстовом формате, используя либо нотацию разделения через `=` либо через пробел. Это отличает формат файла `tomee.xml` от файла `server.xml`, где все свойства указываются через атрибуты XML тега.

Пример:

```
<Resource id="moviesDatabase" type="DataSource">
  userName= postgres
  password = postgres
  jdbcDriver = org.postgresql.Driver
  jdbcUrl = jdbc:postgresql://localhost:5432/postgres
</Resource>
```

Здесь `id` и `type` это служебные свойства ресурса, которые указаны как атрибуты XML тэга, все пользовательские настройки указываются в теле текста XML элемента с пробелом в качестве разделителя. Также можно использовать `=` в качестве разделителя имени свойства и его значения.

Значение `id` должно быть уникальным, для того чтобы идентифицировать ресурс.

## 5. Host Manager – Текстовый интерфейс

### Введение

Приложение **Axiom Libercat Host Manager** позволяет создавать и удалять виртуальные хосты и управлять ими в Axiom Libercat. Это руководство представлено вместе со следующей документацией:

- [Виртуальный хостинг и Axiom Libercat](#) для получения более подробной информации о виртуальном хостинге.
- [The Host Container](#) для получения более подробной информации о базовой конфигурации xml виртуальных хостов и описании атрибутов.

Приложение **Axiom Libercat Host Manager** является частью установки Axiom Libercat и по умолчанию использует следующий контекст: `/host-manager`. Вы можете использовать host manager следующим образом:

- Используя графический интерфейс пользователя, доступный в: `{server}:{port}/host-manager/html`.
- Используя набор минимальных HTTP-запросов, подходящих для написания скриптов. Этот режим доступен здесь: `{server}:{port}/host-manager/text`.

Оба способа позволяют добавлять, удалять, запускать и останавливать виртуальные хосты. Изменения могут быть сохранены с помощью команды `persist`. В этом документе рассматривается текстовый интерфейс. Для получения более подробной информации о графическом интерфейсе см. [Host Manager App – HTML Interface](#).

### Конфигурирование доступа к приложению Manager

Описание ниже использует `$CATALINA_HOME`, чтобы соотнести с базовым каталогом Axiom Libercat. Это каталог для установки Axiom Libercat, например, `C:/tomcat8` или `/usr/share/tomcat8`.

Host Manager требует от пользователя одной из следующих ролей:

- `admin-gui` - используйте эту роль для графического веб-интерфейса,
- `admin-script` - используйте эту роль для веб-интерфейса при написании скриптов.

Чтобы включить доступ к текстовому интерфейсу Host Manager, либо присвойте пользователю надлежащую роль, либо создайте новый профиль с подходящей ролью. Например, откройте `${CATALINA_BASE}/conf/tomcat-users.xml` и введите следующее:

```
<user username="test" password="chang3m3N#w" roles="admin-script"/>
```

В последующих настройках нет необходимости. Теперь, когда вы получите доступ к `{server}:{port}/host-manager/text/{COMMAND}`, вы сможете войти с новыми учетными данными. Например:

```
$ curl -u ${USERNAME}:${PASSWORD} http://localhost:8080/host-  
manager/text/list  
OK - Listed hosts  
localhost:
```

Помните, что в случае, если вы извлечете своих пользователей, используя механизм `DataSourceRealm`, `JDBCRealm` или `JNDIRealm`, добавьте подходящую роль в базу данных или сервер каталогов, соответственно.

## Список команд

Поддерживаются следующие команды:

- list
- add
- remove
- start
- stop
- persist

В следующих подразделах имя пользователя и пароль предполагаются **test:test**. Используйте учетные данные, созданные в предыдущих разделах, для вашей рабочей среды.

### Команда List

Используйте команду **list**, чтобы увидеть доступные виртуальные хосты в вашем экземпляре Axiom Libercat.

*Пример команды:*

```
curl -u test:test http://localhost:8080/host-manager/text/list`
```

*Пример ответа:*

```
OK - Listed hosts
localhost:
```

## Команда Add

Используйте команду **add**, чтобы добавить новый виртуальный хост. Параметры, используемые для команды **add** :

- Строка **name**: Имя виртуального хоста. **Обязательна**
- Строка **aliases**: Псевдонимы вашего виртуального хоста.
- Строка **appBase**: Базовый путь приложения, которое будет обслуживаться этим виртуальным хостом. Укажите относительный или абсолютный путь.
- Булево **manager**: Если истина, приложение Manager добавлено к виртуальному хосту. Вы можете получать доступ к нему с контекстом */manager*.
- Булево **autoDeploy**: Если истина, Axiom Libercat автоматически перезагружает приложения, размещенные в каталоге appBase.
- Булево **deployOnStartup**: Если истина, Axiom Libercat автоматически загружает приложения, размещенные в каталоге appBase при запуске.
- Булево **deployXML**: Если истина, файл */META-INF/context.xml* прочитан и используется в Axiom Libercat.
- Булево **copyXML**: Если истина, Axiom Libercat копирует файл */META-INF/context.xml* и использует оригинальную копию независимо от обновлений файла приложения/META-INF/context.xml\_.

*Пример команды:*

```
curl -u test:test http://localhost:8080/host-
manager/text/add?name=www.awesome-server.com&aliases=awesome-server.com&appBase
/mnt/appDir&deployOnStartup=true
```

*Пример ответа:*

```
add: Adding host [www.awesomeserver.com]
```

## Команда Remove

Используйте команду **remove** для удаления виртуального хоста. Параметры, используемые для команды **remove** :

- Строка **name**: Имя виртуального хоста, выбранного для удаления. **Обязательна**

*Пример команды:*

```
curl -u test:test http://localhost:8080/host-  
manager/text/remove?name=www.awesomeserver.com
```

*Пример ответа:*

```
remove: Removing host [www.awesomeserver.com]
```

## Команда Start

Используйте команду **start** для запуска виртуального хоста. Параметры, используемые для команды **start** :

- Строка **name**: Имя виртуального хоста, выбранного для запуска. **Обязательна**

*Пример команды:*

```
curl -u test:test http://localhost:8080/host-  
manager/text/start?name=www.awesomeserver.com
```

*Пример ответа:*

```
OK - Host www.awesomeserver.com started
```

## Команда Stop

Используйте команду **stop** для останова виртуального хоста. Параметры, используемые для команды **stop** :

- Строка **name**: Имя виртуального хоста, выбранного для останова. **Обязательна**

*Пример команды:*

```
curl -u test:test http://localhost:8080/host-  
manager/text/stop?name=www.awesome-server.com
```

*Пример ответа:*

```
OK - Host www.awesome-server.com stopped
```

## Команда Persist

Используйте команду **persist** для принуждения виртуального хоста в **server.xml**. Параметры, используемые для команды **persist** :

- Строка **name**: Имя виртуального хоста для принуждения. **Обязательна**

Эта функция по умолчанию отключена. Чтобы включить эту опцию, сначала настройте слушатель `StoreConfigLifecycleListener` . Для этого добавьте следующий слушатель в свой *server.xml*:

```
<Listener  
  className="org.apache.catalina.storeconfig.StoreConfigLifecycleListener"/>
```

*Пример команды:*

```
curl -u test:test http://localhost:8080/host-  
manager/text/persist?name=www.awesome-server.com
```

*Пример ответа:*

```
OK - Configuration persisted
```

*Пример ручного ввода:*

```
<Host appBase="www.awesome-server.com" name="www.awesome-server.com"  
  deployXML="false" unpackWARs="false">  
</Host>
```



## 6. Руководство по приложению Manager

### Введение

Во многих ситуациях промышленной эксплуатации очень полезно иметь возможность развернуть новое веб-приложение или свернуть существующее приложение без необходимости выключения и перезапуска всего контейнера. Кроме того, вы можете затребовать, чтобы существующее приложение перезагрузилось, даже если вы не объявили его `reloadable` в файле конфигурации сервера Axiom Libercat.

Для поддержки этих возможностей Axiom Libercat содержит веб-приложение (устанавливаемое по умолчанию на пути контекста `/manager`), которое поддерживает следующие функции:

- Развертывание нового веб-приложения из подгруженного контента WAR-файла.
- Развёртывание через приложение Manager работает **только для веб-приложений** и **не работает для J2EE приложений**.
- Развертывание нового веб-приложения на пути контекста из файловой системы сервера.
- Список развернутых в настоящий момент веб-приложений, а также сессий, которые в данный момент активны для этих веб-приложений.
- Перезагрузка существующего веб-приложения для отражения изменений в содержимом `/WEB-INF/classes` или `/WEB-INF/lib`.
- Перечень значений свойств OS и JVM.
- Список доступных глобальных ресурсов JNDI для использования в инструментах развертывания, которые подготавливают `<ResourceLink>` элементы, вложенные в `<Context>` описание развертывания.
- Запуск остановленного приложения (таким образом делая его снова доступным).
- Останов существующего приложения (так что оно становится недоступным) без его сворачивания.
- Сворачивание развернутого веб-приложения и удаление его базового каталога документов (если только оно не было развернуто из файловой системы).

Установка Axiom Libercat по умолчанию включает в себя Manager (Диспетчер). Чтобы добавить экземпляр веб-приложения Manager `Context` к новому хосту, установите файл конфигурации

контекста `manager.xml` в папку `$CATALINA_BASE/conf/[enginename]/[hostname]`. Пример:

```
<Context privileged="true" antiResourceLocking="false"
    docBase="${catalina.home}/webapps/manager">
    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\.0\.0\.1" />
</Context>
```

Если у вас Axiom Libercat настроен на поддержку виртуальных хостов (веб-сайтов), вам понадобится сконфигурировать Manager для каждого.

Имеется три способа использования веб-приложения **Manager**.

- В качестве приложения с пользовательским интерфейсом, который вы используете в своем браузере. Здесь приведен пример URL, где вы можете заменить `localhost` на имя хоста своего веб-сайта: <http://localhost:8080/manager/html>.
- Минимальная версия, использующая только HTTP-запросы, которая пригодна для использования скриптами, настраиваемыми системными администраторами. Команды выдаются как часть запроса URI, а ответы имеют форму простого текста, который можно легко проанализировать и обработать. См. [Поддерживаемые команды диспетчера] (#Поддерживаемые-команды-диспетчера) для получения подробной информации.
- Удобный набор описаний задач для инструмента сборки *Ant* (версии 1.4 или поздней). См. [Выполнение команд диспетчера с помощью Ant](#) для получения подробной информации.

## Конфигурирование доступа к приложению Manager

Описание ниже использует имя переменной `$CATALINA_BASE` для соотнесения с базовым каталогом, с которым соотносится большинство относительных путей. Если вы не сконфигурировали Axiom Libercat для работы в несколько экземпляров, задав каталог `CATALINA_BASE`, то значение `$CATALINA_BASE` будет установлено как значение `$CATALINA_HOME`, каталог установки Axiom Libercat.

Было бы совершенно небезопасно поставлять Axiom Libercat с настройками по умолчанию, которые позволили бы любому в Интернете исполнять приложение Manager на вашем сервере. Поэтому приложение Manager поставляется с требованием, чтобы любой, кто попытается его использовать, должен был себя аутентифицировать с помощью имени пользователя и пароля, имеющих одну из ролей **manager-xxx**, связанных с ними (имя роли зависит от требуемой функциональности). Более того, в файле пользователей по умолчанию (`$CATALINA_BASE/conf/tomcat-users.xml`) нет имени пользователя, назначенного этим ролям. Поэтому доступ к приложению Manager по умолчанию полностью отключен.

Вы можете найти имена ролей в файле `web.xml` веб-приложения Manager. Доступные роли:

- **manager-gui** – Доступ к интерфейсу HTML.
- **manager-status** – Доступ только к странице «Server Status» («Статус сервера»).
- **manager-script** – Доступ к дружественному по отношению к инструментам интерфейсу обычного текста, описанному в данном документе, и к странице «Server Status».
- **manager-jmx** – Доступ к JMX-прокси интерфейсу и к странице «Server Status».

Интерфейс HTML защищен от атак CSRF (Межсайтовая подделка запроса), но текстовый и JMX-интерфейсы невозможно защитить. Это означает, что тем пользователям, которым разрешен доступ к текстовому и JMX-интерфейсам, нужно быть осторожными при получении доступа к приложению Manager с помощью веб-браузера. Для поддержания защиты от CSRF:

- Если для доступа к приложению Manager вы используете веб-браузер с использованием пользователя, имеющего роли **manager-script** или **manager-jmx** (например, для тестирования интерфейса обычного текста или JMX-интерфейса), вы ДОЛЖНЫ после этого закрыть все окна браузера, чтобы завершить сессию. Если вы не закроете браузер и зайдете на другие сайты, вы можете стать жертвой CSRF-атаки.
- Рекомендуется никогда не давать ролей **manager-script** или **manager-jmx** пользователям, имеющим роль **manager-gui**.



#### Примечание:

JMX-прокси интерфейс является эффективным низкоуровневым корнеподобным административным интерфейсом контейнера Axiom Libercat. Можно сделать многое, если знать, какие команды вызывать. Вам следует соблюдать осторожность при активации роли **manager-jmx**.

Чтобы разрешить доступ к веб-приложению Manager, вам нужно либо создать новую комбинацию имени пользователя/пароля и связать с ней одну из ролей **manager-xxx**, либо добавить роль **manager-xxx** к какой-нибудь существующей комбинации имени пользователя/пароля. Поскольку большая часть этого документа описывает использование текстового интерфейса, в этом примере будет использоваться имя роли **manager-script**. Точный способ конфигурирования имен пользователей/паролей зависит от того, какой [Realm implementation](#) вы используете:

- *UserDatabaseRealm* плюс *MemoryUserDatabase* или *MemoryRealm* – *UserDatabaseRealm* и *MemoryUserDatabase* конфигурируются в принимаемом по умолчанию `$CATALINA_BASE/conf/server.xml`. Как *MemoryUserDatabase*, так и *MemoryRealm* читают файл формата XML, по умолчанию хранящийся в `$CATALINA_BASE/conf/tomcat-users.xml`,

который можно редактировать любым текстовым редактором. Этот файл содержит XML `<user>` для каждого отдельного пользователя, который может выглядеть следующим образом:

```
<user username="craigmcc" password="secret" roles="standard,manager-script" />
```

который определяет имя пользователя и пароль, используемые этим индивидуумом для входа в систему, и имена ролей, с которыми он или она ассоциированы. Вы можете добавить роль **manager-script** в разделенный запятыми атрибут `roles` для одного или нескольких существующих пользователей и/или создать новых пользователей с присвоенной ролью.

- *DataSourceRealm* или *JDBCRealm* – Ваша информация о пользователях и ролях хранится в базе данных, доступ к которой осуществляется через JDBC. Добавьте роль **manager-script** одному или нескольким существующим пользователям и/или создайте одного или несколько новых пользователей с этой ролью, следуя стандартным процедурам для вашей среды.
- *JNDIRealm* – Ваша информация о пользователях и ролях хранится в службе каталогов, доступ к которой осуществляется через LDAP. Добавьте роль **manager-script** одному или нескольким существующим пользователям и/или создайте одного или несколько новых пользователей с этой ролью, следуя стандартным процедурам для вашей среды.

Когда вы в первый раз попытаетесь выдать одну из команд приложения Manager, описанных в следующем разделе, вам будет предложено зарегистрироваться с помощью БАЗОВОЙ аутентификации. Вводимые вами имя пользователя и пароль не имеют значения, поскольку они идентифицируют правомочного пользователя в базе данных пользователей, который обладает ролью **manager-script**.

Вдобавок к ограничениям по паролю доступ к веб-приложению Manager может ограничиваться **remote IP address** или хостом путем добавления `RemoteAddrValve` или `RemoteHostValve`. См. [valves documentation](#), чтобы увидеть подробную информацию. Пример ограничения доступа к localhost по IP-адресу:

```
<Context privileged="true">  
    <Valve className="org.apache.catalina.valves.RemoteAddrValve"  
        allow="127\.0\.0\.1"/>  
</Context>
```

## Интерфейс HTML

Ориентированный на пользователя интерфейс HTML веб-приложения Manager находится по адресу:

```
http://{host}:{port}/manager/html
```

Как уже говорилось выше, вам нужна роль **manager-gui**, чтобы вам был разрешен доступ к нему. Имеется отдельный документ, предоставляющий помощь по этому интерфейсу: [HTML Manager documentation](#).

Интерфейс HTML защищен от атак CSRF (Межсайтовая подделка запроса). Каждый доступ к страницам HTML генерирует случайный токен, который хранится в вашей сессии и включается во все ссылки на странице. Если ваше следующее действие не будет иметь правильного значения токена, это действие будет отклонено. Если время токена истекло, вы можете начать снова с главной страницы или со страницы *List Applications* приложения Manager.

## Поддерживаемые команды диспетчера

Все команды, которые приложение Manager знает, как обрабатывать, определяются в одном запросе URI наподобие этого:

```
http://{host}:{port}/manager/text/{command}?{parameters}
```

где `{host}` и `{port}` представляют собой имя хоста и номер порта, на которых работает Axiom Libercat, `{command}` представляет собой команду приложения Manager, которую вы хотите выполнить, а `{parameters}` представляет собой параметры запроса, специфичные для этой команды. В иллюстрациях ниже настройте хост и порт в соответствии с вашей установкой.

Команды обычно исполняются запросами HTTP GET. Команда `/deploy` имеет форму, которая исполняется запросом HTTP PUT.

### Общие параметры

Большинство команд принимают один или более из следующих параметров запроса:

- **path** – Путь контекста (включая начальную косую черту) веб-приложения, с которым вы работаете. Для выбора КОРНЕВОГО веб-приложения укажите `/`. **ПРИМЕЧАНИЕ:** В самом приложении Manager невозможно выполнять административные команды.
- **version** – Версия этого веб-приложения, используемая функцией [parallel deployment](#). Если вы используете параллельное развертывание везде, где требуется путь, вам следует указать версию вдобавок к пути, и именно сочетание пути и версии должно быть уникальным, а не просто путь.
- **war** – URL файла архива веб-приложения (WAR) или имя пути каталога, который содержит веб-приложение, либо файл `«.xml»` конфигурации контекста (Context). Вы можете использовать

указатели URL в любом из следующих форматов:

- **file:/absolute/path/to/a/directory** – Абсолютный путь каталога, который содержит распакованную версию веб-приложения. Этот каталог будет прикреплен к пути контекста, который вы указываете без всяких изменений.
- **file:/absolute/path/to/a/webapp.war** – Абсолютный путь файла архива веб-приложения (WAR). Это действительно **только** для команды `/deploy`, и это единственный приемлемый формат для этой команды.
- **file:/absolute/path/to/a/context.xml** – Абсолютный путь файла «.xml» конфигурации контекста веб-приложения, который содержит элемент конфигурации контекста (Context configuration).
- **directory** – Имя каталога для контекста веб-приложения в базовом каталоге приложений хоста.
- **webapp.war** – Имя файла war веб-приложения, находящегося в базовом каталоге приложений хоста.

Каждая команда будет возвращать ответ в формате `text/plain` (т. е. обычный ASCII без всякой разметки HTML), облегчающем чтение как людьми, так и программами). Первая строка ответа будет начинаться либо с `OK` либо `FAIL`, указывая, была ли команда запроса успешной или нет. В случае сбоя остальная часть первой строки будет содержать описание встретившейся проблемы. Некоторые команды содержат дополнительные строки с информацией, как это описано ниже.

*Примечание по локализации* – Приложение Manager ищет свои строки сообщений в наборах ресурсов, поэтому возможно, что эти строки были переведены для вашей платформы. В примерах ниже показана версия сообщений на английском языке.

## Удаленное развертывание нового архива приложения (WAR)

`http://localhost:8080/manager/text/deploy?path=/foo`

Загрузите файл архива веб-приложения (WAR), который указан как данные запроса в этом запросе HTTP PUT, установите его в каталог `appBase` соответствующего виртуального хоста, и запустите, извлекая имя файла WAR, добавленное в `appBase` из указанного пути. Потом приложение можно будет свернуть (и удалить соответствующий файл WAR) при помощи команды `/undeploy`.

Эта команда выполняется запросом HTTP PUT.

Файл с расширением `.WAR` может содержать специфичную для Axiom Libercat конфигурацию

развертывания, путем включения XML-файла конфигурации контекста в `/META-INF/context.xml`.

В число параметров URL входят следующие:

- `update` - Если установлено в значение «истина», любое существующее обновление будет свернуто первым. Значение по умолчанию установлено в «ложь».
- `tag` - Указывает имя тега, это позволяет связывать развернутое веб-приложение с тегом или DNS-именем. Если веб-приложение свернуто, его можно потом при необходимости снова развернуть с помощью одного только тега.



#### Примечание:

Эта команда логически противоположна команде `/undeploy`.

Если установка и запуск завершились успешно, вы получите следующий ответ:

```
OK - Deployed application at context path /foo
```

В противном случае ответ будет начинаться с `FAIL` и будет включать в себя сообщение об ошибке. Возможные причины проблем включают:

- *Приложение уже существует по пути /foo*

Контекстные пути должны быть уникальны для всех запущенных в данный момент веб-приложений. Таким образом, вы должны свернуть существующее веб-приложение при помощи этого контекстного пути или выбрать другой путь для нового. Параметр `update` может быть задан как параметр URL, со значением `true`, чтобы избежать этой ошибки. В этом случае свертывание будет реализовано на существующем приложении до развертывания.

- *Возможные исключения*

При запуске нового веб-приложения возникло исключение. Для просмотра подробностей просмотрите журнал Axiom Libercat, но объяснения проблемы скорее всего содержатся в `/WEB-INF/web.xml` файле, или найдены отсутствующие классы при инициализации слушателей событий и фильтров приложения.

## Развертывание нового приложения из локального пути

Разверните и запустите новое веб-приложение, прикрепленное к выбранному контексту `path` (который не может использоваться другим веб-приложением). Эта команда логически противоположна `/undeploy`.

Эта команда выполняется запросом HTTP PUT. Существует ряд разных способов использования команды развертывания.

## Развертывание ранее развернутого веб-приложения

```
http://localhost:8080/manager/text/deploy?path=/footoo&tag=footag
```

Ее можно использовать для развертывания ранее развернутого веб-приложения, которое было развернуто с помощью атрибута `tag`. Обратите внимание, что рабочий каталог веб-приложения Manager будет содержать развернутые ранее архивы WAR; его удаление может привести к невыполнению развертывания.

## Развертывание каталога или WAR по URL-адресу

Разверните каталог веб-приложения или файл «.war», находящийся на сервере Axiom Libercat. Если `path` не указан, путь и версию можно извлечь из имени каталога или имени war-файла. Параметр `war` указывает URL (включая схему `file:` ) либо для каталога, либо для файла архива веб-приложения (WAR). Поддерживаемый синтаксис для URL, относящегося к WAR-файлу, описан на странице Javadocs для класса `java.net.JarURLConnection` . Используйте только те URL, которые относятся ко всему WAR-файлу.

В этом примере веб-приложение, находящееся в каталоге `/path/to/foo` на сервере Axiom Libercat, запущено как контекст веб-приложения под именем `/footoo`.

```
http://localhost:8080/manager/text/deploy?path=/footoo&war=file:/path/to/foo
```

В данном примере файл «.war» `/path/to/bar.war` на сервере Axiom Libercat запущен как контекст веб-приложения под именем `/bar`. Заметьте, что здесь нет параметра `path`, поэтому путь контекста по умолчанию устанавливается в имя файла архива веб-приложения без расширения `.war`.

```
http://localhost:8080/manager/text/deploy?war=file:/path/to/bar.war
```

## Развертывание каталога или WAR из базы приложений хоста

Разверните каталог веб-приложения или файл `.war`, находящийся в вашем каталоге Host appBase. Путь и необязательная версия извлекаются из каталога или имени war-файла.

В этом примере веб-приложение, находящееся в подкаталоге под именем `foo`, в каталоге Host appBase сервера Axiom Libercat развертывается как контекст веб-приложения под именем `/foo`.



Заметьте, что используемый путь контекста – это имя каталога веб-приложения.

```
http://localhost:8080/manager/text/deploy?war=foo
```

В данном примере файл «.war» `bar.war`, находящийся в каталоге Host appBase сервера Axiom Libercat, развертывается как контекст веб-приложения под именем `/bar`.

```
http://localhost:8080/manager/text/deploy?war=bar.war
```

## Развертывание с использованием XML-файла конфигурации Context

Если флаг Host `deployXML` установлен в значение «истина», вы можете развернуть веб-приложение с помощью файла `.xml` конфигурации контекста и дополнительного файла `.war` или каталога веб-приложения. Контекст `path` не используется при развертывании веб-приложения с помощью файла `.xml` конфигурации контекста.

Файл `.xml` конфигурации контекста может содержать действительный XML для контекста веб-приложения, точно так же, как если бы он был сконфигурирован в вашем файле конфигурации Axiom Libercat `server.xml`. Пример:

```
<Context path="/foobar" docBase="/path/to/application/foobar">
</Context>
```

Если необязательный параметр `war` устанавливается в URL для файла `.war` или каталога веб-приложения, то он переопределяет любой `docBase`, сконфигурированный в файле `.xml` конфигурации контекста.

Здесь приведен пример развертывания приложения с помощью файла «.xml» конфигурации контекста.

```
http://localhost:8080/manager/text/deploy?config=file:/path/context.xml
```

Здесь приведен пример развертывания приложения с помощью файла `.xml` конфигурации контекста и файла «.war» веб-приложения, находящегося на сервере.

```
http://localhost:8080/manager/text/deploy
?config=file:/path/context.xml&war=file:/path/bar.war
```

## Примечания к развертыванию

Если Host сконфигурирован с `unpackWARs=true` и вы развертываете war-файл, архив war будет

распакован в каталог в вашем каталоге Host appBase.

Если архив war или каталог приложения установлен в вашем каталоге Host appBase, и либо Host сконфигурирован с `autoDeploy=true`, либо путь контекста должен соответствовать имени каталога или имени war-файла без расширения `.war`.

Для обеспечения безопасности, когда приложениями могут управлять недоверенные пользователи, флаг Host deployXML может быть установлен в значение «ложь». Это не позволяет недоверенным пользователям развертывать веб-приложения с помощью XML-файла конфигурации, а также не позволяет им развертывать каталоги приложений или файлы «.war», находящиеся вне их Host appBase.

## Ответ на развертывание

Если установка и запуск завершились успешно, вы получите следующий ответ:

```
OK - Deployed application at context path /foo
```

В противном случае ответ будет начинаться с `FAIL` и будет включать в себя сообщение об ошибке. Возможные причины проблем включают:

- *Приложение уже существует по пути /foo*

Контекстные пути должны быть уникальны для всех запущенных в данный момент веб-приложений. Таким образом, вы должны свернуть существующее веб-приложение при помощи этого контекстного пути или выбрать другой путь для нового. Параметр `update` может быть задан как параметр URL, со значением `true`, чтобы избежать этой ошибки. В этом случае свертывание будет реализовано на существующем приложении до развертывания.

- *База документов не существует или не является читаемым каталогом*

URL, указанный параметром `war`, должен определять каталог на этом сервере, который содержит «нераспакованную» версию веб-приложения или абсолютный URL файла архива веб-приложения (WAR), который содержит это приложение. Исправьте значение, указанное параметром `war`.

- *Возможные исключения*

При запуске нового веб-приложения возникло исключение. Для просмотра подробностей просмотрите журнал Axiom Libercat, но объяснения проблемы скорее всего содержатся в `/WEB-INF/web.xml` файле, или найдены отсутствующие классы при инициализации слушателей событий и фильтров приложения.

- *Был указан неверный URL приложения*

URL, указанный вами для каталога или веб-приложения, был недействительным. Такие URL должны начинаться с `file:`, а URL для WAR-файла должны оканчиваться расширением `.war`.

- *Указан неверный контекстный путь*

Контекстный путь должен начинаться с символа `«/»`. Для ссылки на КОРНЕВОЕ веб-приложение используйте `«/»`.

- *Контекстный путь должен соответствовать каталогу или имени WAR-файла:*

Если архив war или каталог приложения установлен в вашем каталоге Host appBase, и либо Host сконфигурирован с `autoDeploy=true`, либо путь контекста должен соответствовать имени каталога или имя war-файла без расширения `.war`.

- *Могут быть установлены только веб-приложения из каталога веб-приложений Host*

Если флаг Host deployXML установлен в значение «ложь», то эта ошибка возникнет, если будет сделана попытка развернуть каталог веб-приложений или файл `.war` за пределами каталога Host appBase.

## Текущий перечень развернутых приложений

`http://localhost:8080/manager/text/list`

Перечислите контекстные пути, текущий статус (`running` или `stopped`) и число активных сессий для всех развернутых в данный момент веб-приложений. Типовой ответ сразу после запуска Axiom Libercat может выглядеть таким образом:

```
OK - Listed applications for virtual host localhost
/webdav:running:0:webdav
/examples:running:0:examples
/manager:running:0:manager
/:running:0:ROOT
/test:running:0:test##2
/test:running:0:test##1
```

## Перезагрузка существующего приложения

`http://localhost:8080/manager/text/reload?path=/examples`

Дайте сигнал существующему приложению закрыться и перезагрузиться. Это может быть полезным, когда контекст веб-приложения не является перезагружаемым и вы обновили классы

или файлы свойств в каталоге `/WEB-INF/classes` или когда вы добавили или обновили файлы `jar` в каталоге `/WEB-INF/lib`.

Если команда выполнена успешно, вы получите следующий ответ:

```
OK - Reloaded application at context path /examples
```

В противном случае ответ будет начинаться с `FAIL` и будет включать в себя сообщение об ошибке. Возможные причины проблем включают:

- *Возможные исключения*

При перезапуске веб-приложения возникло исключение. Просмотрите журналы Axiom Libercat, чтобы узнать подробности.

- *Указан неверный контекстный путь*

Контекстный путь должен начинаться с символа `«/»`. Для ссылки на КОРНЕВОЕ веб-приложение используйте `«/»`.

- *Контекст не существует для пути /foo*

По указанному вами контекстному пути нет развернутого приложения.

- *Не указан контекстный путь*

Требуется параметр `path`.

- *На WAR, развернутом на пути /foo, перезагрузка не поддерживается*

В настоящее время перезагрузка приложения (для подхвата изменений в классах или файле `web.xml`) не поддерживается, если веб-приложение развертывается прямо из WAR-файла. Она работает только тогда, когда веб-приложение развертывается из нераспакованного каталога. Если вы используете WAR-файл, вам следует `undeploy`, а затем `deploy` или `deploy` с параметром `update`, чтобы приложение снова подхватило ваши изменения.

## Перечень свойств ОС и JVM

```
http://localhost:8080/manager/text/serverinfo
```

Перечисляет информацию о версии Axiom Libercat, ОС и свойствах JVM.

В случае возникновения ошибки ответ начинается с `FAIL` и будет включать в себя сообщение об ошибке. Возможные причины проблем включают:

- *Возможные исключения*

При попытке нумерации системных свойств возникло исключение. Просмотрите журналы Axiom Libercat, чтобы узнать подробности.

## Перечень доступных глобальных ресурсов JNDI

```
http://localhost:8080/manager/text/resources[?type=xxxxx]
```

Вывод глобальных ресурсов JNDI, доступных для использования в ссылках ресурсов для файлов контекстной конфигурации. Если вы укажете параметр запроса `type`, значением должно быть полностью квалифицированное имя класса Java типа ресурса, в котором вы заинтересованы (например, вы указали бы `javax.sql.DataSource`, чтобы получить имена доступных источников данных JDBC). Если вы не укажете параметр запроса `type`, будут возвращены ресурсы всех типов.

В зависимости от того, указан ли параметр запроса `type`, первая строка нормального ответа будет такой:

```
OK - Listed global resources of all types
```

или

```
OK - Listed global resources of type xxxxx
```

за которой последуют по одной строке для каждого ресурса. Каждая строка состоит из полей, разделенных двоеточием («:»), следующим образом:

- *Global Resource Name* – имя этого глобального ресурса JNDI, который будет использоваться в атрибуте `global` элемента `<ResourceLink>`.
- *Global Resource Type* – полностью квалифицированное имя класса Java этого глобального ресурса JNDI.

В случае возникновения ошибки ответ начинается с `FAIL` и будет включать в себя сообщение об ошибке. Возможные причины проблем включают:

- *Возможные исключения*

При попытке перечисления глобальных ресурсов JNDI возникло исключение. Просмотрите журналы Axiom Libercat, чтобы узнать подробности.

- *Нет доступных глобальных ресурсов JNDI*

Сервер Axiom Libercat, на котором вы работаете, был сконфигурирован без глобальных

ресурсов JNDI.

## Статистика сессии

```
http://localhost:8080/manager/text/sessions?path=/examples
```

Вывод таймаута сессии по умолчанию для веб-приложения и число активных в данный момент сессий, которые попадают в одноминутные диапазоны их фактического времени таймаута. Например, после перезапуска Axiom Libercat и последующего выполнения одной из выборок JSP в веб-приложении `/examples` вы можете получить нечто вроде этого:

```
OK - Session information for application at context path /examples
Default maximum session inactive interval 30 minutes
<1 minutes: 1 sessions
1 - <2 minutes: 1 sessions
```

## Истекшие сессии

```
http://localhost:8080/manager/text/expire?path=/examples&idle=num
```

Вывод статистики сессий (наподобие вышеприведенной команды `/sessions`) и закрытие в связи с истечением срока сессий, которые бездействуют дольше, чем `num` минут. Чтобы закончить в связи с истечением срока все сессии, используйте `&idle=0`.

```
OK - Session information for application at context path /examples
Default maximum session inactive interval 30 minutes
1 - <2 minutes: 1 sessions
3 - <4 minutes: 1 sessions
>0 minutes: 2 sessions were expired
```

Фактически `/sessions` и `/expire` являются синонимами для одной и той же команды. Разница заключается в присутствии параметра `idle`.

## Запуск существующего приложения

```
http://localhost:8080/manager/text/start?path=/examples
```

Подача сигнала остановленному приложению перезапуститься и снова сделаться доступным. Остановка и пуск полезны, например, если требуемая вашему приложению база данных становится временно недоступной. Обычно лучше остановить веб-приложение, которое

полагается на эту базу данных, чем позволить пользователям непрерывно сталкиваться с исключениями базы данных.

Если команда выполнена успешно, вы получите следующий ответ:

```
OK - Started application at context path /examples
```

В противном случае ответ будет начинаться с `FAIL` и будет включать в себя сообщение об ошибке. Возможные причины проблем включают:

- *Возможные исключения*

При попытке запуска веб-приложения возникло исключение. Просмотрите журналы Axiom Libercat, чтобы узнать подробности.

- *Указан неверный контекстный путь*

Контекстный путь должен начинаться с символа «/». Для ссылки на КОРНЕВОЕ веб-приложение используйте «/».

- *Контекст не существует для пути /foo*

По указанному вами контекстному пути нет развернутого приложения.

- *Не указан контекстный путь*

Требуется параметр `path`.

## Остановка существующего приложения

```
http://localhost:8080/manager/text/stop?path=/examples
```

Подача сигнала существующему приложению сделаться недоступным, но оставаться развернутым. Каждый запрос, поступающий, пока приложение остановлено, увидит ошибку HTTP 404, и это приложение будет выглядеть как «остановленное» по команде перечисления приложений.

Если команда выполнена успешно, вы получите следующий ответ:

```
OK - Stopped application at context path /examples
```

В противном случае ответ будет начинаться с `FAIL` и будет включать в себя сообщение об ошибке. Возможные причины проблем включают:

- *Возможные исключения*

При попытке остановить веб-приложение возникло исключение. Просмотрите журналы Axiom Libercat, чтобы узнать подробности.

- *Указан неверный контекстный путь*

Контекстный путь должен начинаться с символа «/». Для ссылки на КОРНЕВОЕ веб-приложение используйте «/».

- *Контекст не существует для пути /foo*

По указанному вами контекстному пути нет развернутого приложения.

- *Не указан контекстный путь* Требуется параметр `path`.

## Отмена развертывания существующего приложения

```
http://localhost:8080/manager/text/undeploy?path=/examples
```



### Важно:

Эта команда удалит все артефакты веб-приложений, которые существуют в каталоге `appBase` (обычно это каталог "webapps") для этого виртуального хоста. Это удалит `.WAR` приложения, если таковые присутствуют, каталоги приложений, получающиеся либо из развертывания в распакованную форму, либо из расширения `.WAR`, так же как и определение контекста XML из каталога `$CATALINA_BASE/conf/[enginename]/[hostname]/`. Если вы просто хотите остановить приложение, вам следует вместо этого использовать команду `/stop`.

Подача сигнала существующему приложению мягко закрыться и удалить его из Axiom Libercat (что также делает этот контекстный путь доступным для последующего использования). Кроме того, удаляется корневой каталог документа, если он существует в каталоге `appBase` (обычно «webapps») для этого виртуального хоста. Эта команда логически противоположна `/deploy`.

Если команда выполнена успешно, вы получите следующий ответ:

```
OK - Undeployed application at context path /examples
```

В противном случае ответ будет начинаться с `FAIL` и будет включать в себя сообщение об ошибке. Возможные причины проблем включают:



- *Возможные исключения*

При попытке свернуть веб-приложение возникло исключение. Просмотрите журналы Axiom Libercat, чтобы узнать подробности.

- *Указан неверный контекстный путь*

Контекстный путь должен начинаться с символа «/». Для ссылки на КОРНЕВОЕ веб-приложение используйте «/».

- *Не существует контекста с именем /foo*

Нет развернутого приложения с указанным именем.

- *Не указан контекстный путь* Требуется параметр `path`.

## Обнаружение утечек памяти

```
http://localhost:8080/manager/text/findleaks[?statusLine=[true|false]]
```



### Примечание:

Данная диагностика запускает полную сборку мусора. Ее следует использовать с особой осторожностью в производственных системах.

Диагностика обнаружения утечек пытается выявить веб-приложения, которые вызвали утечки памяти, когда их останавливали, перезагружали или свортывали. Результаты должны всегда подтверждаться профилировщиком. Диагностика использует дополнительные функциональные возможности, предоставляемые реализацией StandardHost. Это будет работать, если используется настраиваемый хост, который не расширяет StandardHost.

Запуск явным образом полного сбора мусора из кода Java получает документальное подтверждение ненадежного. Более того, в зависимости от используемой JVM, имеются варианты, чтобы отключить явный запуск GC, например, `-XX:+DisableExplicitGC`. Если вы захотите убедиться, что диагностика успешно прогоняла полный GC, вам понадобится выполнить проверку с помощью таких инструментов, как GC logging, JConsole или аналогичных.

Если команда выполнена успешно, вы получите следующий ответ:

```
/leaking-webapp
```

Если вы захотите увидеть строку статуса включенной в ответ, тогда включите параметр запроса

`statusLine` в запрос со значением `true`.

Весь контекстный путь для веб-приложения, которое было остановлено, перезагружено или свернуто, но чьи классы из предыдущих выполнений остаются загруженными в память, тем самым вызывая утечку памяти, будет указан на новой строке. Если приложение было перезагружено несколько раз, оно может быть указано несколько раз.

В случае неудачного выполнения команды ответ будет начинаться с `FAIL` и будет включать в себя сообщение об ошибке.

## Информация шифра SSL/TLS коннектора

`http://localhost:8080/manager/text/sslConnectorCiphers`

Диагностика коннекторов/шифров SSL перечисляет шифры SSL/TLS, которые в данный момент сконфигурированы для каждого коннектора. Для NIO и NIO2 перечисляются имена отдельных комбинаций шифров. Для APR возвращается значение `SSLCipherSuite`.

Ответ будет выглядеть примерно так:

```
OK - Connector / SSL Cipher information
Connector[HTTP/1.1-8080]
  SSL is not enabled for this connector
Connector[HTTP/1.1-8443]
  TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
  TLS_DHE_RSA_WITH_AES_128_CBC_SHA
  TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
  TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
  ...
```

## Информация цепочки сертификатов SSL/TLS коннектора

`http://localhost:8080/manager/text/sslConnectorCerts`

Диагностика коннекторов/шифров SSL указывает цепочку сертификатов, которая в данный момент сконфигурирована для каждого виртуального хоста.

Ответ будет выглядеть примерно так:

```
OK - Connector / Certificate Chain information
Connector[HTTP/1.1-8080]
  SSL is not enabled for this connector
```

```
Connector[HTTP/1.1-8443]-_default_-RSA
[
[
  Version: V3
  Subject: CN=localhost, OU=Axiom Libercat PMC, O=The Apache Software
Foundation, L=Wakefield, ST=MA, C=US
  Signature Algorithm: SHA256withRSA, OID = 1.2.840.113549.1.1.11
  ...
```

## Информация доверенного сертификата SSL/TLS коннектора

<http://localhost:8080/manager/text/sslConnectorTrustedCerts>

Диагностика коннекторов/сертификатов SSL указывает доверенные сертификаты, которые в данный момент сконфигурированы для каждого виртуального хоста.

Ответ будет выглядеть примерно так:

```
OK - Connector / Trusted Certificate information
Connector[HTTP/1.1-8080]
SSL is not enabled for this connector
Connector[HTTP/1.1-8443]-_default_
[
[
  Version: V3
  Subject: CN=Axiom Libercat Test CA, OU=Axiom Libercat PMC, O=The Apache
Software Foundation, L=Wakefield, ST=MA, C=US
  ...
```

## Перезагрузка конфигурации TLS

<http://localhost:8080/manager/text/sslReload?tlsHostName=name>

Перезагрузите файлы конфигурации TLS (перезагружаются файлы сертификатов и ключей, но не запускается повторный синтаксический анализ `server.xml`). Чтобы перезагрузить файлы для всех хостов, не указывайте параметр `tlsHostName`.

```
OK - Reloaded TLS configuration for [_default_]
```

## Дамп потоков

`http://localhost:8080/manager/text/threaddump`

Вывод дампа потока JVM.

Ответ будет выглядеть примерно так:

```
OK - JVM thread dump
2014-12-08 07:24:40.080
Full thread dump Java HotSpot(TM) Client VM (25.25-b02 mixed mode):

"http-nio-8080-exec-2" Id=26 cpu=46800300 ns usr=46800300 ns blocked 0
for -1 ms waited 0 for -1 ms
    java.lang.Thread.State: RUNNABLE
        locks java.util.concurrent.ThreadPoolExecutor$Worker@1738ad4
        at sun.management.ThreadImpl.dumpThreads0(Native Method)
        at sun.management.ThreadImpl.dumpAllThreads(ThreadImpl.java:446)
        at
org.apache.tomcat.util.Diagnostics.getThreadDump(Diagnostics.java:440)
        at
org.apache.tomcat.util.Diagnostics.getThreadDump(Diagnostics.java:409)
        at
org.apache.catalina.manager.ManagerServlet.threadDump(ManagerServlet.java:557
)
        at
org.apache.catalina.manager.ManagerServlet.doGet(ManagerServlet.java:371)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:618)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:725)
    ...
```

## Информация о виртуальной машине

`http://localhost:8080/manager/text/vminfo`

Вывод некоторой диагностической информации о виртуальной машине Java.

Ответ будет выглядеть примерно так:

```
OK - VM info
2014-12-08 07:27:32.578
Runtime information:
```

```
vmName: Java HotSpot(TM) Client VM
vmVersion: 25.25-b02
vmVendor: Oracle Corporation
specName: Java Virtual Machine Specification
specVersion: 1.8
specVendor: Oracle Corporation
managementSpecVersion: 1.2
name: ...
startTime: 1418012458849
uptime: 393855
isBootClassPathSupported: true

OS information:
...
```

## Сохранение конфигурации

```
http://localhost:8080/manager/text/save
```

Если эта команда указана без всяких параметров, она сохраняет текущую конфигурацию сервера в `server.xml`. Существующий файл при необходимости переименовывается как резервная копия.

Если указана с параметром `path`, который соответствует пути развернутого веб-приложения, тогда конфигурация этого веб-приложения будет сохранена в файле `context.xml` с соответствующим именем в `xmlBase` для текущего Хоста.

Чтобы использовать эту команду, должен присутствовать `StoreConfig` MBean. Обычно это конфигурируется с помощью [StoreConfigLifecycleListener](#).

В случае неудачного выполнения команды ответ будет начинаться с `FAIL` и будет включать в себя сообщение об ошибке.

## Статус сервера

Из следующих ссылок вы можете просмотреть информацию о статусе сервера. Любая из ролей **manager-xxx** позволяет получить доступ к этой странице.

Следующая команда отображает информацию о статусе сервера в формате HTML.

```
http://localhost:8080/manager/status
http://localhost:8080/manager/status/all
```

Команда ниже отображает информацию о статусе сервера в формате XML.

```
http://localhost:8080/manager/status?XML=true  
http://localhost:8080/manager/status/all?XML=true
```

Выводится номер версии сервера и JVM, поставщик JVM, имя и номер ОС, за которыми следует тип архитектуры. Также, имеется информация об использовании памяти JVM и информация о Axiom Libercat AJP и HTTP-коннекторах.

- Информация о потоках - Максимальное число потоков, минимальное и максимальное число запасных потоков, количество потоков и текущее количество занятых потоков.
- Информация о запросах - Максимальное время обработки и время обработки, количество запросов и ошибок, количество полученных и отправленных байтов.
- Таблица, показывающая Stage (Стадия), Time (Время), Bytes Sent (количество отправленных байтов), Bytes Receive (прием байтов), Client (Клиент), VHost (виртуальный хост) и Request (Запрос). В таблице перечислены все существующие запросы. Вот список возможных стадий потока:
  - *«Parse and Prepare Request» («Анализ и подготовка запроса»)* - Заголовки запросов анализируются или имеет место необходимая подготовка к чтению тела запроса (если было указано кодирование передачи).
  - *«Service» («Обслуживание»)* - Поток обрабатывает запрос и генерирует ответ. Эта стадия следует за стадией «Parse and Prepare Request» и предшествует стадии «Finishing». В этой стадии всегда находится хотя бы один поток (страница статуса сервера).
  - *«Finishing» («Завершение»)* - Конец обработки запроса. Любой остаток ответа, все еще находящийся в выходном буфере, отправляется клиенту. За этой стадией следует «Keep-Alive», если целесообразно сохранять соединение активным, или «Ready», если «Keep-Alive» не является целесообразным.
  - *«Keep-Alive» («Проверяемое на активность»)* - Поток сохраняет соединение открытым для клиента на случай, если клиент отправит еще один запрос. Если будет получен еще один запрос, следующей стадией будет «Parse and Prepare Request». Если запрос не будет получен до того, как истечет время сохранения активным, соединение будет закрыто, и следующей стадией будет «Ready».
  - *«Ready» («Готовность»)* - Поток находится в состоянии покоя и готовности к использованию.

Если вы используете команду `/status/all`, будет доступна дополнительная информация по

каждому из развернутых веб-приложений.

## Использование прокси-сервлета JMX

### Что такое прокси-сервлет JMX

JMX Proxy Servlet – это облегченный прокси-сервер для получения и установки внутренних компонентов tomcat. (Или любой класс, который был экспонирован через MBean-компонент). Его использование не очень удобно для пользователя, но пользовательский интерфейс чрезвычайно полезен для встраивания скриптов командной строки в целях мониторинга и изменения внутренних компонентов tomcat. При помощи этого прокси-сервера вы можете делать две вещи: получать информацию и задавать информацию. Чтобы хорошо понять JMX Proxy Servlet, вам нужно иметь общее понимание JMX. Если вы не будете знать, что такое JMX, тогда приготовьтесь к тому, что вы запутаетесь.

### Команда JMX Query

Команда имеет следующий синтаксис:

```
http://webserver/manager/jmxproxy/?qry=STUFF
```

Где **STUFF** – запрос JMX, который вы хотите выполнить. Например, вот несколько запросов, которые вы можете захотеть выполнить:

- `qry=%3Atype%3DRequestProcessor%2C` → `type=RequestProcessor` - определяет местоположение всех воркеров, которые могут обрабатывать запросы и сообщать об их состоянии.
- `qry=%3Aj2eeType=Servlet%2c` → `j2eeType=Servlet` - возвращает все загруженные сервлеты.
- `qry=Catalina%3Atype%3DEnvironment%2Cresourcetype%3DGlobal%2Cname%3DsimpleValue` → `Catalina:type=Environment,resourcetype=Global,name=simpleValue` - ищет конкретный MBean по заданному имени.

Вам нужно будет поэкспериментировать с этим, чтобы действительно понять его возможности. Если вы не предоставите параметра `qry`, тогда будут показаны все компоненты MBean. Мы действительно рекомендуем посмотреть исходный код tomcat и понять спецификацию JMX, чтобы получить лучшее понимание всех запросов, которые вы можете выполнять.

## Команда JMX Get

JMXProxyServlet также поддерживает команду «get», которую можно использовать для получения значения конкретного атрибута MBean. Общий вид команды `get`:

```
http://webserver/manager/jmxproxy/?get=BEANNAME&att=MYATTRIBUTE&key=MYKEY
```

Вы должны предоставить следующие параметры:

1. `get` - Полное имя bean-компонента:
2. `att` - Атрибут, который вы хотите извлечь
3. `key` - (необязательный) Ключ к атрибуту CompositeData MBean

Если команда прошла успешно, ответом будет ОК, в ином случае будет показано сообщение об ошибке. Например, для извлечения текущих данных из heap-памяти (динамически распределяемой памяти приложения), выполните следующую команду:

```
http://webserver/manager/jmxproxy/?get=java.lang:type=Memory&att=HeapMemoryUsage
```

Для вывода только «использованного» ключа:

```
http://webserver/manager/jmxproxy/  
?get=java.lang:type=Memory&att=HeapMemoryUsage&key=used
```

## Команда JMX Set

Теперь, когда вы можете опрашивать MBean, можно заняться внутренними компонентами Axiom Libercat. Общий вид команды `set` (установить):

```
http://webserver/manager/jmxproxy/?set=BEANNAME&att=MYATTRIBUTE&val=NEWVALUE
```

Необходимо предоставить 3 параметра запроса:

1. `set` - Полное имя bean-компонента:
2. `att` - Атрибут, который вы хотите изменить
3. `val` - Новое значение

Если команда прошла успешно, ответом будет ОК, в ином случае будет показано сообщение об ошибке. Например, если вы хотите начать отладку «на лету» для `ErrorReportValve`. Следующая



команда установит отладку в 10.

```
http://localhost:8080/manager/jmxproxy/  
?set=Catalina%3Atype%3DValve%2Cname%3DErrorReportValve%2Chost%3Dlocalhost  
&att=debug&val=10
```

результат – (YMMV):

```
Result: ok
```

Вот что вы увидите, если подадите невалидное значение. В следующем примере мы пытаемся установить отладку равной «cow»:

```
http://localhost:8080/manager/jmxproxy/  
?set=Catalina%3Atype%3DValve%2Cname%3DErrorReportValve%2Chost%3Dlocalhost  
&att=debug&val=cow
```

Результат отладки с невалидным значением

```
Error: java.lang.NumberFormatException: For input string: "cow"
```

## Команда JMX Invoke

Команда `invoke` позволяет вызывать методы в MBeans. Общий вид команды:

```
http://webserver/manager/jmxproxy/?invoke=BEANNAME&op=METHODNAME&ps=COMMASEPARATEDPARAMETERS
```

Например, чтобы вызвать метод `findConnectors()` метода **Service**, используйте:

```
http://localhost:8080/manager/jmxproxy/?invoke=Catalina%3Atype%3DService&op=findConnectors&ps=
```

## Выполнение команд диспетчера с помощью Ant

Вдобавок к способности исполнять команды приложения Manager через HTTP-запросы, как показано выше, Axiom Libercat содержит удобный набор определений Задач (Task) для инструмента сборки *Ant* (версии 1.4 или более поздней). Чтобы использовать эти команды, вам следует выполнить следующие установочные операции:

- Загрузить двоичный дистрибутив Ant с <https://ant.apache.org>. Используйте версию **1.4** или более позднюю.
- Установить дистрибутив Ant в удобный каталог (называемый ANT\_HOME в оставшейся части этих инструкций).
- Добавьте путь \$ANT\_HOME/bin в свою переменную среды PATH.
- Сконфигурировать хотя бы одну комбинацию имени пользователя/пароля в своей базе данных пользователей контейнера Axiom Libercat, которая содержит роль `manager-script`.

Чтобы использовать настраиваемые задачи внутри Ant, вам следует сначала объявить их с элементом `<import>`. Поэтому ваш файл `build.xml` может выглядеть так:

```
<project name="My Application" default="compile" basedir=". ">

  <!-- Configure the directory into which the web application is built -->
  <property name="build"      value="${basedir}/build"/>

  <!-- Configure the context path for this application -->
  <property name="path"       value="/myapp"/>

  <!-- Configure properties to access the Manager application -->
  <property name="url"        value="http://localhost:8080/manager/text"/>
  <property name="username"   value="myusername"/>
  <property name="password"   value="mypassword"/>

  <!-- Configure the path to the Axiom Libercat installation -->
  <property name="catalina.home" value="/usr/local/apache-tomcat"/>

  <!-- Configure the custom Ant tasks for the Manager application -->
  <import file="${catalina.home}/bin/catalina-tasks.xml"/>

  <!-- Executable Targets -->
  <target name="compile" description="Compile web application">
    <!-- ... construct web application in ${build} subdirectory, and
        generated a ${path}.war ... -->
  </target>

  <target name="deploy" description="Install web application"
    depends="compile">
    <deploy url="${url}" username="${username}" password="${password}"
      path="${path}" war="file:${build}${path}.war"/>
  </target>
```

```
<target name="reload" description="Reload web application"
        depends="compile">
    <reload url="${url}" username="${username}" password="${password}"
           path="${path}"/>
</target>

<target name="undeploy" description="Remove web application">
    <undeploy url="${url}" username="${username}" password="${password}"
             path="${path}"/>
</target>

</project>
```



#### Примечание:

Определение задач ресурсов через вышеуказанный импорт переопределит тип данных ресурсов, добавленный в Ant 1.7. Если вы хотите использовать тип данных ресурсов, вам нужно будет использовать поддержку пространства имен Ant, чтобы модифицировать `catalina-tasks.xml`, чтобы поставить задачи Axiom Libercat в соответствие с их собственным пространством имен.

Теперь вы можете исполнять такие команды, как `ant deploy`, чтобы развернуть приложение в работающем экземпляре Axiom Libercat, или с помощью `ant reload` дать указание Axiom Libercat перезагрузить его. Обратите также внимание, что наиболее интересные значения в этом файле `build.xml` определяются как заменяемые свойства, чтобы вы могли переопределить их значения из командной строки. Например, вы можете посчитать рискованным для безопасности включить реальный пароль диспетчера в свой исходный код файла `build.xml`. Чтобы избежать этого, пропустите свойство пароля и укажите его из командной строки следующим образом:

```
ant -Dpassword=secret deploy
```

## Отражение результатов выполнения задачи

Используя *Ant* версии **1.6.2** или более поздней, задачи Catalina предоставляют возможность фиксировать их результат в свойствах или внешних файлах. Они напрямую поддерживают следующее подмножество атрибутов типа `<redirector>`:

| Атрибут  | Описание                                                                                                                                                                                                                                                                                                            | Требуется |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| output   | Имя файла, в который записывать результат. Если поток ошибок не будет также перенаправляться в файл или свойство, он будет появляться в этом результате.                                                                                                                                                            | Нет       |
| error    | Файл, в который должна перенаправляться стандартная ошибка команды.                                                                                                                                                                                                                                                 | Нет       |
| logError | Этот атрибут используется, когда вы хотите видеть результат ошибки в журнале Ant и вы перенаправляете вывод в файл/свойство. Результат ошибки не будет включаться в выходной файл/свойство. Если вы перенаправляете ошибку с атрибутами <i>error</i> или <i>errorProperty</i> , это не окажет никакого воздействия. | Нет       |
| append   | Должны ли файлы вывода и ошибок дополняться или перезаписываться. Значения по умолчанию <i>false</i> .                                                                                                                                                                                                              | Нет       |

| Атрибут          | Описание                                                                                                                                                                            | Требуется |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| createemptyfiles | Должны ли файлы вывода и ошибок создаваться, даже если они пустые. Значения по умолчанию <code>true</code> .                                                                        | Нет       |
| outputproperty   | Имя свойства, в котором должен храниться результат команды. Если только поток ошибок не перенаправляется в отдельный файл или поток, это свойство будет содержать результат ошибки. | Нет       |
| errorproperty    | Имя свойства, в котором должна храниться стандартная ошибка команды.                                                                                                                | Нет       |

Также можно определить пару дополнительных атрибутов:

| Атрибут     | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Требуется |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| alwaysLog   | Этот атрибут используется, когда вы хотите увидеть результаты, которые фиксируете, появляющиеся также в журнале Ant. Его не следует использовать, если только вы не фиксируете результат задачи. Значения по умолчанию <code>false</code> . <i>Этот атрибут будет напрямую поддерживаться &lt;redirector&gt; в Ant 1.6.3</i>                                                                                                                                                                                                                   | Нет       |
| failonerror | Этот атрибут используется, когда вы хотите избежать прекращения выполнения ant при любой ошибке обработки команды диспетчера. Значения по умолчанию <code>true</code> . Значение должно быть установлено на <code>false</code> , если вы хотите фиксировать результат ошибки, иначе выполнение будет прекращаться до того, как что-нибудь можно будет зафиксировать. Этот атрибут воздействует только на исполнение команды диспетчера; любой неправильный или отсутствующий атрибут по-прежнему будет приводить к прекращению выполнения Ant. | Нет       |

Задачи также поддерживают встроенный элемент `<redirector>`, в котором вы можете указать

его полный набор атрибутов, кроме `input`, `inputstring` и `inputencoding`, которые даже будучи принятыми не используются, поскольку не имеют значения в этом контексте. Обратитесь к [ant manual](#) за подробными сведениями об `<redirector>` атрибутах элемента.

Здесь приведен пример выдержки из файла сборки, который показывает, как можно использовать эту поддержку перенаправления результатов:

```
<target name="manager.deploy"
  depends="context.status"
  if="context.notInstalled">
  <deploy url="${mgr.url}"
    username="${mgr.username}"
    password="${mgr.password}"
    path="${mgr.context.path}"
    config="${mgr.context.descriptor}"/>
</target>

<target name="manager.deploy.war"
  depends="context.status"
  if="context.deployable">
  <deploy url="${mgr.url}"
    username="${mgr.username}"
    password="${mgr.password}"
    update="${mgr.update}"
    path="${mgr.context.path}"
    war="${mgr.war.file}"/>
</target>

<target name="context.status">
  <property name="running" value="${mgr.context.path}:running"/>
  <property name="stopped" value="${mgr.context.path}:stopped"/>

  <list url="${mgr.url}"
    outputproperty="ctx.status"
    username="${mgr.username}"
    password="${mgr.password}">
  </list>

  <condition property="context.running">
    <contains string="${ctx.status}" substring="${running}"/>
  </condition>
  <condition property="context.stopped">
    <contains string="${ctx.status}" substring="${stopped}"/>
  </condition>
```

```
<condition property="context.notInstalled">
  <and>
    <isfalse value="${context.running}"/>
    <isfalse value="${context.stopped}"/>
  </and>
</condition>
<condition property="context.deployable">
  <or>
    <istrue value="${context.notInstalled}"/>
    <and>
      <istrue value="${context.running}"/>
      <istrue value="${mgr.update}"/>
    </and>
    <and>
      <istrue value="${context.stopped}"/>
      <istrue value="${mgr.update}"/>
    </and>
  </or>
</condition>
<condition property="context.undeployable">
  <or>
    <istrue value="${context.running}"/>
    <istrue value="${context.stopped}"/>
  </or>
</condition>
</target>
```

### Важное замечание

Плохо организованные цепочки зависимостей задач Ant могут приводить к тому, что задача будет вызываться более одного раза в одном и том же прогоне Ant, даже если это происходит непреднамеренно. Следует проявлять некоторую осторожность, когда вы фиксируете результат задачи, поскольку это может привести к неожиданному результату, например:

- При фиксации в свойстве вы обнаружите в нем результат только *первого* вызова, потому что свойства Ant являются неизменяемыми и однажды установленные не могут быть изменены.
- При фиксации в файле каждый прогон будет его перезаписывать, и вы обнаружите в нем результат только *последнего* вызова, если только вы не используете атрибут `append="true"`, и в этом случае вы увидите добавленный к файлу результат каждого вызова задачи.



## 7. Обновление

Выпущенный продукт в рамках одного выпуска старшей версии находится на поддержке определенное время. В течение этого периода производится исправление обнаруженных в эксплуатации дефектов. После исправления производится либо выпуск планового обновления, содержащего исправление дефекта, либо, при необходимости, дополнительно выпускается критическое исправление. Мы рекомендуем всегда обновлять Axiom Libercat до последней версии в рамках одного выпуска (например, с версии Axiom Libercat 8.0.15 на версию 8.0.16). В общем случае для обновления изделия Axiom Libercat установите новую версию продукта, скопируйте вашу конфигурацию в новую версию и проверьте работу веб приложений в новой версии Axiom Libercat. Если все работает без ошибок, вы можете перейти на новую версию. При возникновении ошибок и дополнительных вопросов, обращайтесь в службу поддержки клиентов Axiom.

### Обновление с помощью программ менеджеров пакетов



#### Важно:

Перед обновлением необходимо сделать резервные копии каталогов `$CATALINA_BASE/conf`, `$CATALINA_BASE/apps`, `$CATALINA_BASE/webapps`. Если для работы сервера использовались дополнительные файлы сценариев в каталоге `$CATALINA_HOME/bin` (например, `$CATALINA_HOME/bin/setenv.sh`, `$CATALINA_HOME/bin/setenv.bat`) - добавьте в резервную копию эти файлы. Если для работы сервера использовались дополнительные библиотеки в каталоге `$CATALINA_HOME/lib` - добавьте в резервную копию эти библиотеки. В результате обновления сервера произойдет полное удаление файлов конфигурации, сценариев запуска, библиотек и установленных приложений.

После создания резервной копии можно запустить установку пакета с более новой версией сервера. Для более подробной информации см. главу [Установка Axiom Libercat](#).

После завершения установки нового сервера, в случае необходимости, скопируйте конфигурационные файлы (каталог `conf`), приложения (каталоги `apps`, `webapps`) из резервной копии в место установки обновленного сервера (`$CATALINA_HOME`).



#### Важно:

Перед копированием конфигурационных файлов необходимо ознакомиться с документом "Примечания к выпуску" для новой версии продукта на предмет наличия возможных

изменений в конфигурационных файлах (может возникнуть ситуация, когда старые конфигурационные файлы не будут работать или будут работать иным образом для нового сервера).

## Обновление в ручном режиме



### Важно:

Перед обновлением рекомендуется сделать резервные копии каталогов `$CATALINA_BASE/bin`, `$CATALINA_BASE/conf`, `$CATALINA_BASE/apps`, `$CATALINA_BASE/webapps`, `$CATALINA_BASE/lib`. Если для работы сервера использовались дополнительные файлы сценариев в каталоге `$CATALINA_HOME/bin` (например, `$CATALINA_HOME/bin/setenv.sh`, `$CATALINA_HOME/bin/setenv.bat`) - добавьте в резервную копию эти файлы. Если для работы сервера использовались дополнительные библиотеки в каталоге `$CATALINA_HOME/lib` - добавьте в резервную копию эти библиотеки.

Обновление в ручном режиме подразумевает копирование содержимого архива с обновленным сервером в каталог `$CATALINA_HOME`.

После завершения процедуры копирования файлов нового сервера, в случае необходимости, скопируйте из резервной копии:

1. файлы сценариев в каталог `$CATALINA_HOME/bin`;
2. дополнительные библиотеки в каталог `$CATALINA_HOME/lib`.

В случае когда переменная окружения `$CATALINA_HOME` совпадает с `$CATALINA_BASE`, дополнительно скопируйте:

1. конфигурационные файлы в каталог `$CATALINA_BASE/conf`;
2. приложения в каталог `$CATALINA_BASE/apps` и/или `$CATALINA_BASE/webapps`.

В случае когда переменная окружения `$CATALINA_HOME` НЕ совпадает с `$CATALINA_BASE`, убедитесь, что в файле `$CATALINA_HOME/bin/setenv.sh` (`$CATALINA_HOME/bin/setenv.bat` для Windows) переменная `$CATALINA_BASE` установлена в правильное значение. Для дополнительной информации см. главу [Базовые настройки Axiom Libercat](#).

**Важно:**

Перед копированием конфигурационных файлов необходимо ознакомиться с документом "Примечания к выпуску" для новой версии продукта на предмет наличия возможных изменений в конфигурационных файлах (может возникнуть ситуация, когда старые конфигурационные файлы не будут работать или будут работать иным образом для нового сервера).

## 8. Развертывание J2EE приложения

### Введение

Axiom Libercat является сервером J2EE приложений и развёртывание таких приложений, как правило, осуществляется в выделенный каталог, который прописывается в файле `conf/tomee.xml`. В конфигурации по умолчанию содержится соответствующая строка, нужно только её раскомментировать.

```
<Deployments dir="apps" />
```

Здесь `apps` имя каталога, который нужно создать аналогично каталогу `webapps` (имя каталога можно задать произвольно).

В отличие от каталога `webapps`, где развёртывание происходит автоматически (при настройках по умолчанию) развёртывание в каталоге `apps` (в настройках по умолчанию) происходит только при старте сервера. Для того чтобы поменять это поведение, нужно добавить атрибут `autoDeploy="true"`:

```
<Deployments dir="apps" autoDeploy="true"/>
```

В каталог для J2EE приложений `apps` можно копировать EAR файлы, а также файлы архивов, которые могут входить в состав EAR приложения, такие как архивы JAR (например клиентское приложение J2EE), модули ресурсов RAR.

### Варианты упаковки приложения

Приложение можно паковать традиционным способом (по спецификации J2EE), но также для развёртывания на сервере Axiom Libercat можно собирать WAR архив и класть в `webapps` каталог. К примеру, в случае с EAR приложением, которое включает в себя архив с EJB (JAR) и архив с веб-приложением (WAR) вместо упаковки с использованием EAR можно упаковать всё одним архивом (этот архив не должен внутри себя содержать подмодулей-архивов) как WAR файл и разворачивать приложение через `webapps`.

При упаковке J2EE приложения одним архивом (WAR) справедливы следующие замечания:

- Используется один и тот же класслоадер для сервлетов и EJB (нет разграничения между загрженным классами)

- EJB и сервлеты могут использовать все те же самые сторонние библиотеки (например Spring)
- EAR не используется
- Дескрипторы `web.xml` и `ejb-jar.xml` лежат в одном и том же архиве (WAR файле)
- EJB компонентам доступны классы сервлетов и наоборот

Необходимо понимать что это такой тип упаковки не является J2EE совместимым. Но это допустимая возможность в Java EE 6.

Для J2EE совместимого приложения выполняются следующие правила:

- Один и тот же класслоадер не может использоваться для EJB компонент и сервлетов.
- Должно быть по меньшей мере 3 класслоадера: один для EAR файла, по одному на каждый `ejb-jar` модуль и по одному на каждый WAR модуль.
- Классы сервлетов имеют доступ к EJB компонентам, но EJB компоненты не имеют доступа к сервлетам. Чтобы поддерживать это J2EE накладывает ограничения на упаковку модулей:
  1. Классы EJB компонент и классы сервлетов не могут находиться в одном модуле/архиве.
  2. Необходимо по меньшей мере 3 архива для возможности использования EJB компонент и сервлетов: один для EAR, один для EJB (JAR) и один для сервлетов (WAR).
  3. Общие библиотеки должны быть помещены в EAR модуль и включены через `Class-Path` запись в манифесте EAR архива/модуля.

## Развёртывание с помощью maven плагина

Axiom Libercat сервер предоставляет возможность удалённого развёртывания через внутреннюю EJB компоненту. Используя служебный вызов удалённой компоненты через HTTP, можно развернуть J2EE приложение удалённо. Такую возможность предоставляет [TomEE Maven Plugin](#). Для этого используется его цель `tomee:deploy`.

Вот пример конфигурации плагина для удалённого развёртывания:

```
<plugin>
  <groupId>org.apache.tomee.maven</groupId>
  <artifactId>tomee-maven-plugin</artifactId>
  <version>8.0.15</version>
  <configuration>
    <tomeeHost>localhost</tomeeHost>
    <tomeeHttpPort>8080</tomeeHttpPort>
```

```
<path>target/path-to-project-ear.ear</path>
<useBinaries>true</useBinaries>
</configuration>
</plugin>
```

Для того чтобы EJB компонента, дающая возможность удалённого развёртывания, работала необходимо поменять настройки по умолчанию в файле конфигурации `conf/system.properties`:

1. Нужно выставить значение свойства `tomee.remote.support` в `true` (в файле уже есть необходимая строка, нужно её просто раскомментировать)
2. Нужно выставить значение свойства `openejb.system.apps` в `true` (в файле уже есть необходимая строка, нужно её просто раскомментировать).

Необходимо обратить внимание на то, что развёртывание подобным образом предназначено в основном для разработчиков, так как приложение кладётся не в каталог для J2EE приложений (`apps`), а во временный каталог (`temp` по умолчанию), где приложение используется только в текущей сессии работы сервера. То есть оно не будет развёрнуто на сервере при его рестарте. Таким образом это удобный инструмент во время разработки для возможности быстрого развёртывания после коррекции кода разработчиком, но не для конечного приложения.

Полезно также отметить что указанные настройки включают вместе с EJB компонентой для удалённого развёртывания также JMX компоненты с именем `openejb.user.mbeans:application=openejb,group=org.apache.openejb.assembler.monitoring,name=JMXDeployer`. Эта компонента даёт возможность провести развёртывание с помощью вызова JMX операции `deploy` (вызов в итоге делегируется в EJB компоненту).

## 9. Развертывание веб-приложения

### Введение

Развертывание – это термин, используемый для описания процесса установки веб-приложения (либо сторонний WAR, либо собственное веб-приложение) на сервер Axiom Libercat.

Развертывание веб-приложения на сервер Axiom Libercat может быть осуществлено различными способами.

- Статически: веб-приложение настраивается до запуска Axiom Libercat.
- Динамически; за счет прямого использования уже развернутых веб-приложений (использование функции *автоматического развертывания*) или удаленно, с использованием веб-приложения Axiom Libercat Manager.

[Руководство по приложению Manager](#) представляет собой веб-приложение, которое можно использовать как интерактивно (через ГПИ HTML), так и программно (через API на основе URL), для развертывания и управления веб-приложениями.

Существует несколько способов развертывания с применением веб-приложения Manager. Axiom Libercat ставит задачи для инструмента сборки Apache Ant. [Axiom Libercat Maven Plugin](#) отвечает за интегрирование с Apache Maven. Также есть инструмент, называемый Client Deployer, который можно использовать через командную строку и который добавляет такие функции, как компилирование и проверка веб-приложений, а также упаковка веб-приложений в файлы ресурсов веб-приложений (WAR).

### Установка

Для статического развертывания веб-приложений установка не требуется, так как Axiom Libercat осуществляет это автоматически. Установка также не требуется для функций развертывания с использованием Axiom Libercat Manager, хотя некоторая конфигурация необходима, как подробно описано в [Руководстве по приложению Manager](#). Однако установка требуется, если вы хотите использовать Axiom Libercat Client Deployer (TCD).

TCD не упакован в дистрибутив ядра Axiom Libercat, поэтому его нужно скачивать отдельно из Downloads. Файл загрузки обычно называется *apache-tomcat-8.5.x-deployer*.

TCD требует установленных Apache Ant 1.6.2 или более поздней и Java. Список системных переменных должен содержать переменную `ANT_HOME`, указывающую на корневую папку вашей

установки Ant, и переменную `JAVA_HOME`, указывающую на вашу установку Java. Кроме того, вы должны убедиться, что команда Ant и команда компилятора `javac` запущены из командной оболочки вашей операционной системы.

1. Загрузите дистрибутив TCD.
2. Распакуйте программный пакет TCD в любое место.
3. Прочитайте [Развертывание с использованием пакета Deployer клиента](#)

## Несколько слов о Контексте

Говоря о развертывании веб-приложений, необходимо понимать концепцию *Context*. Context – так Axiom Libercat называет веб-приложение.

Чтобы настроить Context в Axiom Libercat требуется *Context Descriptor*. Context Descriptor – это простой XML-файл, содержащий конфигурацию Axiom Libercat для Context, например имена ресурсов или конфигурацию диспетчера сессий. В ранних версиях Axiom Libercat содержимое конфигурации Context Descriptor часто хранилось в основном файле конфигурации `Tomcat_server.xml`, но теперь это не рекомендуется (хотя все еще работает).

Context Descriptor не только помогает Axiom Libercat понять, как настроить Context, но и другие инструменты, например, Axiom Libercat Manager и TCD, часто используют Context Descriptor для должного выполнения своих задач.

Расположение Context Descriptor:

1. `$CATALINA_BASE/conf/[enginename]/[hostname]/[webappname].xml`
2. `$CATALINA_BASE/webapps/[webappname]/META-INF/context.xml`

Файлы в (1) названы `[webappname].xml`, а файлы в (2) – `context.xml`. Если Context Descriptor для Context отсутствует, Axiom Libercat настраивает Context с помощью значений по умолчанию.

## Развертывание при запуске Axiom Libercat

Если вы не хотите пользоваться Axiom Libercat Manager или TCD, тогда вы должны осуществлять развертывание своих веб-приложений статически для Axiom Libercat, после запуска Axiom Libercat. Место, куда происходит развертывание веб-приложений для подобного типа развертывания, называется `appBase` и указывается для хоста. Вы либо копируете так называемое *exploded web application*, т.е. несжатое веб-приложение, либо файл `.WAR` со сжатыми ресурсами веб-приложения.



Веб-приложения в этом месте, с указанием атрибута хоста (хост по умолчанию – «localhost») `appBase` (`appBase` по умолчанию – «`$CATALINA_BASE/webapps`»), будет развернуто при запуске Axiom Libercat, только если значение атрибута хоста `deployOnStartup` имеет значение «true».

В этом случае при запуске Axiom Libercat будет следующая последовательность развертывания:

1. Сначала происходит развертывание Context Descriptor.
2. Затем развертываются распакованные веб-приложения, не связанные с Context Descriptor. Если у них есть соответствующий `.WAR`-файл в `appBase` и он новее, чем распакованное веб-приложение, папка распаковки будет перемещена, а веб-приложение будет повторно взято из `.WAR`.
3. Развертываются `.WAR`-файлы.

## Развертывание на действующем сервере Axiom Libercat

Можно проводить развертывание веб-приложений на работающий сервер Axiom Libercat.

Если атрибут хоста `autoDeploy` «true», хост будет пытаться развернуть и обновить веб-приложения динамически, при необходимости, например, если новый `.WAR`-файл скидывается в `appBase`. Чтобы это сработало, хост должен иметь включенную фоновую обработку, которая в конфигурации по умолчанию включена.

`autoDeploy` переводится в «true», и запускается Axiom Libercat, что позволяет:

- Развертывать `.WAR`-файлы, скопированные в хост `appBase`.
- Развертывать распакованные веб-приложения, скопированные в хост `appBase`.
- Повторно развертывать веб-приложения, которые уже были развернуты из `.WAR`-файла, когда предоставляется новый `.WAR`-файл. В этом случае распакованное веб-приложение удаляется, а `.WAR`-файл развертывается снова. Обратите внимание на то, что распаковка не происходит, если хост настроен так, что `.WAR`-файлы не распаковываются с атрибутом `unpackWARs`, установленным как «false», в случае чего веб-приложение будет просто развернуто заново, как сжатый архив.
- Перезагрузка веб-приложения, если файл `/WEB-INF/web.xml` (или любой другой ресурс, определенный как `WatchedResource`) обновлен.
- Повторное развертывание веб-приложения, если файл Context Descriptor, из которого было развернуто веб-приложение, обновлен.
- Повторное развертывание зависимых веб-приложений, если глобальный или локальный для

хоста файл Context Descriptor, используемый веб-приложением, обновлен.

- Повторное развертывание веб-приложения, если файл Context Descriptor (с именем, соответствующим пути Context ранее развернутого веб-приложения) добавлен в каталог `$CATALINA_BASE/conf/[enginename]/[hostname]/`.
- Удаление веб-приложения, если база документа (docBase) удалена. Обратите внимание, что в Windows должны быть включены функции антиблокировки (см. конфигурацию Context), иначе невозможно будет удалить ресурсы запущенных веб-приложений.

Обратите внимание, что перезагрузка веб-приложения может также настраиваться в загрузчике; в этом случае загруженные классы будут проверены на наличие изменений.

## Развертывание с использованием Axiom Libercat Manager

Axiom Libercat Manager описан на своей странице [Libercat-Manager-App-How-To](#).

## Развертывание с использованием пакета Deployer клиента

Также развертывание веб-приложений может производиться с помощью Axiom Libercat Client Deployer. Это программный пакет, который может использоваться для проверки, компилирования, сжатия в .WAR-файл и развертывания веб-приложений для серверов Axiom Libercat для производства или разработки. Необходимо отметить, что эта функция использует Axiom Libercat Manager, а значит, должен работать соответствующий сервер Axiom Libercat.

Предполагается, что для использования TCD пользователь знаком с Apache Ant. Apache Ant – это инструмент сборки с применением скриптов. TCD поставляется в заранее запакованном виде со скриптом сборки. Требуется лишь базовое понимание Apache Ant (установка, как было указано ранее на этой странице, и знание применения командной оболочки операционной системы и настраиваемых переменных среды).

В TCD входят задачи Ant, компилятор страниц Jasper для компилирования JSP до развертывания, а также задача, проверяющая Context Descriptor для веб-приложений. Задача проверки (класс `org.apache.catalina.ant.ValidatorTask`) позволяет работать только с одним параметром: с базовым путем распакованного веб-приложения.

TCD использует распакованное веб-приложение для входных данных (см. список используемых свойств ниже). Веб-приложение, программно развернутое с помощью Deployer, может включать в себя Context Descriptor в `/META-INF/context.xml`.

TCD включает в себя готовый к применению скрипт Ant со следующими задачами:

- `compile` (по умолчанию) - Скомпилировать и проверить веб-приложение. Это можно использовать отдельно, работающий сервер Axiom Libercat не требуется. Скомпилированное приложение работает только на соответствующей версии сервера *Tomcat X.Y.Z* и может не работать на другой версии Axiom Libercat, так как код, генерируемый Jasper, зависит от компонента среды выполнения. Следует отметить, что эта задача также автоматически компилирует все исходные файлы Java, находящиеся в папке `/WEB-INF/classes` веб-приложения.
- `deploy` - Развертывание веб-приложения (скомпилированного или нет) на сервер Axiom Libercat.
- `undeploy` - Удаление веб-приложения.
- `start` - Запуск веб-приложения.
- `reload` - Перезагрузка веб-приложения.
- `stop` - Остановка веб-приложения.

Чтобы настроить развертывание, создайте файл с именем `deployer.properties` в корневой папке установки TCD. В этом файле добавьте следующую пару `name=value` (имя=значение) в строку:

Также нужно убедиться, что пользователь был задан для целевого Axiom Libercat Manager (который использует TCD), иначе TCD не сможет пройти аутентификацию с Axiom Libercat Manager и развертывание закончится неудачей. Для этого см. страницу, посвященную Axiom Libercat Manager.

- `build` - Используемой папкой сборки по умолчанию является `${build}/webapp/${path}` (`${build}`, по умолчанию указывая на `${basedir}/build`). В конце выполнения задачи `compile` WAR-файл веб-приложения будет расположен здесь: `${build}/webapp/${path}.war`.
- `webapp` - Папка, содержащая распакованное веб-приложение, которое будет скомпилировано и проверено. Папка по умолчанию: `myapp`.
- `path` - Путь развернутого context веб-приложения по умолчанию: `/myapp`.
- `url` - Абсолютный URL-адрес веб-приложения Axiom Libercat Manager для запущенного сервера Axiom Libercat, который будет использоваться для развертывания и удаления веб-приложения. По умолчанию `deployer` попытается получить доступ к экземпляру Axiom Libercat, запущенному на `localhost`: `http://localhost:8080/manager/text`.
- `username` - Имя пользователя Axiom Libercat Manager (пользователь должен иметь права `manager-script`).
- `password` - Пароль Axiom Libercat Manager.

# 10. Конфигурация менеджера транзакций

## Введение

Axiom Libercat использует [Apache Geronimo](#) в качестве менеджера транзакций (Transaction Manager). Глобальный менеджер транзакций сервера создаётся один раз во время старта сервера в одном экземпляре. Менеджер транзакций управляет границами и всем жизненным циклом транзакций: созданием, началом, фазами, действиями, окончанием.

## Конфигурация

Менеджер транзакций может быть сконфигурирован либо через основной конфигурационный файл `conf/tomee.xml` со следующим содержимым в качестве примера:

```
<TransactionManager id="myId" type="TransactionManager">
    defaultTransactionTimeout 10 minutes
</TransactionManager>
```

Для настройки менеджера транзакций используются следующие основные параметры:

| Свойство                  | Тип         | Значение по умолчанию                   | Описание                                       |
|---------------------------|-------------|-----------------------------------------|------------------------------------------------|
| defaultTransactionTimeout | время       | 10 minutes                              | Таймаут транзакций по умолчанию.               |
| tmId                      | массив байт | Axiom Libercat генерирует идентификатор | Идентификатор менеджера транзакций (см. ниже). |

|            |        |       |                                                                                                |
|------------|--------|-------|------------------------------------------------------------------------------------------------|
| txRecovery | булево | false | Включает запись журнала транзакций и восстановление незаконченных транзакций по этому журналу. |
|------------|--------|-------|------------------------------------------------------------------------------------------------|

Как показано выше эти свойства задаются либо через свойства в текстовом узле основного конфигурационного файла `tomee.xml` элемента `TransactionManager` либо в файле `system.properties`.

## Восстановление транзакций

Восстановление транзакций автоматически запускается при старте сервера если задано свойство `txRecovery` (установлено в `true`) для транзакций, которые были по каким-то причинам не были корректно завершены. Для того чтобы можно было провести восстановление необходимо записывать журнал транзакций. Свойство `txRecovery` отвечает также и за включение записи журнала транзакций. Менеджер транзакций `Geronimo` использует HOWL журнал для записи данных транзакций в файлы.

Свойства менеджера транзакций, относящиеся к восстановлению.

| Свойство        | Тип    | Значение по умолчанию | Описание                                                                                                                                                 |
|-----------------|--------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| adler32Checksum | булево | true                  | Требует включенного свойства <code>txRecovery</code> . Если включено используется алгоритм Adler-32 для вычисления чексуммы при записи в лог транзакции. |

| Свойство        | Тип         | Значение по умолчанию | Описание                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|-------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bufferSizeKb    | целое число | 32                    | Требуется включенного свойства txRecovery. Задаёт размер буфера в килобайтах, используемые при записи журнала транзакций.                                                                                                                                                                                                                                                                                                           |
| checksumEnabled | булево      | true                  | Требуется включенного свойства txRecovery. Если включено чексуммы считаются от содержимого каждого буфера перед записью содержимого на диск.                                                                                                                                                                                                                                                                                        |
| flushSleepTime  | время       | 50 Milliseconds       | Требуется включенного свойства txRecovery. Время, которое поток ForceManager спит между принуждением журнала к сбросу буферов. Во время периодов низкой активности, потоки могут ждать значительное время (возможно бесконечное) заполнения буферов для сброса. Во избежание этой ситуации функционал журналирования запускает ForceManager поток который периодически просыпается и форсирует операции IO пока другие потоки ждут. |
| logFileDir      | строка      | txlog                 | Требуется включенного свойства txRecovery. Каталог файла журнала транзакций.                                                                                                                                                                                                                                                                                                                                                        |

| Свойство         | Тип         | Значение по умолчанию | Описание                                                                                    |
|------------------|-------------|-----------------------|---------------------------------------------------------------------------------------------|
| logFileExt       | строка      | log                   | Требуется включенного свойства txRecovery. Расширение файла журнала транзакций.             |
| logFileName      | строка      | howl                  | Требуется включенного свойства txRecovery. Имя файла журнала транзакций.                    |
| maxBlocksPerFile | целое число | -1                    | Требуется включенного свойства txRecovery. Максимальное кол-во блоков в файле журнала.      |
| maxBuffers       | целое число | 0                     | Требуется включенного свойства txRecovery. Максимальное количество задействованных буферов. |
| maxLogFiles      | целое число | 2                     | Требуется включенного свойства txRecovery. Максимальное количество файлов журнала.          |
| minBuffers       | целое число | 4                     | Требуется включенного свойства txRecovery. Минимальное количество буферов.                  |

| Свойство                     | Тип         | Значение по умолчанию | Описание                                                                                                                                                                                                       |
|------------------------------|-------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| threadsWaitingForceThreshold | целое число | -1                    | Требует включенного свойства txRecovery. Максимальное количество потоков которое должно ждать форсирования операций IO. Установка значения может оказать эффект на задержку ожидания форсирования операций IO. |
| txRecovery                   | булево      | false                 | Включает запись журнала транзакций и восстановление незаконченных транзакций по этому журналу.                                                                                                                 |

## Установка уникального идентификатора

Реализация менеджера транзакций использует генератор идентификаторов транзакций (*Xid*). Если разные Axiom Libercat сервера задействуют в транзакциях одни и те же ресурсы (такие ресурсы представлены в транзакциях *XAResource*-ами), как, например, база данных, то следует убедиться что идентификаторы транзакций, передаваемые общим ресурсам различны. Так как идентификаторы транзакций генерируются на основе идентификатора менеджера транзакций (и несовпадение идентификаторов менеджера гарантирует несовпадение сгенерированных идентификаторов транзакций), то в таких случаях необходимо убедиться что разные экземпляры, работающие с одними и теми же ресурсами используют разные идентификаторы. По умолчанию Axiom Libercat генерирует идентификатор менеджера транзакций на основании данных сервера, где установлен Axiom Libercat (используется сетевой адрес и некоторая случайная величина). Причём, этот сгенерированный идентификатор сохраняется в файле `work/tmid.store`. Это даёт возможность переиспользовать единожды сгенерированный идентификатор (при рестарте, чтобы не генерировать каждый раз новый идентификатор). Сохранение идентификатора менеджера транзакций между перезапусками экземпляра сервера важно для функциональности восстановления транзакций.



**Примечание:**

Если нет возможности создать файл `work/tmid.store` (например не хватает прав на запись в каталог), то сгенерированный идентификатор не будет записан и он будет регенерирован заново при следующем старте сервера.

Идентификатор менеджера транзакций можно задать непосредственно через его конфигурацию. Для этого нужно использовать свойство `tmId`. Тип значения этого свойства массив байт. Элементы массива пишутся через запятую.

```
<TransactionManager id="myId" type="TransactionManager">  
    tmId = 7, 5, 3, 11  
</TransactionManager>
```

Задание вручную идентификатора позволяет контролировать его значение, не полагаться на его генерацию и избежать вопросов по поводу хранения сгенерированного идентификатора и возможности записи файла хранения. Но это будет неудобно в том случае если такой идентификатор нужно прописывать на множество серверов.

## 11. Как работает lookup

Axiom Libercat предоставляет несколько различных реализаций клиентов для работы со службой имен и каталогов (JNDI).

### Клиент по умолчанию

В случаях, когда нет необходимости выполнять поиск ресурсов по имени, или же поиск ведется для ресурсов зарегистрированных самим приложением, может быть достаточно клиента по умолчанию. В таком случае код инициализации контекста и поиска ресурса мог бы выглядеть так:

```
final Context ctx = new InitialContext();
DataSource dataSource = (DataSource) ctx.lookup("java:comp/env/jdbc");
```

### OpenEJBInitialContextFactory

Клиент `OpenEJBInitialContextFactory` используется для поиска локальных бинов (EJB) и ресурсов контейнера приложения.

```
Properties properties = new Properties();
properties.setProperty(Context.INITIAL_CONTEXT_FACTORY,
    "org.apache.openejb.core.OpenEJBInitialContextFactory");
final Context ctx = new InitialContext(properties);
DataSource dataSource = (DataSource) ctx.lookup("MyDataSource");
```

### RemoteInitialContextFactory

Для поиска бинов, расположенных на удаленном сервере, используется клиент `RemoteInitialContextFactory`. Данному клиенту дополнительно необходимо свойство `Context.PROVIDER_URL`, в котором перечисляются сетевые адреса серверов, среди которых необходимо осуществлять поиск.

```
Properties properties = new Properties();
properties.setProperty(Context.INITIAL_CONTEXT_FACTORY,
    "org.apache.openejb.core.OpenEJBInitialContextFactory");
properties.put(Context.PROVIDER_URL,
    "failover:ejbd://192.168.1.1:4201,ejbd://192.168.1.2:4201");
final Context ctx = new InitialContext(properties);
```

```
MyRemoteEjb ejb = (MyRemoteEjb) ctx.lookup("java:global/app-  
1.0/MyRemoteEjb!com.example.MyRemoteEjb");
```

## Безопасность

Клиенты могут быть дополнительно сконфигурированы для работы с контекстами, требующими аутентификацию

```
p.put("openejb.authentication.realmName", "my-realm"); // необязательный  
параметр  
p.put(Context.SECURITY_PRINCIPAL, "user");  
p.put(Context.SECURITY_CREDENTIALS, "password");
```

## Конфигурирование JNDI провайдера

Axiom Libercat позволяет конфигурировать JNDI провайдер по умолчанию как глобально, на уровне сервера, так и для отдельного EJB модуля.

Конфигурирование JNDI провайдера осуществляется путем установки следующих свойств:

- `java.naming.factory.initial` - класс фабрики контекстов JNDI (см. варианты реализаций клиентов JNDI выше);
- `java.naming.provider.url` - сетевой адрес сервера, на котором осуществляется поиск ресурсов JNDI.

По умолчанию, данные свойства не определены и используется клиент JNDI по умолчанию. OpenEJB предоставляет следующую конфигурацию для JNDI провайдера:

```
<!--  
# =====  
# Default JndiProvider  
# =====  
-->  
<ServiceProvider  
    id="Default Remote Jndi Provider"  
    service="JndiProvider"  
    types="javax.naming.InitialContext"  
    class-name="org.apache.openejb.core.NoProvider">  
# This provider allows the configuration of InitialContext  
# objects for the lookup of resources outside this server  
#
```

```
# The properties specified here go directly to the properties
# object passed into the constructor of the InitialContext
#
# Example properties might look like the following:
#
#     java.naming.factory.initial =
org.apache.openejb.client.RemoteInitialContextFactory
#     java.naming.provider.url = ejbd://localhost:4201
#
</ServiceProvider>
```

## Конфигурирование JNDI провайдера на уровне сервера

В случае конфигурации на уровне сервера необходимые свойства могут быть установлены несколькими способами:

- в конфигурационном файле `$CATALINA_BASE/conf/system.properties`
- с помощью опции командной строки запуска JVM `-D`.
- в конфигурационном файле `$CATALINA_BASE/conf/tomee.xml`, с использованием элемента `JndiProvider`:

```
<JndiProvider id="myInitialContext" type="javax.naming.InitialContext">
    java.naming.provider.url = http://192.168.1.2:8080/tomee/ejb
    java.naming.factory.initial =
org.apache.openejb.client.RemoteInitialContextFactory
</JndiProvider>
```

## Конфигурирование JNDI провайдера на уровне EJB и модуля

Переопределение конфигурации JNDI провайдера на уровне EJB модуля осуществляется путем установки необходимых свойств следующими способами:

- в файле ресурсов приложения

```
<JndiProvider id="myInitialContext" type="javax.naming.InitialContext">
</JndiProvider>
```

- в конфигурационном файле сервисов, `service-jar.xml`, расположенном в каталоге `META-INF` приложения

```
<ServiceProvider
    id="Default Remote Jndi Provider"
    service="JndiProvider"
    types="javax.naming.InitialContext"
    class-name="org.apache.openejb.core.NoProvider">
    java.naming.provider.url = http://192.168.1.2:8080/tomee/ejb
    java.naming.factory.initial =
org.apache.openejb.client.RemoteInitialContextFactory
</ServiceProvider>
```

## Формат JNDI имен бинов

Axiom Libercat предоставляет имена по умолчанию для бинов (EJB) в следующем формате:

```
{deploymentId}{interfaceType.annotationName}
```

Данный формат может быть изменен с помощью установки свойства `openejb.jndiname.format` в файле `$CATALINA_BASE/conf/system.properties` или через опцию командной строки запуска JVM `-D`.

Помимо `предопределенных имен переменных` `deploymentId` and `interfaceType.annotationName` имеется набор дополнительных переменных, которые также могут быть использованы для определения формата JNDI имен:

| Переменная                       | Описание                                                                                                              |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>moduleId</code>            | Как правило, имя EJB jar файла или идентификатор модуля, описанный в файле дескриптора <code>openejb-jar.xml</code> . |
| <code>ejbType</code>             | STATEFUL, STATELESS, BMP_ENTITY, CMP_ENTITY, or MESSAGE_DRIVEN.                                                       |
| <code>ejbClass</code>            | Полное квалифицированное имя класса EJB бина, например, <code>com.example.MyBean</code> .                             |
| <code>ejbClass.simpleName</code> | Простое имя класса EJB бина, например, <code>MyBean</code> .                                                          |

| Переменная                                  | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ejbClass.packageName</code>           | Имя пакета, в котором определен EJB бин, например, <code>com.example</code> .                                                                                                                                                                                                                                                                                                                                                                            |
| <code>ejbName</code>                        | Имя бина, специфицированное в файле дескрипторе <code>ejb-jar.xml</code> или через атрибут <code>name</code> в аннотациях <code>@Stateful</code> , <code>@Stateless</code> , <code>@MessageDriven</code> .                                                                                                                                                                                                                                               |
| <code>deploymentId</code>                   | Уникальный системный идентификатор EJB бина, по умолчанию соответствует значению переменной <code>ejbName</code> . Может быть переопределен через атрибут <code>deployment-id</code> элемента <code>ejb-deployment</code> в файле дескриптора <code>openejb-jar.xml</code> . Также может быть параметризован через свойство <code>openejb.deploymentId.format</code> .                                                                                   |
| <code>interfaceType</code>                  | Смотри <code>interfaceType.annotationName</code> .                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>interfaceType.annotationName</code>   | Значение этого выражения зависит от примененных аннотаций, <ul style="list-style-type: none"><li>• <code>@LocalBean</code> - <code>LocalBean</code></li><li>• <code>@Local</code> - <code>Local</code></li><li>• <code>@Remote</code> - <code>Remote</code></li><li>• <code>@LocalHome</code> - <code>LocalHome</code></li><li>• <code>@RemoteHome</code> - <code>RemoteHome</code></li><li>• <code>@WebService</code> - <code>Endpoint</code></li></ul> |
| <code>interfaceType.annotationNameLC</code> | То же самое, что и выше, но имена в нижнем регистре.                                                                                                                                                                                                                                                                                                                                                                                                     |

| Переменная                      | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| interfaceType.xmlName           | <p>Значение данного параметра зависит от примененных аннотаций и соответствует именами XML тэгов в файле дескрипторе <code>ejb-jar.xml</code>,</p> <ul style="list-style-type: none"><li>• <code>@LocalBean</code> - localbean</li><li>• <code>@Local</code> - business-local</li><li>• <code>@Remote</code> - business-remote</li><li>• <code>@LocalHome</code> - local-home</li><li>• <code>@RemoteHome</code> - home</li><li>• <code>@WebService</code> - service-endpoint</li></ul> |
| interfaceType.xmlNameCc         | <p>То же самое, что и выше, но имена с использованием нотации <code>CamelCase</code>.</p>                                                                                                                                                                                                                                                                                                                                                                                               |
| interfaceType.openejbLegacyName | <p>Значение данного параметра зависит от примененных аннотаций и соответствует формату имен, принятых в OpenEJB 1.0,</p> <ul style="list-style-type: none"><li>• <code>@LocalBean</code> - LocalBean</li><li>• <code>@Local</code> - BusinessLocal</li><li>• <code>@Remote</code> - BusinessRemote</li><li>• <code>@LocalHome</code> - Local</li><li>• <code>@RemoteHome</code> - ""</li><li>• <code>@WebService</code> - ServiceEndpoint</li></ul>                                     |

| Переменная                 | Описание                                                                                                                                                                                                                                                                                         |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| interfaceClass             | Полное квалифицированное имя интерфейса, который реализует данный EJB бин, например, <code>com.example.MyInterface</code> . В случае, если EJB бин не реализует интерфейс, то выражение сводится к полному квалифицированному имени класса EJB бина, например, <code>com.example.MyBean</code> . |
| interfaceClass.simpleName  | Простое имя интерфейса, который реализует данный EJB бин, например, <code>MyInterface</code> . В случае, если EJB бин не реализует интерфейс, то выражение сводится к простому имени класса EJB бина, например, <code>MyBean</code> .                                                            |
| interfaceClass.packageName | Имя пакета, в котором определен интерфейс, который реализует данный EJB бин, например, <code>com.example</code> . В случае, если EJB бин не реализует интерфейс, то выражение сводится к имени пакета в котором определен данный EJB бин.                                                        |

## Переопределение формата JNDI имен

Axiom Libercat позволяет изменять формат JNDI имен на различных уровнях: на уровне сервера, EJB модуля, конкретного EJB бина, на уровне интерфейсов EJB бина. Более специфичный формат переопределяет более общий формат. Например, можно установить общий формат на уровне сервера, переопределить его на уровне EJB модуля и далее на уровне конкретного EJB бина.



### Важно:

Изменение формата JNDI имен может привести к конфликту имен, в случае такого конфликта процедура развертывания EJB модуля прекращается.



## Переопределение через системные свойства

Данный способ подразумевает установку свойства `openejb.jndiname.format` либо через конфигурационный файл `$CATALINA_BASE/conf/system.properties`, либо используя опцию запуска JVM `-D` из командной строки (например, с помощью файла `$CATALINA_BASE/bin/setenv.sh`).

## Переопределение через свойства файла дескриптора `openejb-jar.xml`

Данный файл поставляется в составе EJB модуля в каталоге `META-INF`.

```
<openejb-jar>
  <properties>
    openejb.deploymentId.format = {ejbName}
    openejb.jndiname.format = {deploymentId}{interfaceType.annotationName}
  </properties>
</openejb-jar>
```

## Переопределение с помощью тэга `jndi` в файле дескриптора `openejb-jar.xml`

Следующим способом можно установить имя `sample` для EJB бина, реализующего интерфейс `com.example.SampleBean`:

```
<openejb-jar>
  <ejb-deployment ejb-name="Sample">
    <jndi name="sample" interface="com.example.SampleBean"/>
  </ejb-deployment>
</openejb-jar>
```

Аналогично можно установить имя `sample` для EJB бина аннотированного с помощью `@Remote`:

```
<openejb-jar>
  <ejb-deployment ejb-name="Sample">
    <jndi name="sample" interface="Remote"/>
  </ejb-deployment>
</openejb-jar>
```

Безусловно установить имя EJB бина:

```
<openejb-jar>
  <ejb-deployment ejb-name="Sample">
```

```
<jndi name="sample"/>
</ejb-deployment>
</openejb-jar>
```

В качестве имени можно также использовать шаблоны, описанные в таблице выше.

```
<openejb-jar>
  <ejb-deployment ejb-name="Sample">
    <jndi name="ejb/{interfaceClass.simpleName}"
interface="com.example.SampleBean"/>
  </ejb-deployment>
</openejb-jar>
```

Также возможно многократное использование тэга `jndi` если требуется более полный контроль над именами, в зависимости от предоставляемых интерфейсов или логических групп интерфейсов (Local, Remote, LocalHome, RemoteHome).

Например, имея EJB бин `SampleBean`, реализующий следующие интерфейсы: `com.example.LocalOne`, `com.example.LocalTwo`, `com.example.RemoteOne`, `com.example.RemoteTwo`, `com.example.SampleBeanHome`, `com.example.SampleBeanLocalHome`, с помощью тэга `jndi` можно применить различные подходы к формированию имен.

Следующие четыре варианты конфигурации эквивалентны друг другу и служат целям демонстрации гибкого контроля над JNDI именами EJB бинов:

```
<openejb-jar>
  <ejb-deployment ejb-name="SampleBean">
    <jndi name="LocalOne" interface="com.example.LocalOne"/>
    <jndi name="LocalTwo" interface="com.example.LocalTwo"/>
    <jndi name="RemoteOne" interface="com.example.RemoteOne"/>
    <jndi name="RemoteTwo" interface="com.example.RemoteTwo"/>
    <jndi name="SampleBeanHome" interface="com.example.SampleBeanHome"/>
    <jndi name="SampleBeanLocalHome"
interface="com.example.SampleBeanLocalHome"/>
  </ejb-deployment>
</openejb-jar>
```

или

```
<openejb-jar>
  <ejb-deployment ejb-name="SampleBean">
    <!-- применяется к LocalOne и LocalTwo -->
```

```
<jndi name="{interfaceClass.simpleName}" interface="Local"/>

<!-- применяется к RemoteOne и RemoteTwo -->
<jndi name="{interfaceClass.simpleName}" interface="Remote"/>

<!-- применяется к SampleBeanHome -->
<jndi name="{interfaceClass.simpleName}" interface="RemoteHome"/>

<!-- применяется к SampleBeanLocalHome -->
<jndi name="{interfaceClass.simpleName}" interface="LocalHome"/>
</ejb-deployment>
</openejb-jar>
```

или

```
<openejb-jar>
  <ejb-deployment ejb-name="SampleBean">
    <!-- применяется к RemoteOne, RemoteTwo, SampleBeanHome, и
SampleBeanLocalHome -->
    <jndi name="{interfaceClass.simpleName}"/>

    <!-- следующие два переопределяют формат выше -->
    <jndi name="LocalOne" interface="com.example.LocalOne"/>
    <jndi name="LocalTwo" interface="com.example.LocalTwo"/>
  </ejb-deployment>
</openejb-jar>
```

или

```
<openejb-jar>
  <ejb-deployment ejb-name="FooBean">
    <!-- applies to LocalOne, LocalTwo, RemoteOne, RemoteTwo, FooHome, and
FooLocalHome -->
    <jndi name="{interfaceClass.simpleName}"/>
  </ejb-deployment>
</openejb-jar>
```

