



Технологии Java EE на практике

Java EE 6 и сервер приложений GlassFish 3

Практическое руководство по установке и конфигурированию сервера приложений GlassFish v.3, а также по разработке приложений Java EE 6 и их развертыванию на этом сервере.

Дэвид Хеффельфингер

DMK
ИЗДАТЕЛЬСТВО

УДК 004.438Java EE
ББК 32.973.26-018.2
X41

Дэвид Хеффельфингер

X41 Java EE 6 и сервер приложений GlassFish 3. Пер. с англ.: Карышев Е.Н. – М.: ДМК Пресс, 2013. – 416 с.: ил.

ISBN 978-5-94074-902-8

Книга представляет собой практическое руководство с очень удобным подходом, позволяющим читателю быстрее освоить технологии Java EE 6. Все рассмотренные основные интерфейсы Java EE 6 и подробная информация о сервере GlassFish 3 подкреплены практическими примерами их использования.

Платформа Java Enterprise Edition (Java EE) 6 является отраслевым стандартом для корпоративных вычислений Java, а сервер приложений GlassFish представляет собой эталонную среду реализации спецификации Java EE. В книге рассматриваются различные соглашения и аннотации Java EE 6, которые помогут существенно упростить разработку корпоративных приложений Java. Описываются последние версии технологий Servlet, JSP, JSF, JPA, EJB и JAX-WS, а также новые дополнения к спецификации Java EE, в частности JAX-RS и CDI. Рассмотрены задачи администрирования, конфигурирования и использования сервера GlassFish 3 для развертывания корпоративных приложений.

Настоящее издание предназначено для разработчиков Java, желающих стать специалистами в разработке корпоративных приложений с использованием платформы Java EE 6. Для изучения материала необходимо иметь некоторый опыт работы с Java, однако знаний в области Java EE или J2EE не требуется.

Книга официально рекомендуется компанией Oracle – разработчиком перечисленных технологий – в качестве учебного пособия.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-1-849510-36-3 (англ.)
ISBN 978-5-94074-902-8 (рус.)

© Published by Packt Publishing Ltd.
© Оформление, перевод на русский язык
ДМК Пресс, 2013

Содержание

Об авторе	5
О рецензентах	6
Содержание	7
Предисловие	16
Темы, освещаемые в книге	16
Что нужно для чтения этой книги	19
Для кого эта книга	19
Соглашения	19
Поддержка клиентов	20
Сообщения об ошибках	21
Незаконное воспроизведение содержимого книги	21
Вопросы	21
1. Знакомство с сервером GlassFish	22
Общий обзор Java EE и GlassFish	22
Новые возможности Java EE	23
JavaServer Faces (JSF) 2.0	23
Enterprise JavaBeans (EJB) 3.1	23
API Персистентности Java (JPA) 2.0	24
Контексты и инжекция зависимости для Java (Web Beans 1.0)	24
API Java Сервлета 3.0	24
API Java для веб-сервисов RESTful (JAX-RS) 1.1	25
API Java для веб-сервисов XML (JAX-WS) 2.2	25
Архитектура Java для связывания с XML (JAXB) 2.2	25
Новые возможности GlassFish v.3	25
Преимущества GlassFish	26

Получение GlassFish	27
Установка GlassFish	28
Зависимости GlassFish	28
Выполнение установки	29
Проверка установки	34
Развертывание нашего первого приложения Java EE	35
Развертывание приложения через веб-консоль	35
Отмена развертывания приложения через веб-консоль	37
Развертывание приложения с помощью командной строки	38
Каталог autodeploy	39
Утилита командной строки asadmin	40
Домены GlassFish	41
Создание доменов	41
Удаление доменов	43
Остановка домена	43
Настройка подключения к базе данных	43
Создание пулов соединений	44
Создание источников данных	48
Заключительные замечания	49
Резюме	49
2. Разработка и развертывание сервлета	50
Что такое сервлет?	50
Написание нашего первого сервлета	51
Компиляция сервлета	52
Конфигурирование сервлета	53
Упаковка веб-приложения	55
Развертывание веб-приложения	56
Тестирование веб-приложения	56
Обработка HTML-форм	58
Переадресация запросов и перенаправление откликов	64
Переадресация запроса	64
Перенаправление отклика	67

Сохранение данных приложения между запросами	69
Новые возможности, появившиеся в Сервлете 3.0	71
Необязательный дескриптор развертывания web.xml	71
Аннотация @WebServlet	71
Передача сервлету параметров инициализации через аннотации	73
Аннотация @WebFilter	74
Аннотация @WebListener	76
Подключаемость	78
Программное конфигурирование веб-приложений	80
Асинхронная обработка	82
Резюме	84
3. JavaServer Pages	85
Введение в JavaServer Pages	85
Разработка нашей первой JSP-страницы	86
Неявные объекты JSP	90
JSP и JavaBeans	97
Повторное использование JSP-контента	101
Пользовательские теги JSP	103
Расширение класса SimpleTagSupport	104
Использование файлов тегов для создания пользовательских тегов JSP	109
Унифицированный язык выражений	113
XML-синтаксис JSP	116
Резюме	118
4. Библиотека стандартных тегов JSP	119
JSTL-библиотека базовых тегов	119
JSTL-библиотека тегов форматирования	127
JSTL-библиотека SQL-тегов	131
JSTL-библиотека XML-тегов	136
Функции JSTL	140
Резюме	143

5. Подключение к базе данных	144
База данных CUSTOMERDB	144
JDBC	145
Извлечение данных из базы данных	146
Изменение информации в базе данных	152
API Персистентности Java	154
Отношения сущности	159
Отношения «один к одному»	159
Отношения «один ко многим»	164
Отношения «многие ко многим»	168
Составные первичные ключи	173
Язык запросов персистентности Java	177
Новые функции, введенные в JPA 2.0	181
API Критериев	181
Поддержка проверки допустимости со стороны бинов	184
Резюме	186
6. JavaServer Faces	187
Введение в JSF 2.0	187
Фэйслеты	187
Необязательный файл faces-config.xml	188
Стандартное расположение ресурсов	188
Разработка нашего первого JSF 2.0-приложения	189
Фэйслеты	189
Этапы проекта	193
Проверка допустимости	195
Группировка компонентов	197
Отправка формы	197
Управляемые бины	198
Контексты управляемых бинов	199
Навигация	200
Пользовательская проверка допустимости данных	201
Создание нестандартных элементов верификации	202
Методы блока проверки допустимости	204

Настройка сообщений JSF по умолчанию	207
Настройка стилей сообщения	207
Настройка текста сообщения	209
Интеграция JSF и JPA	211
Включение Ajax в приложения JSF 2.0	218
Стандартные компоненты JSF	222
Базовые компоненты JSF	222
Тег <f:actionListener>	222
Тег <f:ajax>	222
Тег <f:attribute>	223
Тег <f:convertDateTime>	223
Тег <f:convertNumber>	223
Тег <f:converter>	224
Тег <f:event>	224
Тег <f:facet>	225
Тег <f:loadBundle>	225
Тег <f:metadata>	225
Тег <f:param>	226
Тег <f:phaseListener>	226
Тег <f:selectItem>	226
Тег <f:selectItems>	226
Тег <f:setPropertyActionListener>	227
Тег <f:subview>	227
Тег <f:validateBean>	227
Тег <f:validateDoubleRange>	228
Тег <f:validateLength>	228
Тег <f:validateLongRange>	228
Тег <f:validateRegex>	229
Тег <f:validateRequired>	229
Тег <f:validator>	229
Тег <f:valueChangeListener>	229
Тег <f:verbatim>	230
Тег <f:view>	230
Тег <f:viewParam>	230

HTML-компоненты JSF	231
Ter <h:body>	231
Ter <h:button>	231
Ter <h:column>	231
Ter <h:commandButton>	231
Ter <h:commandLink>	232
Ter <h:dataTable>	232
Ter <h:form>	232
Ter <h:graphicImage>	233
Ter <h:head>	233
Ter <h:inputHidden>	233
Ter <h:inputSecret>	233
Ter <h:inputText>	234
Ter <h:inputTextarea>	234
Ter <h:link>	234
Ter <h:message>	234
Ter <h:messages>	235
Ter <h:outputFormat>	235
Ter <h:outputLabel>	235
Ter <h:outputLink>	236
Ter <h:outputScript>	236
Ter <h:outputStylesheet>	236
Ter <h:outputText>	236
Ter <h:panelGrid>	236
Ter <h:panelGroup>	237
Ter <h:selectBooleanCheckbox>	238
Ter <h:selectManyCheckbox>	238
Ter <h:selectManyListbox>	238
Ter <h:selectManyMenu>	239
Ter <h:selectOneListbox>	239
Ter <h:selectOneMenu>	239
Ter <h:selectOneRadio>	239
Дополнительные библиотеки компонентов JSF	240
Резюме	240

7. Служба обмена сообщениями Java	241
Настройка GlassFish для использования JMS	241
Создание фабрики JMS-соединений	241
Создание очереди JMS-сообщений	243
Создание темы JMS-сообщений	244
Очереди сообщений	245
Отправка сообщений в очередь сообщений	245
Извлечение сообщений из очереди сообщений	249
Асинхронный прием сообщений из очереди сообщений	250
Просмотр очередей сообщений	253
Темы сообщений	254
Отправка сообщений теме сообщений	254
Получение сообщений от темы сообщений	255
Создание долговременных подписчиков	257
Резюме	260
8. Безопасность	261
Области безопасности	261
Предопределенные области безопасности	262
Область администратора	262
Область файла	264
Стандартная аутентификация через область файла	265
Область сертификата	276
Создание самоподписанных сертификатов	276
Конфигурирование приложений для использования области сертификата	280
Определение дополнительных областей	283
Определение дополнительных областей файла	283
Определение дополнительных областей сертификата	285
Определение области LDAP	286
Определение области Solaris	287
Определение области JDBC	288
Определение пользовательских областей	293
Резюме	298

9. Enterprise JavaBeans	299
Сеансовые бины	300
Простой сеансовый бин	300
Более реалистический пример	303
Вызов сеансовых бинов из веб-приложений	305
Одноэлементный сеансовый бин (Singleton)	306
Асинхронные вызовы метода	307
Управляемые сообщением бины	309
Транзакции в Enterprise JavaBeans	310
Транзакции, управляемые контейнером	311
Транзакции, управляемые бином	313
Жизненный цикл Enterprise JavaBeans	315
Жизненный цикл сеансового бина с сохранением состояния	316
Жизненный цикл сеансового бина, не сохраняющего состояние	319
Жизненный цикл управляемых сообщением бинов	321
Служба таймера EJB	322
Выражения таймера EJB на основе календаря	324
Безопасность EJB	326
Аутентификация клиента	329
Резюме	330
10. Контексты и инжекция зависимости	332
Именованные бины	332
Инжекция зависимости	334
Квалификаторы	335
Контексты именованных бинов	338
Резюме	345
11. Веб-сервисы JAX-WS	346
Разработка веб-сервисов JAX-WS	346
Разработка клиента веб-сервиса	351
Отправка вложений веб-сервисам	357

Представление EJB как веб-сервисов	359
Клиенты веб-сервиса EJB	360
Безопасность веб-сервисов	360
Безопасность веб-сервисов EJB	362
Резюме	364
12. RESTful веб-сервисы в Jersey и JAX-RS	365
Введение в веб-сервисы RESTful и JAX-RS	365
Разработка простого веб-сервиса RESTful	366
Конфигурирование пути к ресурсам REST для нашего приложения	368
Конфигурирование через web.xml	368
Конфигурирование через аннотацию @ApplicationPath	369
Тестирование нашего веб-сервиса	370
Преобразование данных между Java и XML с помощью JAXB	372
Разработка клиента веб-сервиса RESTful	375
Параметры запроса и пути	377
Параметры запроса	377
Отправка параметров запроса через клиентский API Jersey	378
Параметры пути	380
Отправка параметров пути через клиентский API Jersey	382
Приложение А. Отправка электронной почты из приложений Java EE	384
Конфигурирование сервера GlassFish	384
Реализация функциональности доставки электронной почты	387
Приложение Б. Интеграция с IDE	390
NetBeans	390
Eclipse	392
Алфавитный указатель	397

1

Знакомство с сервером GlassFish

В этой главе мы обсудим, как приступить к работе с сервером GlassFish. Вот некоторые из обсуждаемых тем:

- общий обзор Java EE и GlassFish;
- получение сервера приложений GlassFish;
- установка сервера приложений GlassFish;
- проверка установки сервера GlassFish;
- развертывание приложения Java EE;
- установка соединения с базой данных.

Общий обзор Java EE и GlassFish

Спецификация Java EE (ранее называемая J2EE) включает в себя стандартный набор технологий для разработки серверных приложений Java. Технологии Java EE включают, среди прочего, Сервлеты (Servlets), Серверные страницы Java (Java Server Pages (JSP)), Каркас стандартных компонентов Java Server Faces (JSF), Компоненты корпоративных приложений (Enterprise Java Beans (EJB)), Службу обмена сообщениями Java (Java Messaging Service (JMS)), API Персистентности Java (Java Persistence API (JPA)), используемый для сохранения сущностей в базе данных, API Java для веб-сервисов XML (Java API for XML Web Services (JAX-WS)) и API Java для веб-сервисов RESTful (Java API for RESTful Web Services (JAX-RS)). Существует несколько коммерческих вариантов серверов приложений и несколько вариантов серверов приложений с открытым исходным кодом. Серверы приложений Java EE позволяют разработчикам разрабатывать и развертывать Java EE-совместимые приложения; одним из таких серверов приложений является сервер GlassFish. В числе других серверов приложений Java EE с открытым исходным кодом – JBoss Red Hat, Apache Software Foundation’s Geronimo и ObjectWeb JOnAS. Коммерческие варианты сервера – Oracle (ранее BEA) Weblogic, IBM Websphere и Oracle Application Server.

GlassFish является сервером приложений Java EE с открытым исходным кодом, находящимся в свободном доступе. GlassFish лицензируется в соответствии с Общей лицензией разработки и распространения (Common Development and Distribution License (CDDL)).



Чтобы узнать больше о лицензиях GlassFish, обратитесь к сайту http://glassfish.java.net/public/CDDL+GPL_1_1.html.

Как полностью совместимый сервер приложений Java EE, GlassFish предоставляет необходимые библиотеки, позволяющие нам разрабатывать и развертывать Java-приложения, соответствующие спецификации Java EE.

Новые возможности Java EE

Java EE 6 на сегодняшний день является самой последней версией спецификации Java EE, включающей несколько усовершенствований и дополнений. В следующих разделах перечислены основные усовершенствования спецификации, которые представляют интерес для разработчиков корпоративных приложений.

JavaServer Faces (JSF) 2.0

Java EE 6 включает новую версию JSF. В каркасе стандартных компонентов JSF 2.0 появились следующие важные возможности:

- JSF 2.0 принял фэйслеты (Facelets) в качестве части официальной спецификации. Фэйслеты – технология представления, специально разработанная для JSF. В число преимуществ фэйслетов входят следующие возможности: возможность определения представления в XHTML, возможность упрощения создания шаблонов и возможность разработки компонентов JSF только с использованием разметки – без использования кода Java;
- JSF 2.0 также включает возможность конфигурирования приложения JSF с использованием аннотаций, значительно снижая, таким образом, а во многих случаях и полностью устраняя необходимость использования XML для конфигурирования.

Enterprise JavaBeans (EJB) 3.1

Ранние версии спецификации EJB приобрели репутацию сложных в практическом применении.

EJB 3.0 сделал большой шаг по пути к упрощению разработки EJB. EJB 3.1, в свою очередь, добавил новые возможности, которые еще больше упростили разработку EJB, а именно:

- локальные интерфейсы теперь являются не обязательными, поскольку фактический экземпляр бина может быть инжектирован (injected) в локального клиента;
- одиночный (Singleton) сеансовый бин может использоваться для управления состояниями приложения;

- сеансовые бины теперь могут быть вызваны асинхронно, позволяя нам использовать сеансовые бины для задач, которые ранее были зарезервированы для JMS и управляемых сообщением бинов;
- улучшенная служба таймера EJB теперь позволяет нам планировать задания декларативно через аннотации;
- Enterprise JavaBeans могут быть упакованы в файл веб-архива (Web ARchive (WAR)). Эта возможность значительно упрощает упаковку EJB, поскольку ранее требовался файл корпоративного архива (Enterprise ARchive (EAR)) для совместной упаковки в один модуль веб-функциональности и EJB-функциональности.

API Персистентности Java (JPA) 2.0

JPA был введен в качестве части стандартной спецификации Java EE в версии 5. JPA был призван заменить *Сущностные бины* (Entity Beans), использовавшиеся на тот момент в качестве стандартного каркаса для объектно-реляционного отображения в Java EE. JPA перенял идеи сторонних объектно-реляционных каркасов, таких как Hibernate, JDO и т. д., и сделал их частью стандарта.

JPA 2.0 в сравнении с JPA 1.0 улучшен по нескольким направлениям:

- несущностные коллекции теперь могут быть сохранены с помощью использования аннотаций: `@ElementCollection` и `@CollectionTable`;
- запросы JPA могут быть созданы с помощью нового API Критериев, уменьшая тем самым зависимость от JPQL;
- улучшен язык запросов JPA (JPQL) путем добавления поддержки для SQL-подобных выражений CASE, а также операторов NULLIF и COALESCE.

Контексты и инжекция зависимости для Java (Web Beans 1.0)

Контексты (Context) и инжекция (Injection) зависимости представляют собой API, который помогает упростить разработку корпоративных приложений. Этот API помогает унифицировать веб- и транзакционные уровни приложения Java EE. Например, контексты и инжекция зависимости позволяют использовать Enterprise JavaBeans (EJB) в качестве управляемых бинов JSF.

API Java Сервлета 3.0

Сервлеты являются строительными блоками всех веб-приложений Java. Ранние веб-приложения Java обращались к API сервлета напрямую. За прошедшие годы было создано несколько API-надстроек над API сервлета, некоторые из них – в рамках стандарта, а некоторые – сторонними разработчиками. Все каркасы веб-приложений Java, такие как JSF, Struts, Wicket, Tapestry и т. д., опираются на API сервлета, вы-

полняя его работу «за кулисами». API самого сервлета не сильно изменился за прошедшее время. Java EE 6 включает несколько усовершенствований API сервлета, таких как аннотации, веб-фрагменты и асинхронные запросы.

API Java для веб-сервисов RESTful (JAX-RS) 1.1

JAX-RS является API Java для разработки веб-сервисов JAX-RS. Веб-сервисы JAX-RS используют архитектуру передачи состояния представления (Representational State Transfer (REST)).

Java EE 6 принял JAX-RS в качестве части официальной спецификации Java EE.

API Java для веб-сервисов XML (JAX-WS) 2.2

JAX-WS является API Java для веб-сервисов XML. JAX-WS используется для разработки традиционных веб-сервисов на основе SOAP. Java EE 6 включает обновленную спецификацию JAX-WS. JAX-WS 2.2 является корректировочной версией с незначительными улучшениями и усовершенствованиями по сравнению с JAX-WS 2.0.

Архитектура Java для связывания с XML (JAXB) 2.2

JAXB используется для отображения классов Java на XML и обратно. Java EE 6 включает обновленную корректировочную версию JAXB.

Новые возможности GlassFish v.3

GlassFish v.3 – первый сервер приложений, поддерживающий спецификацию Java EE 6 в полном объеме. Данное обстоятельство не должно удивлять, поскольку GlassFish является эталонной реализацией спецификации Java EE. GlassFish v.3 предлагает следующие заслуживающие внимания особенности:

- *имеет модульную архитектуру, основанную на OSGi.* Архитектура на основе OSGi позволяет GlassFish иметь подключаемые модули, позволяя нам, таким образом, запускать его только с теми функциональными возможностями, которые нам действительно необходимы. В результате не придется тратить впустую ресурсы, такие как память и вычислительные ресурсы ЦП, на функциональность, которая нами не используется;
- *является встраиваемым сервером; может быть встроен в существующую JVM.* Это позволяет нам писать приложения Java со встроенным

в них сервером GlassFish. Чтобы использовать эту возможность, следует просто добавить библиотеки GlassFish к нашему проекту;

- *является расширяемым сервером; может быть адаптирован для поддержки дополнительных технологий, которые не являются частью спецификации Java EE.* Несколько расширений для него доступны из формы центра обновления GlassFish, например поддержка Grails (платформа веб-приложений на основе Groovy) и JRuby on Rails. Функция расширяемости GlassFish v.3 позволяет разработчикам и поставщикам приложений реализовывать свои собственные расширения GlassFish.

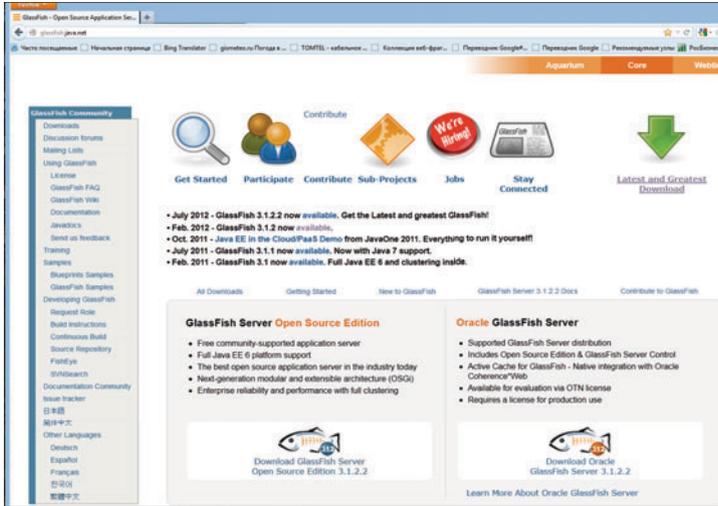
Преимущества GlassFish

Имеется много вариантов серверов приложений Java EE, но почему чаще всего выбирают именно GlassFish? Помимо очевидных преимуществ бесплатного сервера, GlassFish предлагает и многие другие:

- *имеет доступную коммерческую поддержку.* Коммерческая поддержка относительно недорога. Многие корпоративные покупатели программного обеспечения не будут использовать ПО, для которого не доступна коммерческая поддержка, поэтому невысокая стоимость коммерческой поддержки позволяет использовать GlassFish и в тех рыночных «нишах», где иного не предусмотрено;
- *является эталонной реализацией спецификации Java EE.* Это означает, что другие производители серверов приложений могут использовать GlassFish для того, чтобы убедиться, что их продукт удовлетворяет спецификации. Теоретически GlassFish может использоваться для устранения неисправностей других серверов приложений. Если приложение, развернутое на другом сервере приложений, не функционирует надлежащим образом, при том что оно работает правильно, будучи развернутым на сервере GlassFish, ошибка с наибольшей степенью вероятности заключена в другом сервере приложений;
- *поддерживает самые последние версии спецификации Java EE.* Поскольку GlassFish является эталонной реализацией спецификации Java EE, он реализует самые последние нововведения в спецификации раньше, чем любые другие серверы приложений на рынке. Действительно, на момент написания этой книги GlassFish является единственным Java EE-совместимым сервером приложений на рынке, который поддерживает спецификацию Java EE 6 в полном объеме.

Получение GlassFish

Сервер GlassFish может быть загружен с веб-узла <http://glassfish.java.net/>. При вводе этого URL в адресной строке веб-обозревателя открывается страница, показанная на следующем снимке экрана:



Щелкнув по ссылке для скачивания, мы перейдем к странице, содержащей таблицу такого вида:

GlassFish Server	The GlassFish Server is available in different releases to address different requirements. For more details, check out this detailed comparison between v2 and v3 .	
GlassFish v2.1	A <i>final</i> release based on the Java EE 5 standard, with clustering, load balancing and high availability and with Update Center. Commercially supported by Sun and recommended for production environments. Community Release Sun's Supported Release Quick Start Guide Documentation Support Training	 Download
GlassFish v3 Preview	An <i>unsupported</i> , <i>early</i> access release implementing the latest version of the Java EE 6 standard. Includes an extensible core based on OSGi, Admin Console and Update Center. It does not have HA, clustering and other features. Community Release Java EE 6 SDK Preview Quick Start Guide Installation Guide Documentation	 Download
GlassFish v3 Prelude	A <i>final</i> release implementing the web layer of the Java EE 5 standard. It is based on the same core as GF v3 Preview; it does not have production features but it is more stable. Community Release Sun's Supported Release Quick Start Guide Installation Guide Documentation Support Patches	 Download

Во время написания этой книги GlassFish 3 еще не был выпущен официально, но, как видно из предыдущего снимка экрана, имелась Java EE 6-совместимая версия,

доступная для предварительного просмотра. Щелкнув по ссылке для загрузки (**Download**) для этой версии, мы перейдем к следующей странице:

How do I get GlassFish v3 Preview?

GlassFish v3 Preview Community Distributions

GlassFish v3 Preview (en)	Size (MB)	GlassFish v3 Web Profile Preview (en)	Size (MB)	Description
Windows Installer File	50	Windows Installer File	30	GUI-based installer for Windows
Self-Extracting Installer File	50	Self-Extracting Installer File	30	GUI-based installer for Solaris, Linux and MacOS X
Zip File	71	Zip File	40	Platform-independent download file

Required JDK Version

Installations require JDK 6. The minimum (and certified) version of the JDK software that is required depends on operating system:

- For supported operating systems except MacOS, the minimum required version is 1.6.0_13.
- For the MacOS operating system, the minimum required version is 1.6.0_7.

Как видно из рисунка, страница имеет ссылки загрузки для всех официально поддерживаемых платформ (Windows, Solaris, Linux и Mac OSx) и, дополнительно, – платформонезависимый ZIP-файл.

Чтобы загрузить GlassFish, нужно просто щелкнуть по ссылке для используемой нами платформы. Файл должен начать загружаться сразу. После того как он будет загружен, мы увидим название вроде такого: `GlassFish-v3-preview-unix.sh`, `GlassFish-v3-preview-windows.exe` или `GlassFish-v3-preview.zip`. Точное имя файла зависит от конкретной версии GlassFish и используемой нами платформы.

Установка GlassFish

Мы будем использовать установщик Unix для пояснения процесса установки. Этот установщик работает под Linux, Solaris и Mac OSx. Установка для Windows практически не отличается от демонстрационного примера.

Процесс установки GlassFish достаточно прост; тем не менее GlassFish предполагает, что в нашей системе должны присутствовать определенные элементы, от которых он зависит.



NetBeans 6.8 поставляется в комплекте с GlassFish v.3. При установке дистрибутива NetBeans Java, сервер GlassFish также будет установлен автоматически.

Зависимости GlassFish

Для установки GlassFish v.3 на нашей рабочей станции должна иметься свежая версия Комплекта разработчика Java (Java Development Kit (JDK)) (требуется JDK 1.6 или более поздняя версия), а в нашем системном пути должны быть исполняемые программы Java. Самый последний JDK можно загрузить с веб-узла: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

Выполнение установки

После того как будет установлен JDK, можно приступить к установке GlassFish v.3, просто запусив на выполнение загруженный файл (возможно, нам придется изменить полномочия для запуска программы-установщика, например запуск от имени администратора, для Windows):

```
../GlassFish-v3-preview-Unix.sh
```

Фактическое имя файла будет зависеть от версии загруженного GlassFish. Для успешной установки GlassFish нужно выполнить следующие действия:

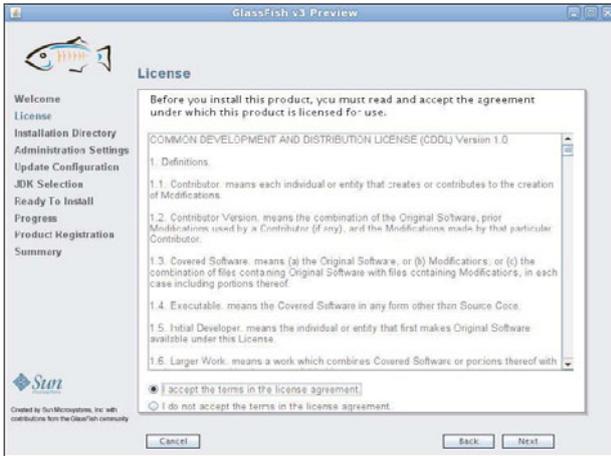
1. После выполнения предыдущей команды установщик GlassFish начнет инициализацию:



По прошествии нескольких секунд мы увидим экран приветствия установщика:



2. После щелчка по кнопке **Далее (Next)** появится следующий экран установщика, предлагающий нам принять условия лицензионного соглашения:

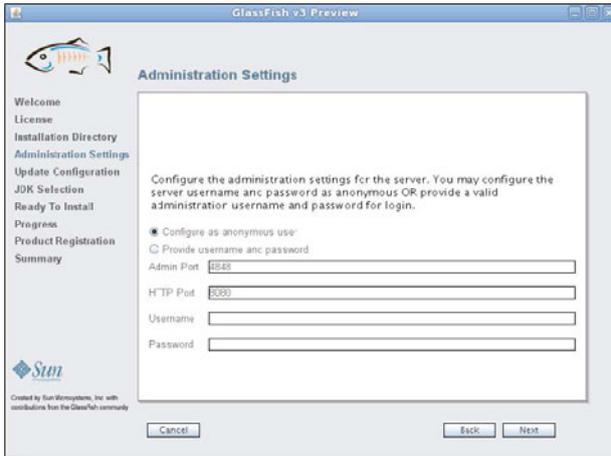


3. Следующая страница установщика запросит у нас каталог установки. Значением по умолчанию для каталога установки является каталог, называемый `GlassFishv3` в нашем домашнем каталоге. Будет разумным оставить это значение, хотя мы можем его изменить.

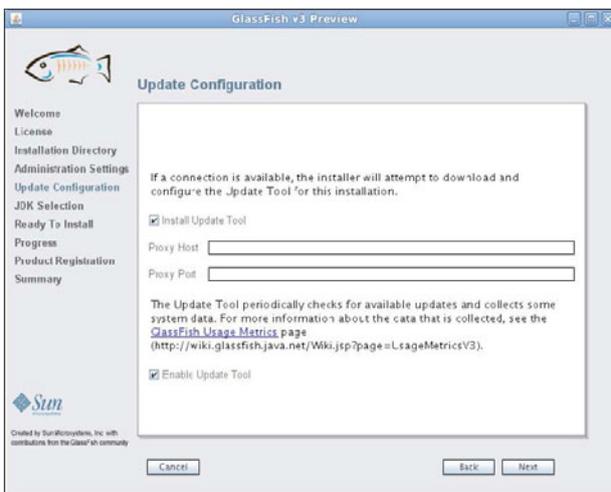


4. Следующая страница установщика позволяет нам настроить порты администратора и HTTP для GlassFish. Кроме того, здесь мы можем указать имя и пароль пользователя, выполняющего роль администратора. По умолчанию не требуется никакой комбинации имени пользователя и пароля для входа в консоль администрирования. Это поведение по умолчанию

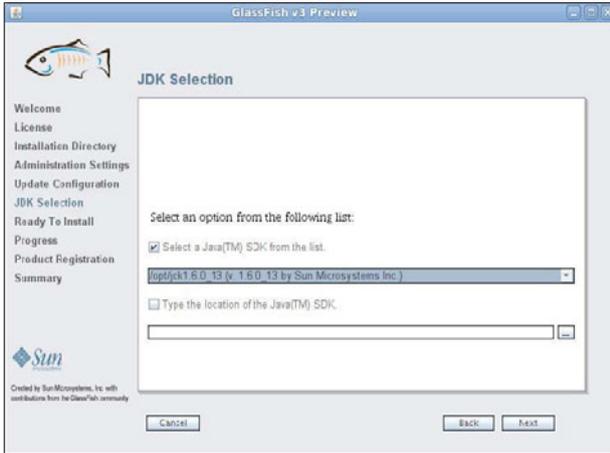
предусмотрено для режима разработки. Мы можем переопределить это поведение и предоставить имя пользователя и пароль на данном шаге мастера установки.



5. На данном этапе установки следует указать, хотим ли мы установить инструмент обновления GlassFish. Инструмент обновления дает возможность легко устанавливать дополнительные модули GlassFish. Поэтому, если позволяет дисковое пространство, рекомендуется его установить. Если мы получаем доступ в Интернет через прокси-сервер, можно ввести имя его хоста или IP-адрес и порт на этой странице мастера установки.



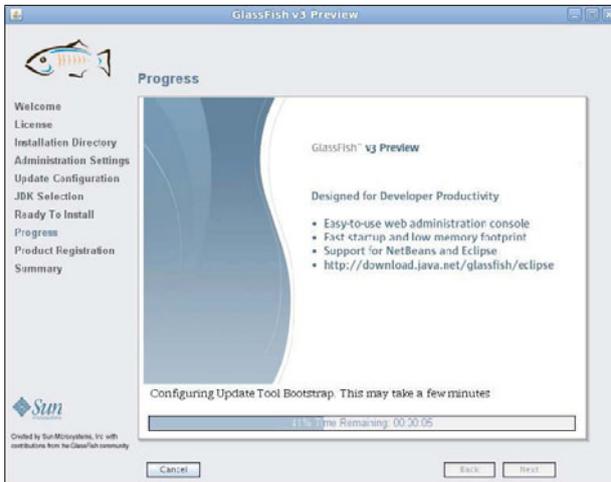
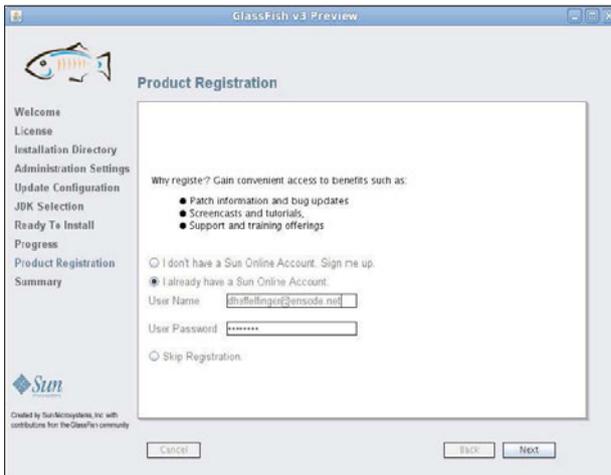
6. Теперь нам предлагаются на выбор два варианта: либо автоматическое обнаружение Java SDK, либо указание местонахождения SDK. По умолчанию выбирается тот Java SDK, который соответствует значению переменной среды `JAVA_HOME`.



7. На данном этапе установщик резюмирует действия, которые он собирается предпринять в процессе установки. Щелкните по кнопке **Установить** (Install), чтобы начать установку:

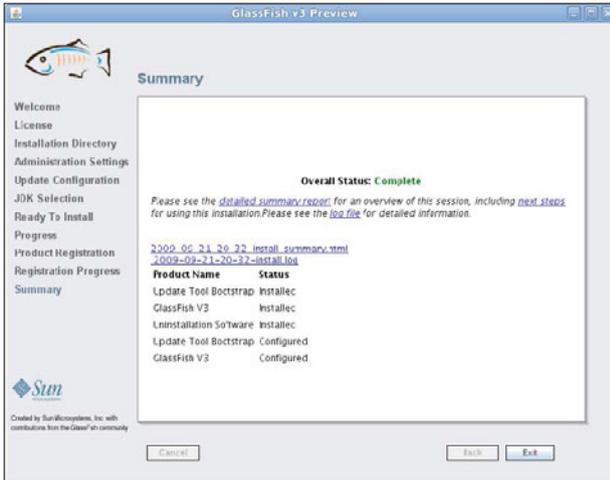


8. Процесс установки показан на следующем снимке экрана:

9. По завершении установки нам будет предложено зарегистрировать нашу копию GlassFish. На данном этапе мы можем связать установку GlassFish с существующей у нас онлайн-учетной записью Sun¹, либо создать новую онлайн-учетную запись Sun, либо пропустить регистрацию:

¹ Здесь речь идет об учетной записи Oracle (начиная с 2011 года, после поглощения Sun Microsystems), к которой можно получить доступ по адресу: <https://login.oracle.com/mysso/signon.jsp>.
– Прим. перев.

10. Следующая страница установщика показывает сводную информацию об установке. Здесь нам нужно просто щелкнуть по кнопке **Выход** (Exit), чтобы выйти из установщика:



Проверка установки

Чтобы запустить GlassFish, нужно изменить текущий каталог на *[Каталог установки GlassFish]/GlassFishv3/bin* и выполнить следующую команду:

```
./asadmin start-domain domain1
```



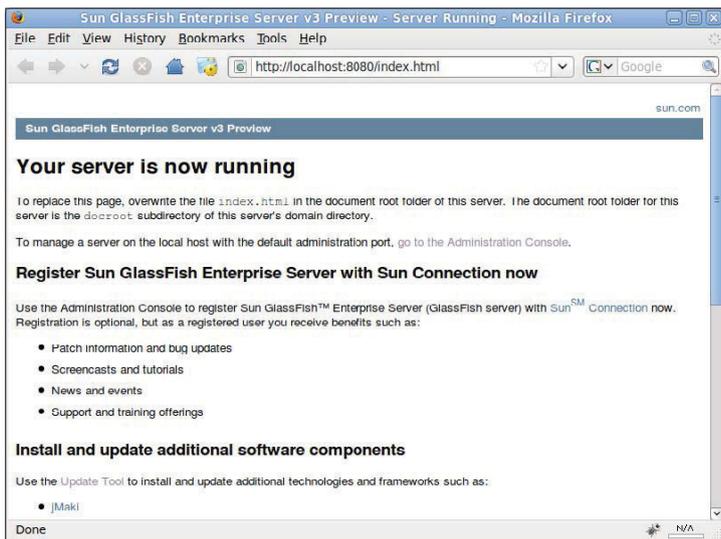
Эта команда, как и большинство команд, приведенных в этой главе, предполагает использование Unix или Unix-подобной операционной системы. Для систем Windows начальных символов `./` не требуется.

Через несколько секунд после выполнения предыдущей команды мы должны увидеть внизу окна терминала сообщение, подобное следующему:

```
Имя запущенного домена: [domain1] и его
расположение: [/home/heffel/GlassFishv3/GlassFish/domains/domain1].
Порт администратора для домена: [4848].
(Name of the domain started: [domain1] and
its location: [/home/heffel/glassfishv3/glassfish/domains/domain1].
Admin port for the domain: [4848].)
```

Затем мы можем открыть окно обозревателя и ввести в его адресной строке следующий URL: <http://localhost:8080>.

Если бы все было нормально, то мы не должны были бы видеть страницу, подобную следующему снимку экрана:



Получение справки

Если какой-либо из наших предыдущих шагов завершился неудачно либо нам требуется справка по общим вопросам, касающимся GlassFish, то мы можем обратиться к большому информационному ресурсу – форуму GlassFish, который можно найти по адресу: <http://www.java.net/forums/glassfish/glassfish>.

Развертывание нашего первого приложения Java EE

Чтобы проверить, правильно ли работает установленный нами сервер GlassFish, развернем WAR-файл (веб-архив) и убедимся, что он развертывается и выполняется надлежащим образом. Прежде чем двигаться дальше, пожалуйста, загрузите файл `simpleapp.war` с веб-сайта www.dmk-press.ru.

Развертывание приложения через веб-консоль

Чтобы развернуть `simpleapp.war`, откройте обозреватель и перейдите к следующему URL: <http://localhost:4848>. Вы должны увидеть экран общих задач консоли администрирования, который выглядит примерно так: