

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Методические указания по выполнению лабораторных работ
по курсу
«Сетевое программное обеспечение»

Лабораторная работа № 2
«Сетевая архитектура корпоративной системы документооборота»

Оглавление

1	Цель работы.	3
2	Теоретическая часть	3
2.1	Сетевая архитектура клиент-сервер.....	3
2.2	Архитектура клиент/сервер	3
2.3	Приложения клиент/сервер	4
2.3.1	Способ разделения приложений на части.....	4
2.3.2	Классы приложений клиент/сервер.....	5
2.3.3	Трехуровневая архитектура клиент/сервер.....	7
2.4	Промежуточное программное обеспечение	8
2.5	Описание архитектуры платформы корпоративной системы документооборота	10
2.5.1	Обзор платформы	10
2.5.2	Управление бизнес процессами (Business Process Management) –	10
2.5.3	Основные компоненты платформы OpenText EPS.....	11
2.5.4	Инсталляция и настройка базы данных	13
3	Задание на выполнение работы	28
4	Контрольные вопросы	28
5	ЛИТЕРАТУРА	29

1 Цель работы.

Целью работы является знакомство с архитектурой и установкой клиент-серверного приложения документооборота.

Продолжительность работы – 4 часа

2 Теоретическая часть

2.1 Сетевая архитектура клиент-сервер

Наиболее значительной тенденцией в развитии информационных систем за последние годы стал рост клиент-серверных систем, быстро вытесняющих как централизованные вычислительные системы с использованием мейнфреймов, так и другие подходы к распределенной обработке данных.

2.2 Архитектура клиент/сервер

Среда клиент/сервер заполнена клиентами и серверами. В роли **клиентов**, как правило, выступают **однопользовательские персональные компьютеры или рабочие станции**, обеспечивающие дружелюбный интерфейс для конечного пользователя (обычно это удобный графический интерфейс, включающий использование окон и мыши). **Основная цель клиента — обеспечить простоту использования и возможность работы со знакомым инструментарием.**

Каждый **сервер** в среде клиент/сервер обеспечивает для клиентов множество **совместно используемых сервисов**. Наиболее распространенный в настоящее время тип сервера — **сервер базы данных**, обычно реляционной.

Сервер обеспечивает одновременный доступ к базе данных многих клиентов и позволяет использовать для управления базой данных высокопроизводительную вычислительную систему.

В дополнение к клиентам и серверам третьим важным компонентом среды клиент/сервер является сеть.

Вычисления клиент/сервер — это **распределенные сетевые вычисления**. Пользователи, приложения и ресурсы распределены, например, в

соответствии с корпоративными требованиями и связаны между собой единой локальной или глобальной вычислительной сетью.

2.3 Приложения клиент/сервер

Основной задачей проектирования приложений архитектуры клиент/сервер является **распределение задач уровня приложения между клиентом и сервером.**

2.3.1 Способ разделения приложений на части

Можно предложить различные схемы разделения приложений на части, причем для каждого конкретного приложения можно предложить свою схему.

Существуют и типовые модели распределенных приложений. В детальной модели предлагается разделить приложение на **шесть** функциональных частей:

средства представления данных на экране, например средства графического пользовательского интерфейса;

логика представления данных на экране описывает правила и возможные сценарии взаимодействия пользователя с приложением: выбор из системы меню, выбор элемента из списка и т. п.;

логика приложения - набор правил для принятия решений, вычислительные процедуры и операции;

логика данных — операции с данными, хранящимися в некоторой базе, которые нужно выполнить для реализации логики приложения;

внутренние операции базы данных — действия СУБД, вызываемые в ответ на выполнение запросов логики данных, такие как поиск записи по определенным признакам;

файловые операции — стандартные операции над файлами и файловой системой, которые обычно являются функциями операционной системы.

На основе этой модели может быть построено несколько схем распределения частей приложения между узлами сети.

У клиента, и у сервера есть базовое программное обеспечение - это

операционная система, работающая на конкретной аппаратной платформе. Платформы и операционные системы клиента и сервера могут быть различны.

Взаимодействие клиентов и серверов обеспечивают разнообразные программы.(Пример- системы, поддерживающие работу с TCP/IP.)

Все функции, выполняемые приложением, могут быть разделены между клиентом и сервером так, чтобы **оптимизировать** использование компьютеров и сетевых ресурсов, а также возможности пользователей по выполнению различных задач и кооперации друг с другом при использовании разделяемых ресурсов.

Определяющим фактором в успешной деятельности среды клиент/сервер является способ взаимодействия пользователя с системой в целом.

Соответственно, важным является дизайн пользовательского интерфейса на клиентской машине. В большинстве систем клиент/сервер применяется графический интерфейс пользователя (graphical user interface – GUI)

На машине клиента используется модуль представления, который является частью распределенного приложения и обеспечивает дружелюбный пользовательский интерфейс.

2.3.2 Классы приложений клиент/сервер

В рамках общей схемы клиент/сервер имеется широкий спектр реализации, каждая из которых по-своему разделяет работу между клиентом и сервером.

Четыре основных класса приложений.

1. • **Обработка на одном узле.** По сути такая обработка не является вариантом среды клиент/сервер в обычном смысле. Такая обработка традиционно использовалась на мейнфреймах, когда практически вся обработка выполнялась центральным узлом, а пользовательский интерфейс осуществлялся посредством неинтеллектуальных терминалов. Даже если пользователь оснащен мощным микрокомпьютером, последний играет роль всего лишь

эмулятора терминала.

2. • **Обработка на сервере.** Одна из основных конфигураций клиент/сервер, в которой клиент отвечает за обеспечение графического интерфейса пользователя, а практически вся обработка выполняется на сервере. Такая конфигурация типична для ранних систем клиент/сервер, в особенности для бизнес-систем уровня отделов. Обоснование использования таких систем заключается в том, что пользовательские машины лучше всего подходят для реализации графического интерфейса пользователя, а поддержку приложений баз данных легче организовать на центральном сервере. Однако, хотя пользователь и получает более удобный и дружелюбный интерфейс, эта конфигурация принципиально не отличается от предыдущей, поэтому не происходит значительного увеличения производительности или существенных изменений поддерживаемых системой бизнес-функций.

3. • **Обработка у клиента.** В другом предельном случае практически вся обработка может осуществляться на машине клиента; на сервере выполняются только программы проверки корректности данных и подобные функции, которые лучше выполнять на сервере. В целом же более сложные функции логики базы данных выполняются на машине клиента. Такая архитектура позволяет пользователю работать с приложениями, приспособленными для его конкретных нужд, и в настоящее время, пожалуй, наиболее распространена.

4. • **Обработка при сотрудничестве.** В этой конфигурации приложение выполняется оптимальным образом, используя сильные стороны машин клиента и сервера и распределенности данных. Эта конфигурация несколько сложнее предыдущих, ее труднее поддерживать, но она обеспечивает более высокую производительность и эффективность работы сети по сравнению с остальными вариантами конфигураций.

Если значительная часть работы передана клиенту, то эта так называемая модель толстого клиента (fat client. В русскоязычной литературе встречается

также перевод "жирный клиент"). Модель получила распространение благодаря такому инструментарию разработки приложений, как PowerBuilder от Powersoft Corp. и SQL Windows от Gupta Corp.

Разрабатываемые с помощью этих инструментов приложения, как правило, применимы на уровне отделов и поддерживают от 25 до 150 пользователей. Основное достоинство модели толстого клиента заключается в использовании всей мощи настольной системы и переносе вычислений с сервера, что делает его работу более эффективной (не позволяя ему стать узким местом в работе среды).

Однако у такого подхода имеется ряд недостатков.

1. Увеличение функциональности вызывает излишнюю загрузку пользовательских машин, что приводит к необходимости дорогостоящего обновления всей клиентской техники.

2. При увеличении количества пользователей компании приходится раскошелиться и на высокоскоростную сеть, которая сможет выдержать интенсивные потоки данных между сервером и толстыми клиентами.

3. Наконец, проще и дешевле поддерживать и обновлять приложения на одном сервере, чем на десятках или сотнях пользовательских машин.

Подход тонкого клиента (thin client). Этот подход ближе к традиционной обработке на сервере и часто становится первым шагом при переходе корпорации от использования мейнфреймов к распределенной среде.

2.3.3 Трехуровневая архитектура клиент/сервер

Традиционная архитектура клиент/сервер включает два уровня, или слоя: уровень клиента и уровень сервера.

В последние годы становится популярной трехуровневая архитектура.

Приложение в ней распределено между машинами трех типов:

1. пользовательской машиной,

2. промежуточным сервером
3. и конечным сервером.

Пользовательская машина представляет собой уже рассматривавшуюся ранее машину клиента, который в трехуровневой модели обычно является тонким клиентом.

Машины промежуточного уровня являются воротами между тонкими клиентами и различными конечными серверами баз данных, преобразуя протоколы и отображая один тип запросов к базам данных в другой. Кроме того, они способны собирать и **обрабатывать данные из разных источников**.

Взаимодействие между промежуточными и конечными серверами также осуществляется по модели клиент/сервер, так что система промежуточного уровня одновременно выступает в роли сервера и клиента.

2.4 Промежуточное программное обеспечение

Трехуровневая архитектура клиент/сервер часто применяются для централизованной реализации в сети некоторых общих для распределенных приложений функций, отличных от файлового сервиса и управления базами данных.

Программные модули, выполняющие такие функции, относят к классу **middleware** - то есть **промежуточному ПО**, располагающемуся между индивидуальной для каждого приложения логикой и сервером баз данных.

Промежуточное ПО - набор стандартных программных интерфейсов и протоколов, располагающихся между **приложением и операционной системой и коммуникационным программным обеспечением**.

При наличии стандартного программного интерфейса упрощается реализация одного и того же приложения на серверах и рабочих станциях разных типов.

Стандартизованные протоколы необходимы для связи различных серверных интерфейсов с клиентами, которым необходим доступ к ним.

С точки зрения логики распределенная система в целом может

рассматриваться как множество доступных пользователю приложений и ресурсов. Пользователя не должно интересовать физическое размещение данных и приложений. Все приложения работают посредством однотипного API.

Промежуточное программное обеспечение, охватывающее все платформы и сети клиентов и серверов, отвечает за маршрутизацию запросов клиентов соответствующему серверу.

Имеется большое количество пакетов промежуточного программного обеспечения — от самых простых до очень сложных. Однако объединяет их то, что они способны скрыть сложности и отличия различных сетевых протоколов и операционных систем. Разработчики клиентского и серверного программного обеспечения обычно предоставляют возможность выбора при работе одного из | нескольких наиболее популярных пакетов промежуточного программного обеспечения.

Эти средства помогают улучшить качество взаимодействия клиентов с серверами за счет промышленной реализации достаточно важных и сложных функций, а также упорядочить поток запросов от множества клиентов к множеству серверов, играя роль регулировщика, распределяющего нагрузку на серверы.

Сервер приложений должен базироваться на мощной аппаратной платформе:

мультипроцессорные системы,

специализированные кластерные архитектуры

ОС сервера приложений должна обеспечивать высокую производительность вычислений, а значит, поддерживать

- многопоточную обработку,
- вытесняющую многозадачность,
- мультипроцессирование,
- виртуальную память

- и наиболее популярные прикладные среды.

2.5 Описание архитектуры платформы корпоративной системы документооборота

2.5.1 Обзор платформы

OpenText Enterprise Process Services (далее EPS) – платформа электронного документооборота для управления бизнес процессами.

EPS позволяет автоматизировать бизнес процессы. Встроенный дизайнер (Process Designer) позволяет легко графически представить бизнес процесс. При необходимости можно добавить всю необходимую для функционирования автоматизированного бизнес процесса информацию.

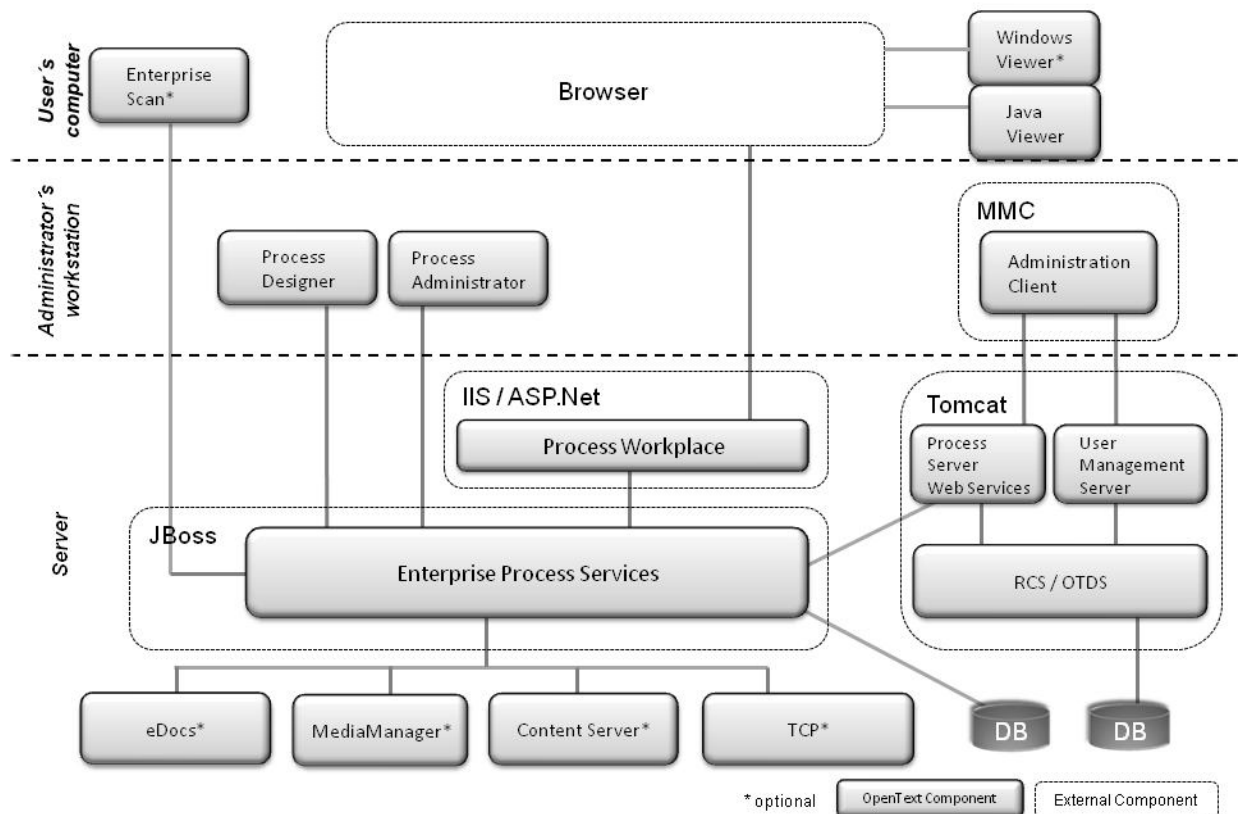
2.5.2 Управление бизнес процессами (Business Process Management) –

Основными шагами внедрения BPM является:

- Анализ
- Проектирование
- Внедрение
- Контроль
- Управление
- Автоматизация

В зависимости от сложности бизнес процесса и задействованных участников автоматизация может быть различной.

2.5.3 Основные компоненты платформы OpenText EPS



2.5.3.1 Database – Microsoft SQL Server 2008 SP1 / Oracle 11.1. / Oracle 11.2.

База данных – один из основных компонентов системы. Хранение всей информации платформы – метаданных системы, пользовательских данных.

Далее приведена инструкция по установке и настройке базы данных, см. «Инсталляция и настройка базы данных».

2.5.3.2 Open Text Process Server

Компонент в целом управляет платформой. Через данный компонент происходит назначение, управление, контроль над всеми бизнес процессами, работающими в системе. Также компонент осуществляет ведение логов работы. ProcessServer состоит из следующих компонентов:

- WebServices. Компонент отвечает за публикацию ProcessorServerAPI (интерфейсов программирования приложений) в виде веб сервиса.
- ReportServer. Компонент отвечает за создание и работу отчетов.

2.5.3.3 OpenText Process Administrator

Компонент, с помощью которого происходит администрирование всех настроек системы, управление процессами и т.д.

2.5.3.4 OpenText Process Designer

Компонент, с помощью которого разрабатываются процессы и далее публикуются на сервер приложений EPS.

2.5.3.5 OpenText Process Report Builder

Компонент, с помощью которого происходит создание и настройка отчетов.

2.5.3.6 OpenText Process Workplace

Компонент, обеспечивающий пользователям веб-доступ к бизнес процессам, отчетам, запросам и т.д.

2.5.3.7 OpenText User Management Server

UMS – сервис в котором ведутся следующие справочники:

- Организационная структура Компании
- Список пользователей
- Список и состав бизнес ролей и бизнес групп

2.5.3.8 OpenText Administration Client

Приложение, с помощью которого администратор работает со справочниками UMS.

2.5.3.9 OpenText Imaging Windows Viewer

Модуль, ответственен за визуальную работу ProcessWorkplace.

2.5.3.10 OpenText Imaging Java Viewer

Данный модуль помогает организовать просмотр файлов непосредственно через веб-браузер.

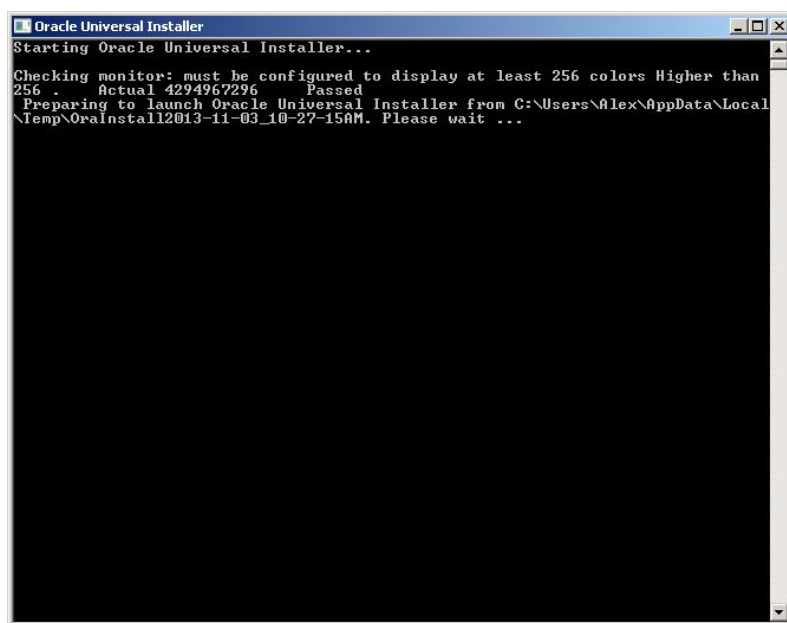
2.5.3.11 OpenText Imaging Enterprise Scan

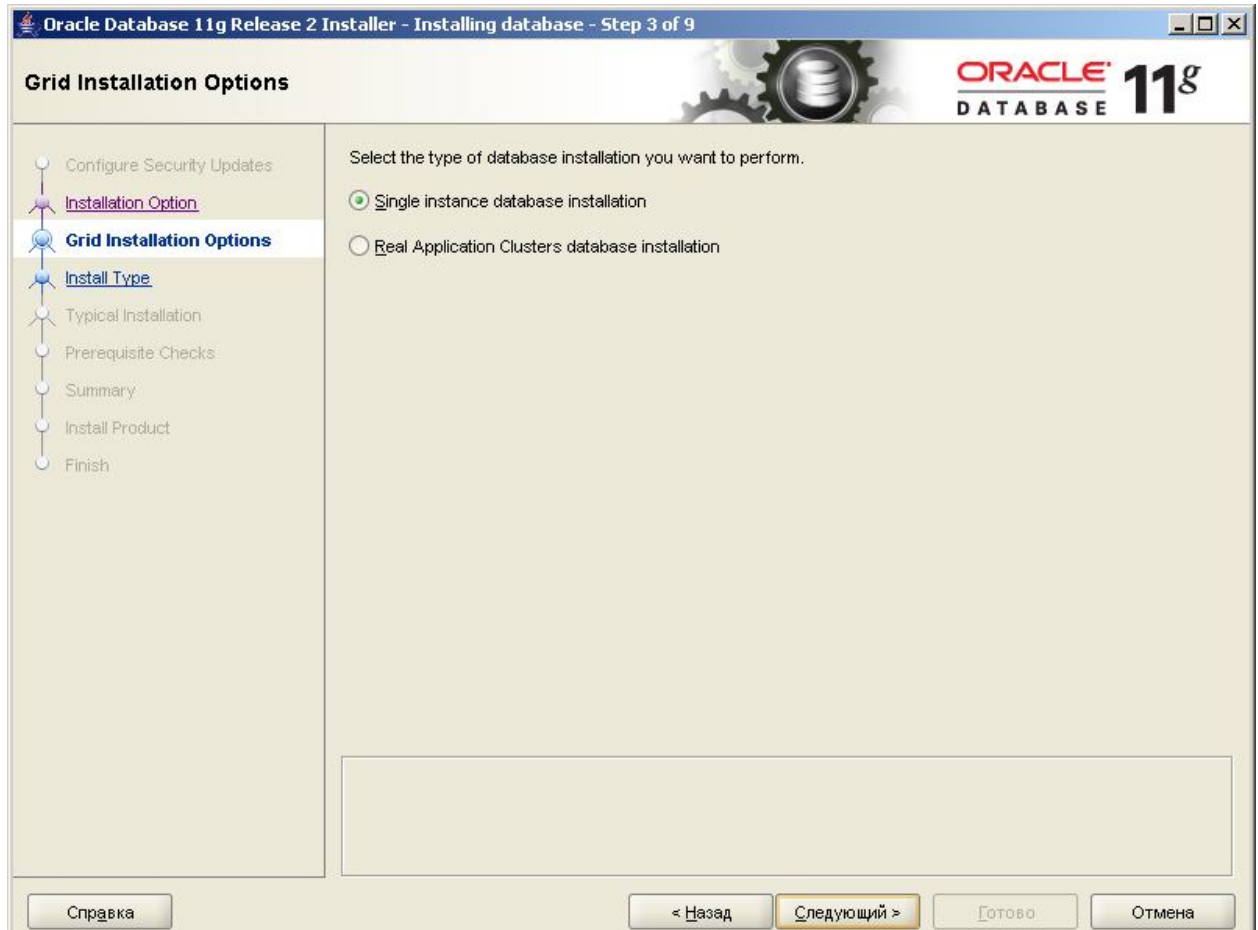
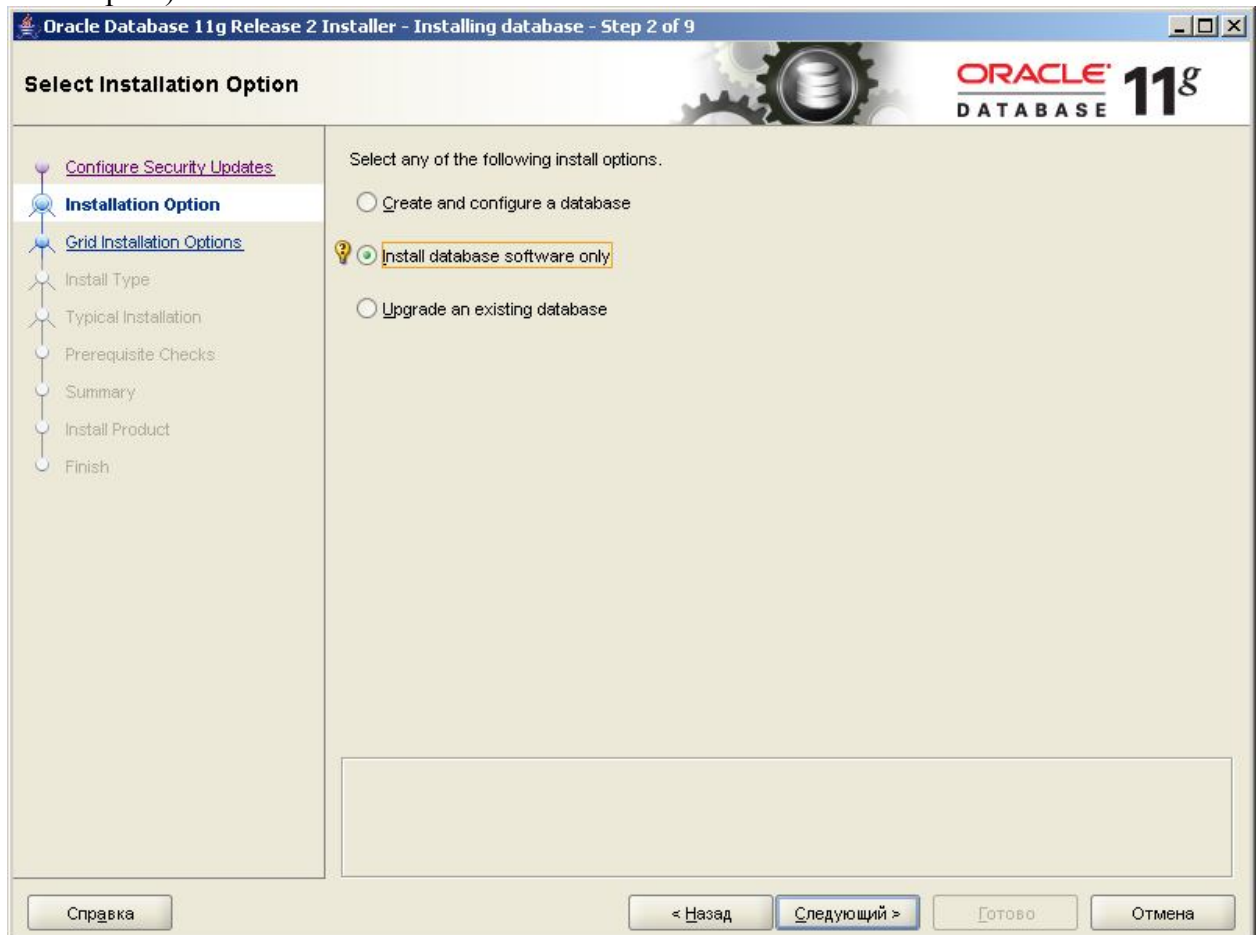
Модуль позволяющий сканировать документы и добавлять их к бизнес процессам в виде приложений.

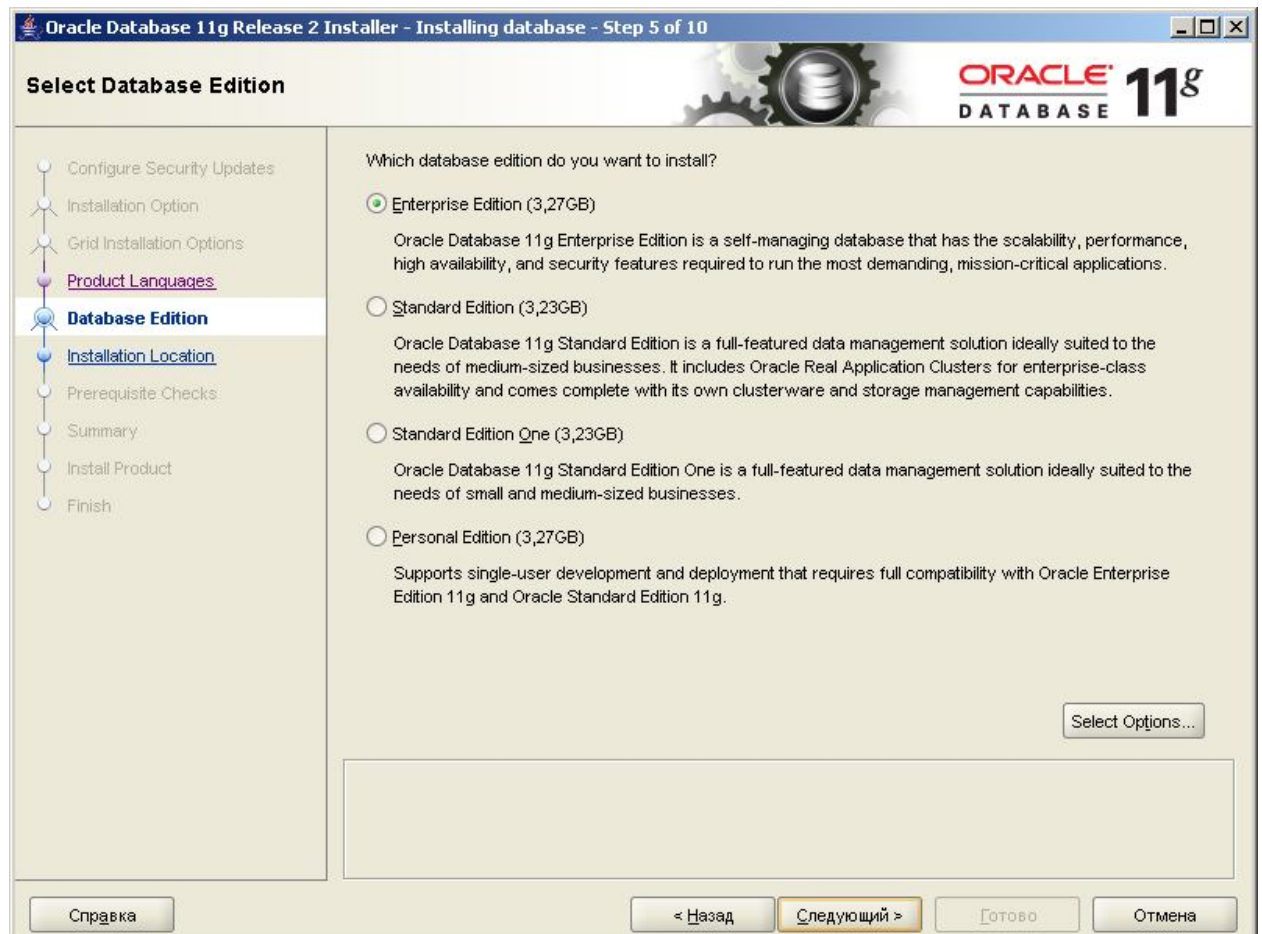
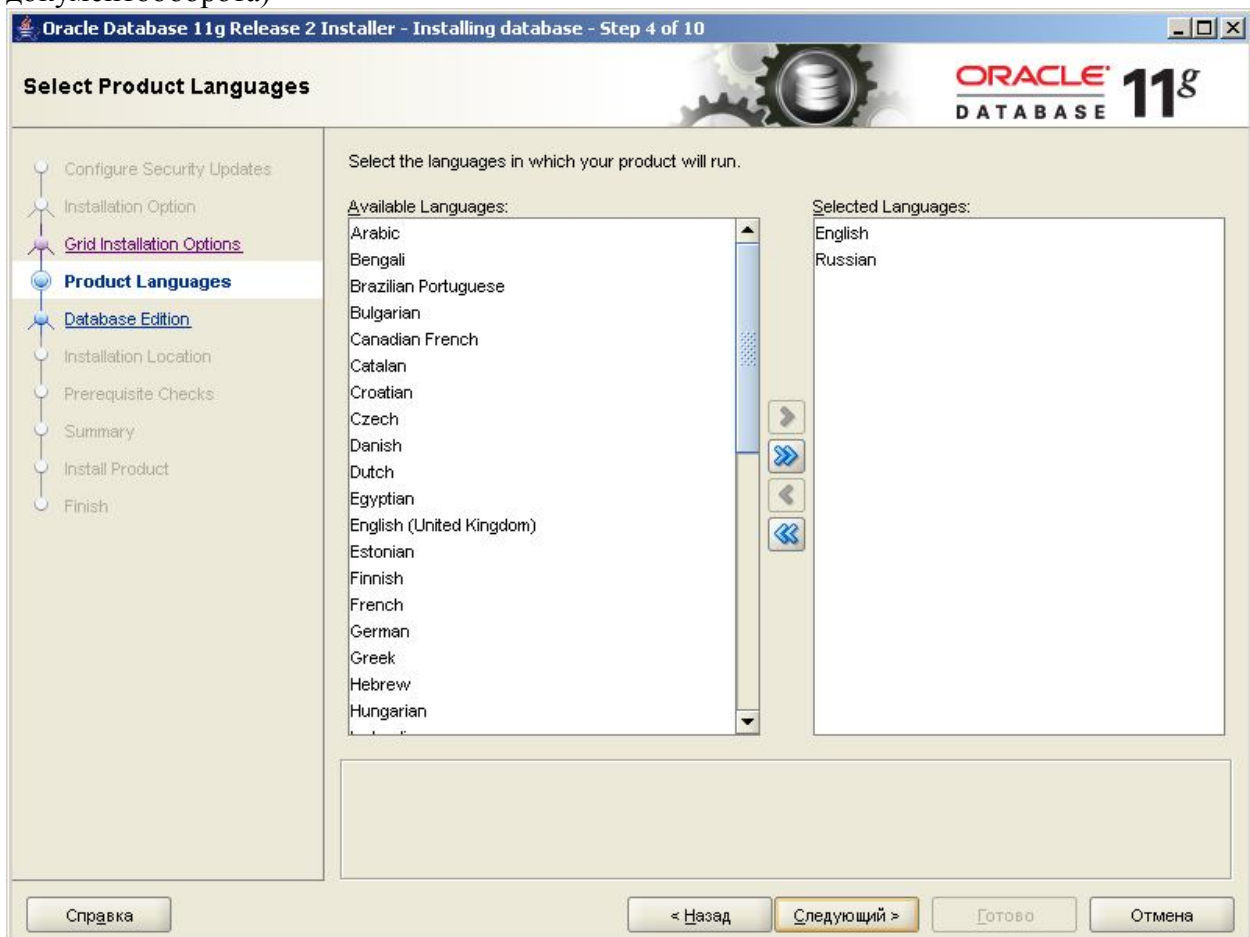
2.5.4 Инсталляция и настройка базы данных

В качестве базы данных наиболее оптимальным вариантом является СУБД Oracle 11.2

Ниже приведена инструкция по установке и настройке базы данных







Oracle Database 11g Release 2 Installer - Installing database - Step 6 of 10

Specify Installation Location

Specify an Oracle base path to place all Oracle software and configuration-related files. This location is the Oracle base directory.

Oracle Base:

Specify a location for storing Oracle software files. This location is the Oracle home directory.

Software Location:

Справка

Oracle Database 11g Release 2 Installer - Installing database - Step 7 of 10

Perform Prerequisite Checks

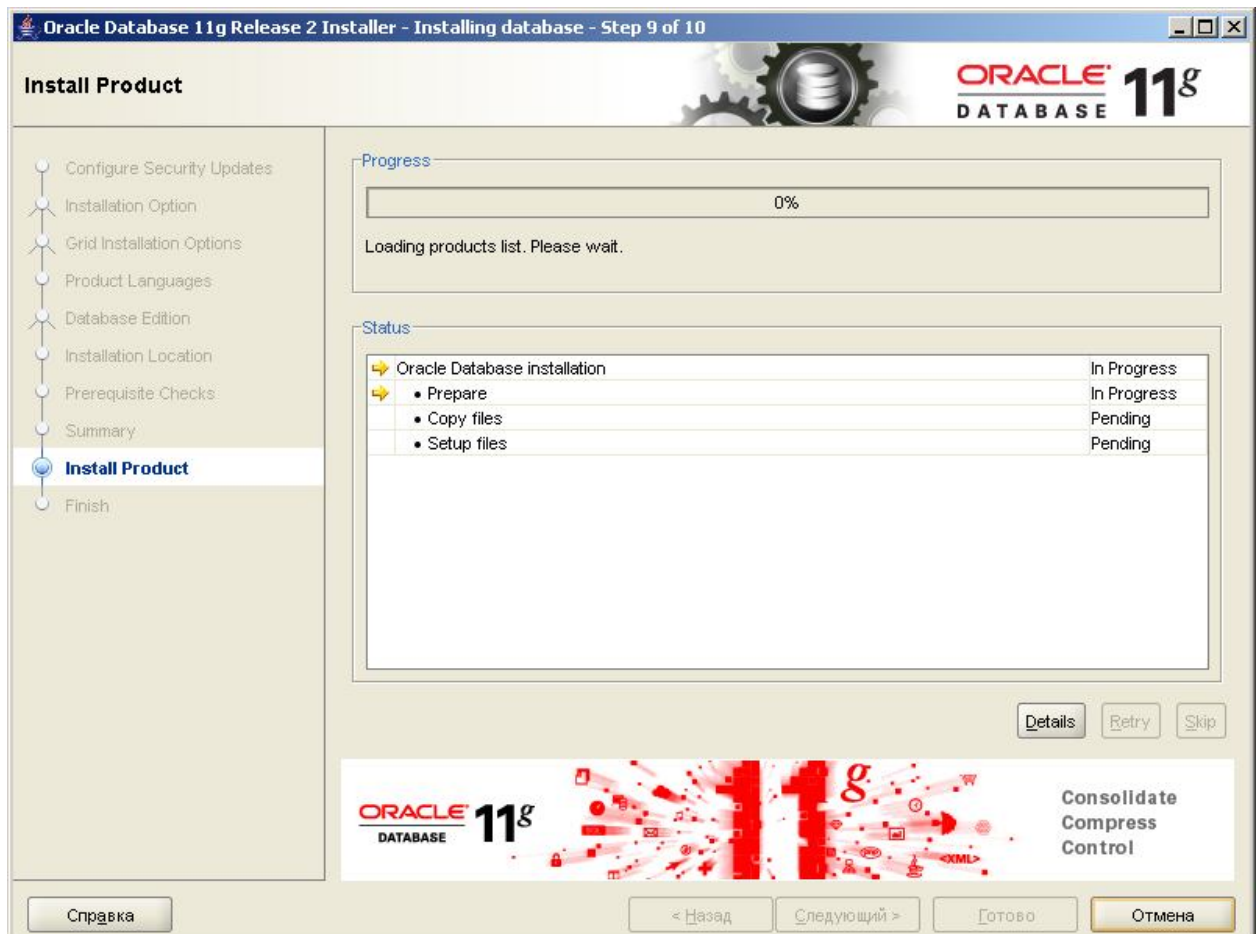
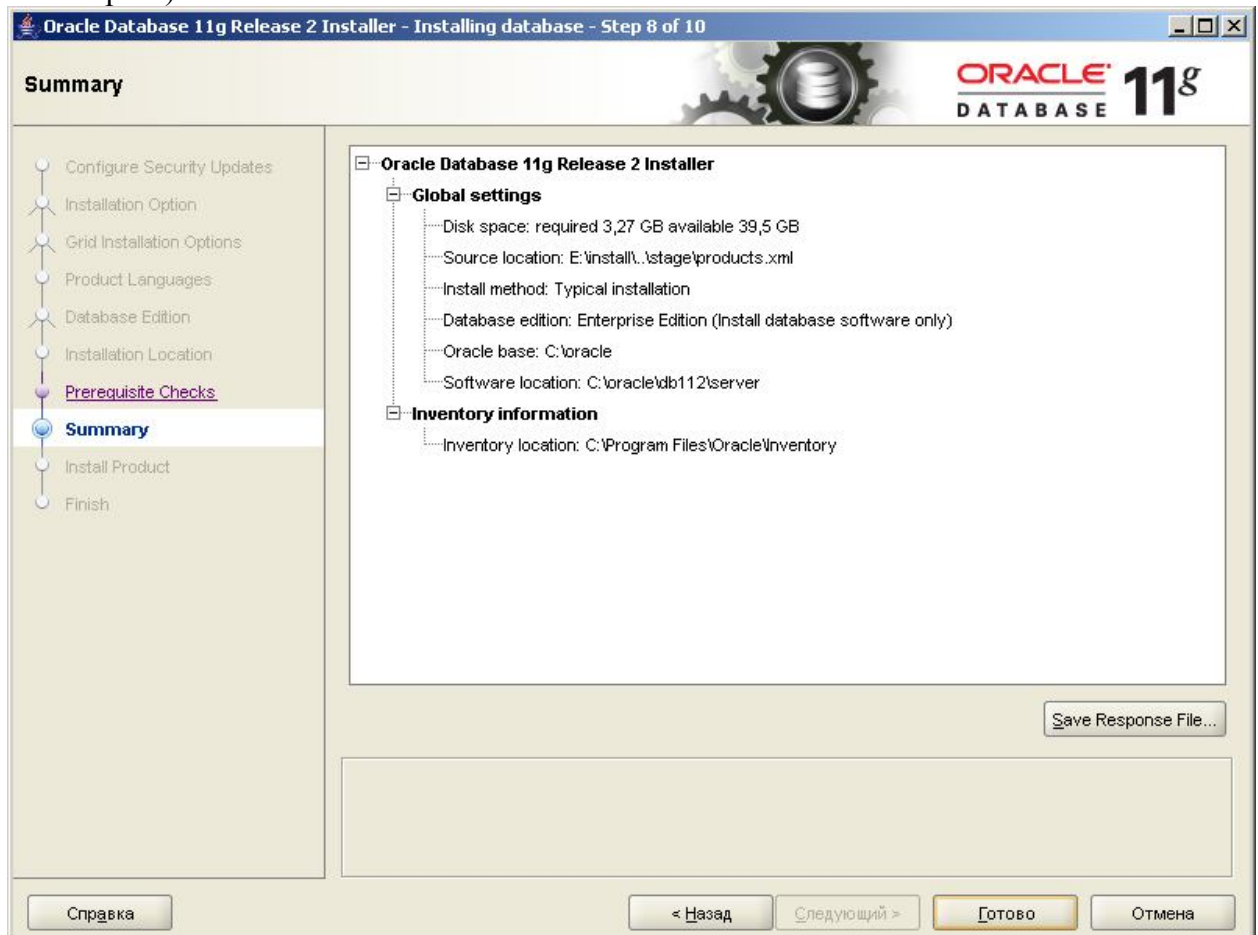
Some of the minimum requirements for installation are not completed. Review and fix the issues listed in the following table, and recheck the system.

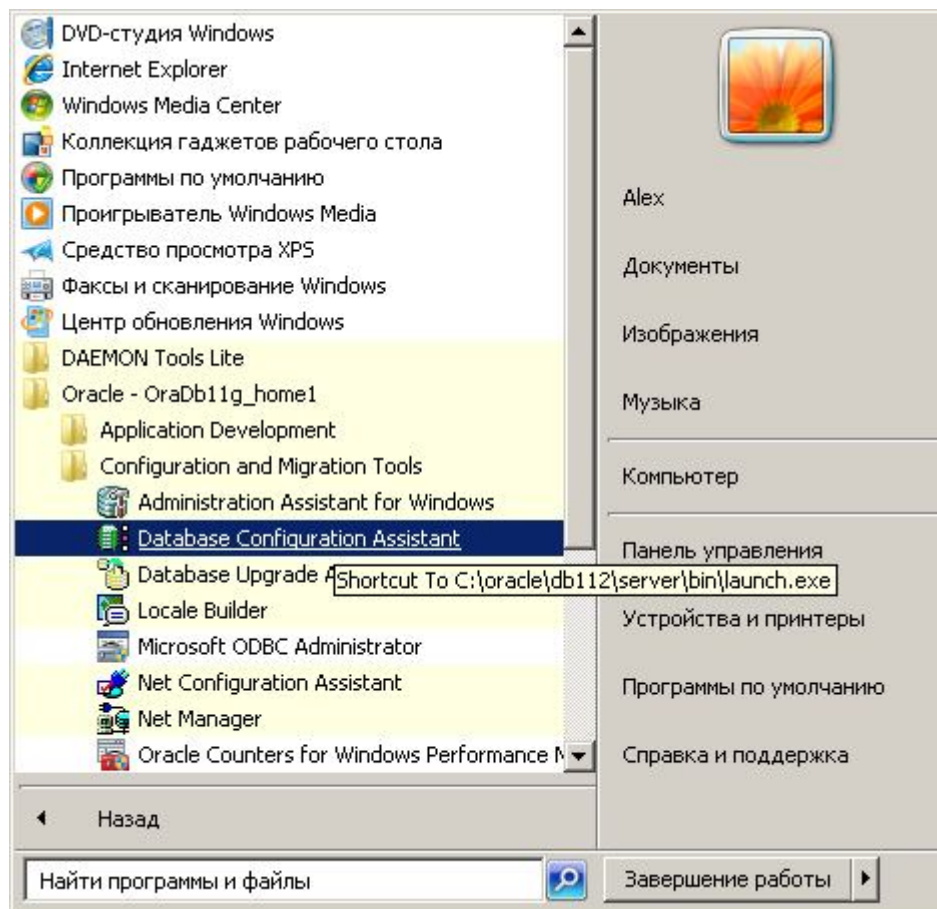
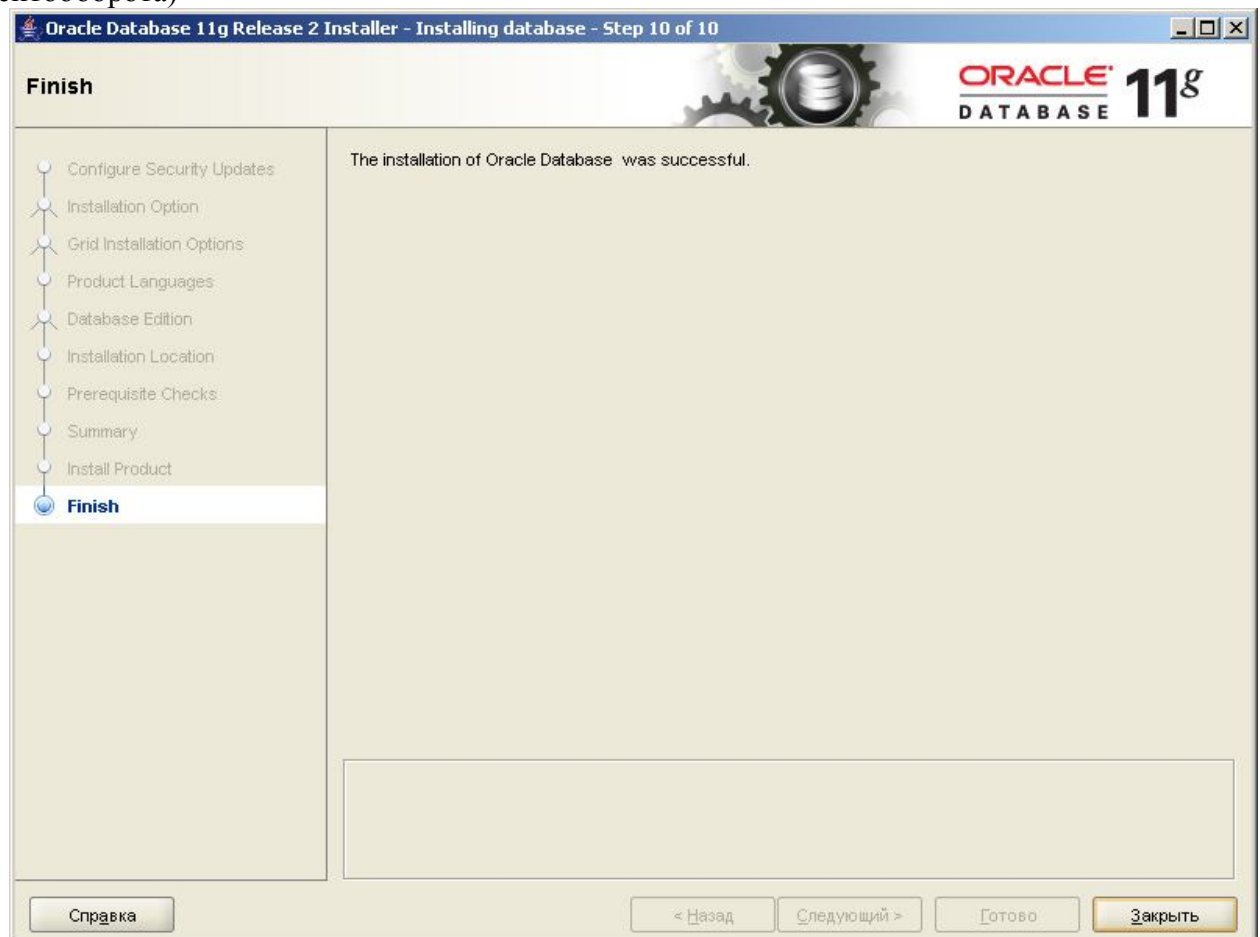
☒

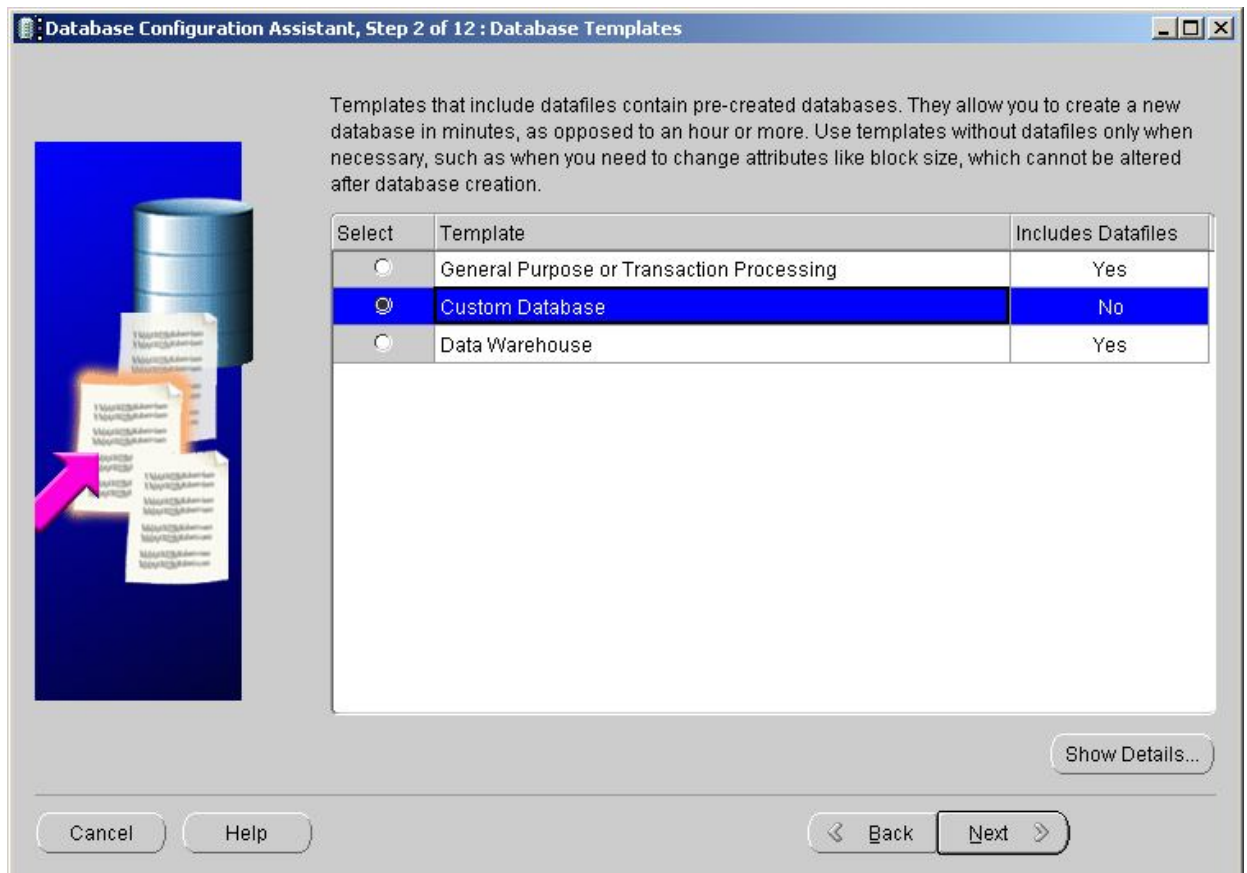
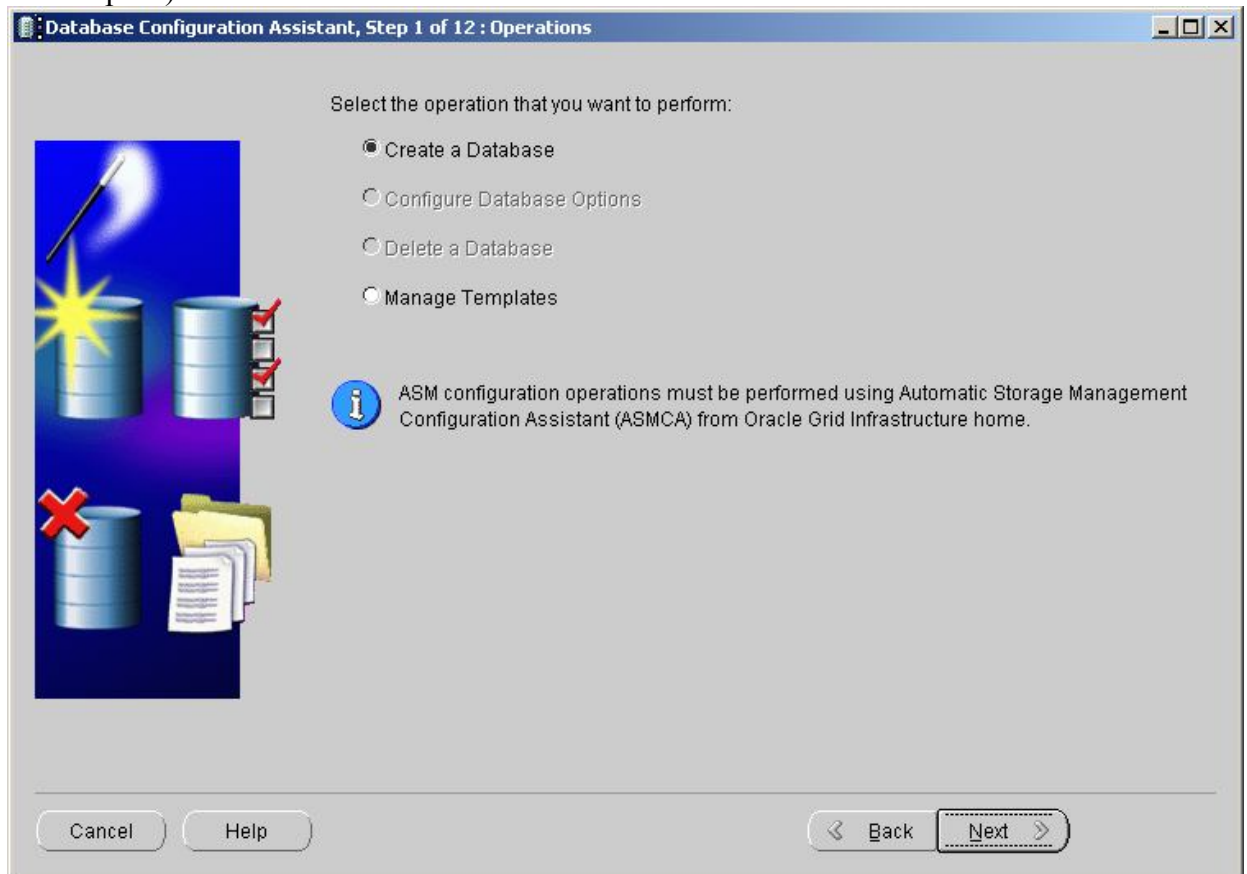
Checks	Status	Fixable
Checks		
Physical Memory	Ignored	
Available Physical Memory	Ignored	
Swap Size	Ignored	
Free Space		
Free Space: Alex-ПК: C:\Users\Alex\AppData\Local\Temp	Ignored	
Architecture	Ignored	
Environment variable: "PATH"	Ignored	

This is a prerequisite condition to test whether the system has at least 922MB (944128.0KB) of total physical memory. [\(more details\)](#)
 Expected Value : n/a
 Actual Value : n/a


Справка







Database Configuration Assistant, Step 3 of 12 : Database Identification



An Oracle database is uniquely identified by a Global Database Name, typically of the form "name.domain".


Global Database Name: EPS

A database is referenced by at least one Oracle instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID).

SID: EPS

Cancel Help < Back Next >

Database Configuration Assistant, Step 4 of 12 : Management Options



Enterprise Manager Automatic Maintenance Tasks

☒ Configure Enterprise Manager

☐ Register with Grid Control for centralized management

Management Service: No Agents Found

☒ Configure Database Control for local management

☐ Enable Alert Notifications

Outgoing Mail (SMTP) Server:

Recipient Email Address:

☐ Enable Daily Disk Backup to Recovery Area

Backup Start Time: 02 00

OS Username:

OS Password:

Cancel Help < Back Next >

Database Configuration Assistant, Step 5 of 12 : Database Credentials

For security reasons, you must specify passwords for the following user accounts in the new database.

☐ Use Different Administrative Passwords

User Name	Password	Confirm Password
SYS		
SYSTEM		
DBSNMP		
SYSMAN		

☒ Use the Same Administrative Password for All Accounts

Password:

Confirm Password:

Cancel Help < Back Next >

Database Configuration Assistant, Step 6 of 12 : Database File Locations

Specify storage type and locations for database files.

Storage Type:

Storage Locations:

☒ Use Database File Locations from Template

☐ Use Common Location for All Database Files

Database Files Location: Browse...

☐ Use Oracle-Managed Files

Database Area: Browse...

i If you want to specify different locations for any database files, pick any of the above options except Oracle-Managed Files and use the Storage page later to customize each file location. If you use Oracle-Managed Files, Oracle automatically generates the names for database files, which can not be changed on the Storage page.

Cancel Help < Back Next > Finish

Database Configuration Assistant, Step 7 of 12 : Recovery Configuration

Choose the recovery options for the database:

☐ Specify Flash Recovery Area

This is used as the default for all disk based backup and recovery operations, and is also required for automatic disk based backup using Enterprise Manager. Oracle recommends that the database files and recovery files be located on physically different disks for data protection and performance.

Flash Recovery Area:

Flash Recovery Area Size:

☒ Enable Archiving

Standard Database Components

Select the standard database components you want to configure for use in your database. Oracle recommends that you always install these components in your database. Deselecting these components may cause you to no longer be able to choose some components on the previous page.


☐ Oracle JVM

☐ Oracle XML DB

☐ Oracle Multimedia

☐ Oracle Application Express

Database Configuration Assistant, Step 8 of 12 : Database Content



Database Components


Select the components you want to configure for use in your database. Provide the tablespace in which you want the component to reside. Components which appear disabled are either not installed or depend on components which are not selected.

<input type="checkbox"/> Oracle Text	<input type="text" value="SYSBAUX"/>
<input type="checkbox"/> Oracle OLAP	<input type="text" value="SYSBAUX"/>
<input type="checkbox"/> Oracle Spatial	<input type="text" value="SYSBAUX"/>
<input type="checkbox"/> Oracle Label Security	<input type="text" value="SYSTEM"/>
<input type="checkbox"/> Sample Schemas	<input type="text" value="SYSBAUX"/>
<input checked="" type="checkbox"/> Enterprise Manager Repository	<input type="text" value="SYSBAUX"/>
<input type="checkbox"/> Oracle Warehouse Builder	<input type="text" value="SYSBAUX"/>
<input type="checkbox"/> Oracle Database Vault	<input type="text" value="SYSBAUX"/>
<input type="checkbox"/> Oracle Database Extensions for .NET	<input type="text" value="SYSBAUX"/>

[Standard Database Components...](#)

Cancel Help < Back Next > Finish

Database Configuration Assistant, Step 9 of 11 : Initialization Parameters



Memory

☐ Typical

Memory Size (SGA and PGA): MB 300 MB 1523 MB

☒ Use Automatic Memory Management [Show Memory Distribution...](#)

☒ Custom

Memory Management

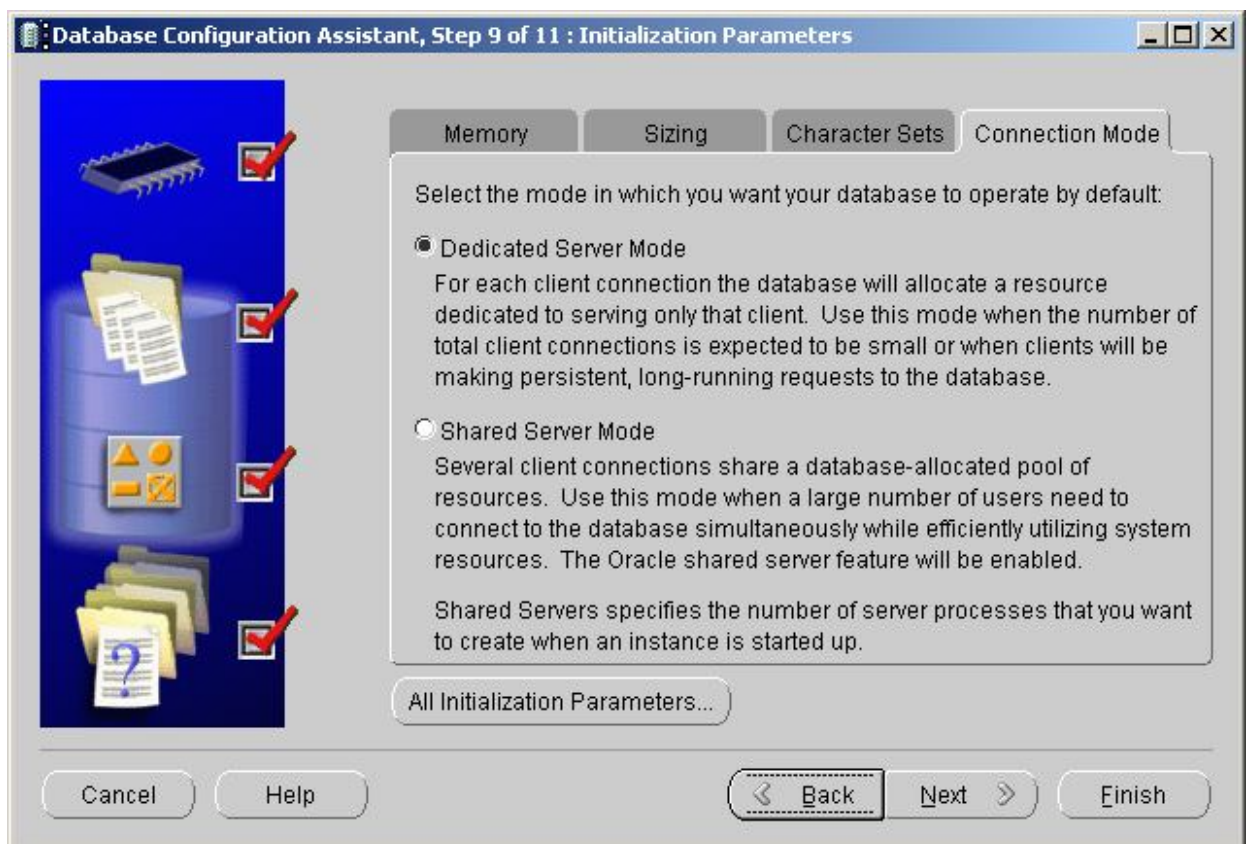
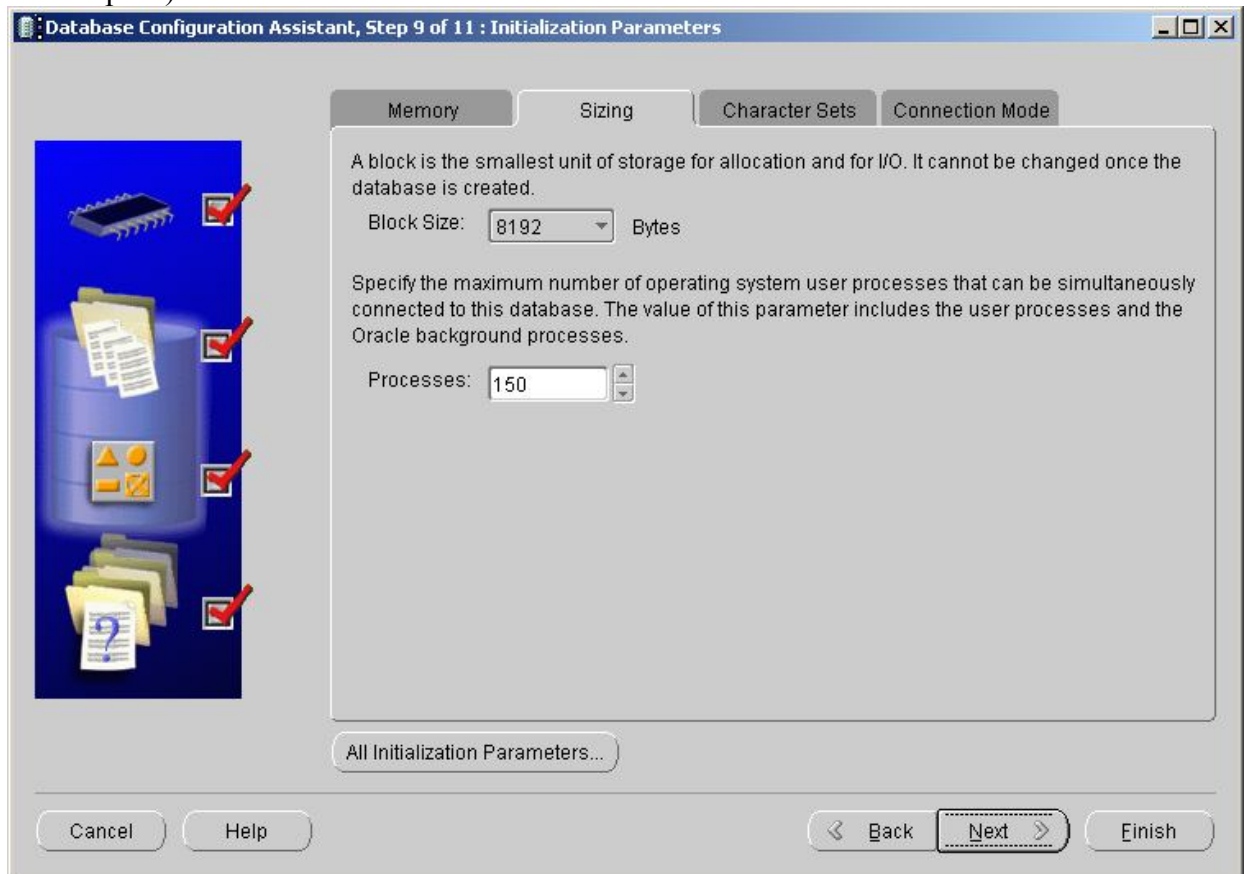
SGA Size: M Bytes

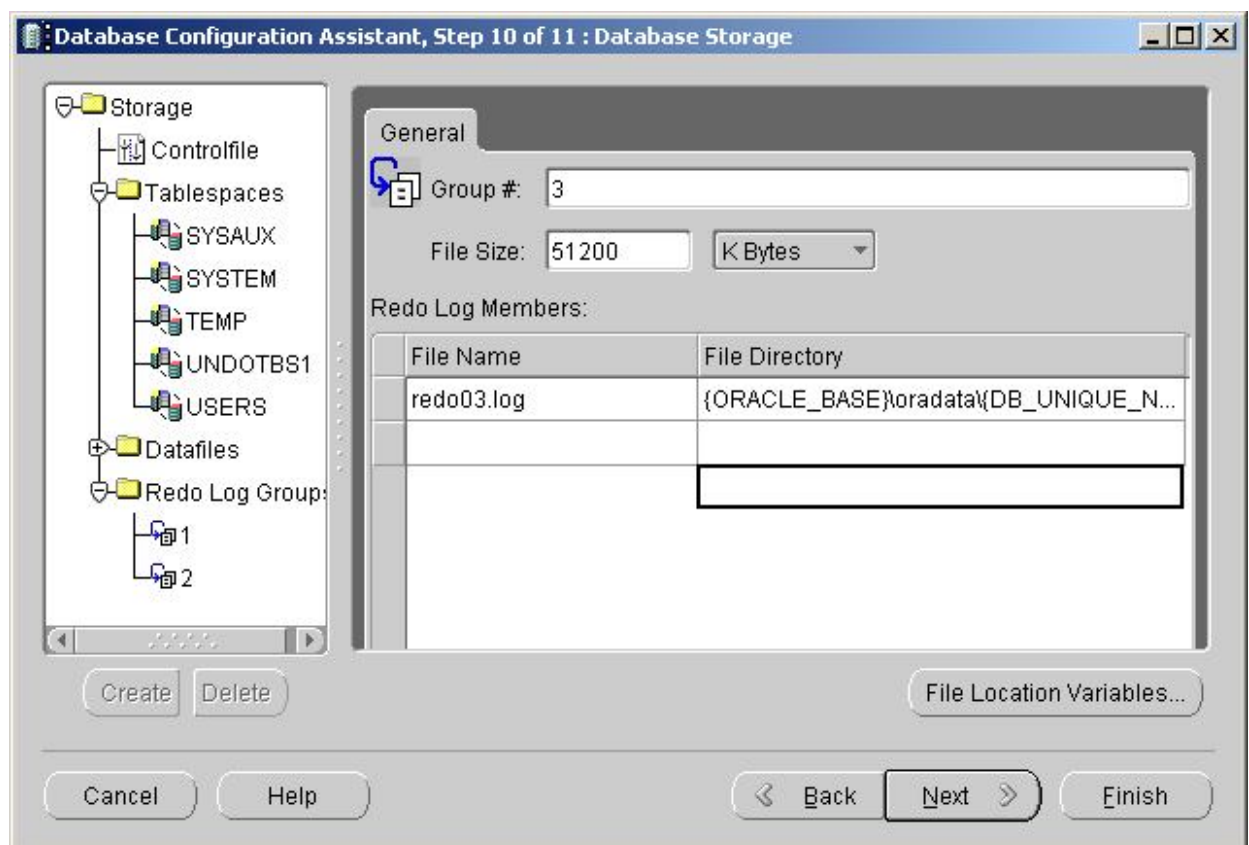
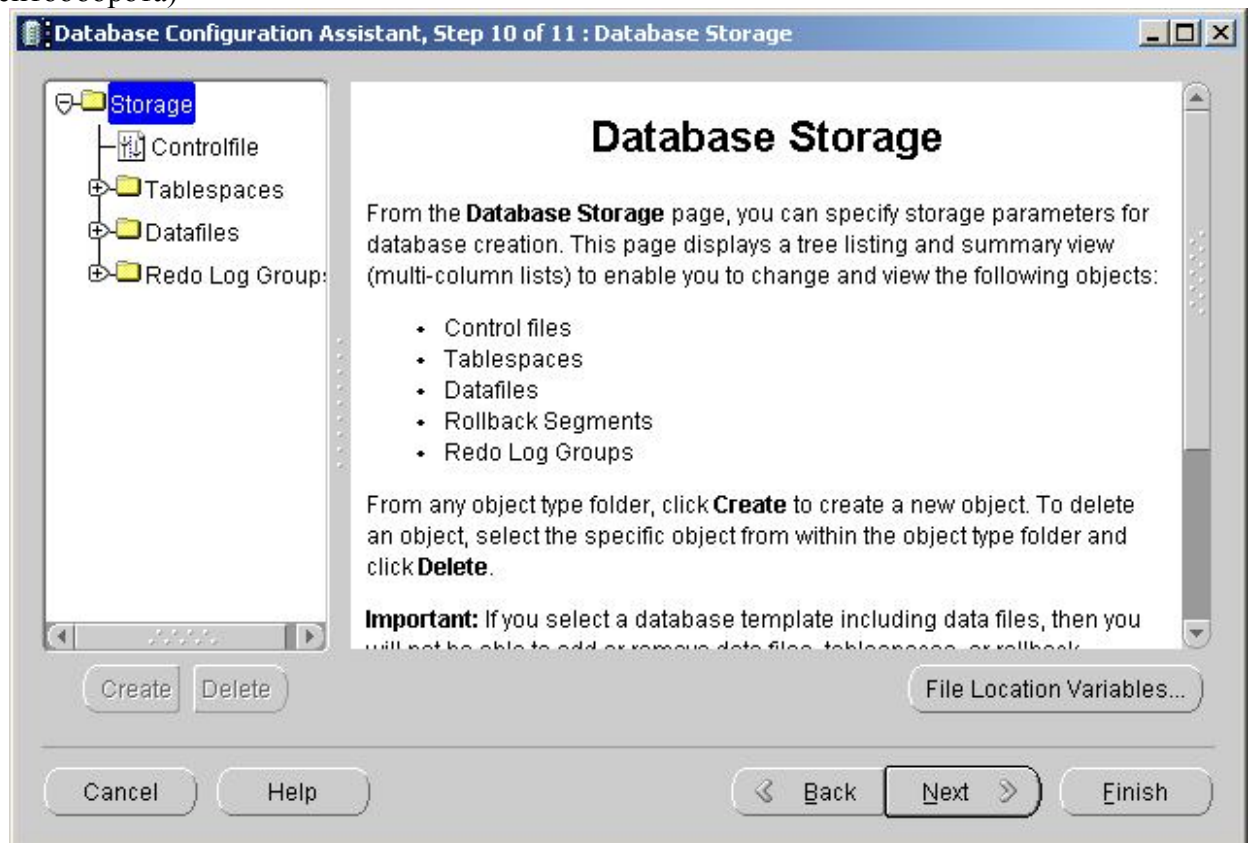
PGA Size: M Bytes

Total Memory for Oracle: 608 M Bytes

[All Initialization Parameters...](#)

Cancel Help < Back Next > Finish





Confirmation

The following operations will be performed:
A database called "EPS" will be created.

Database Details:

Create Database - Summary

Database Configuration Summary

Global Database Name: EPS

Database Configuration Type: Single Instance

SID: EPS

Management Option Type: Database Control

Storage Type: File System

Memory Configuration Type: Automatic Shared Memory Management

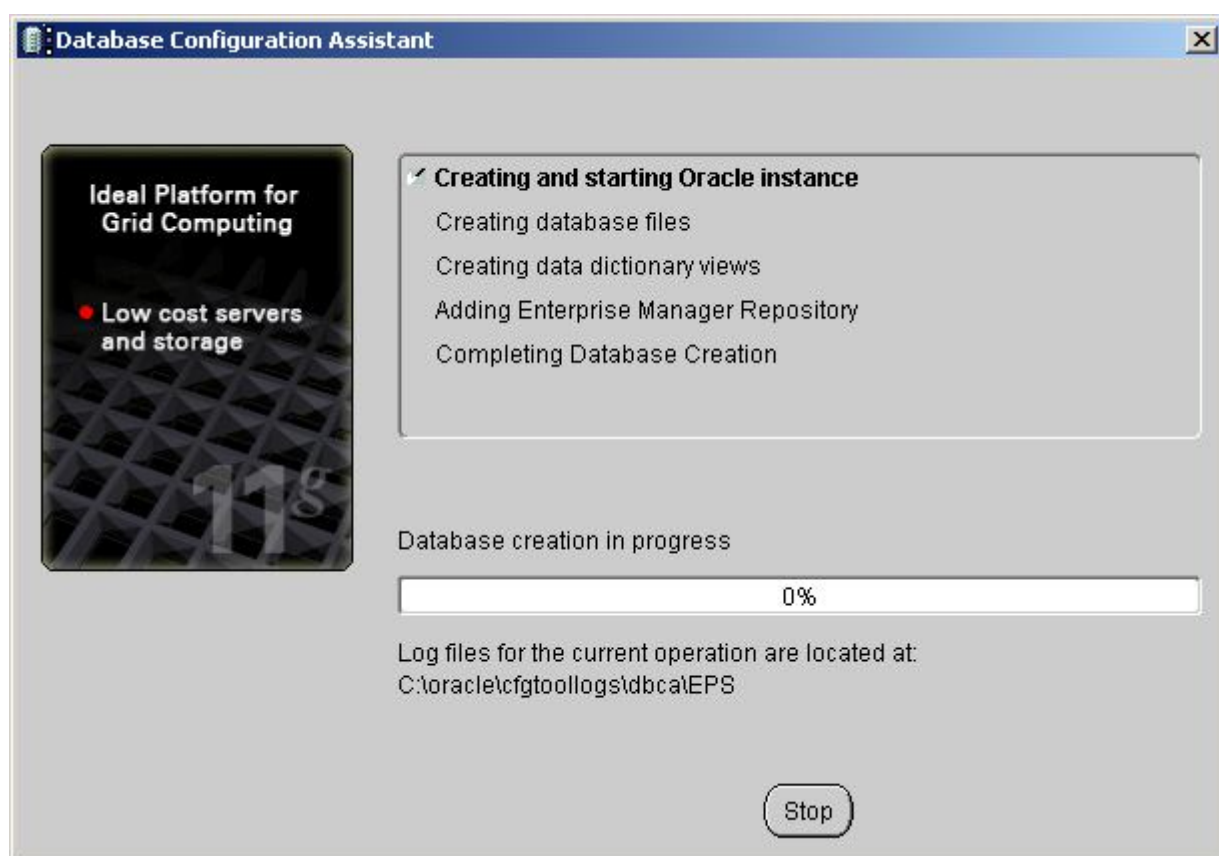
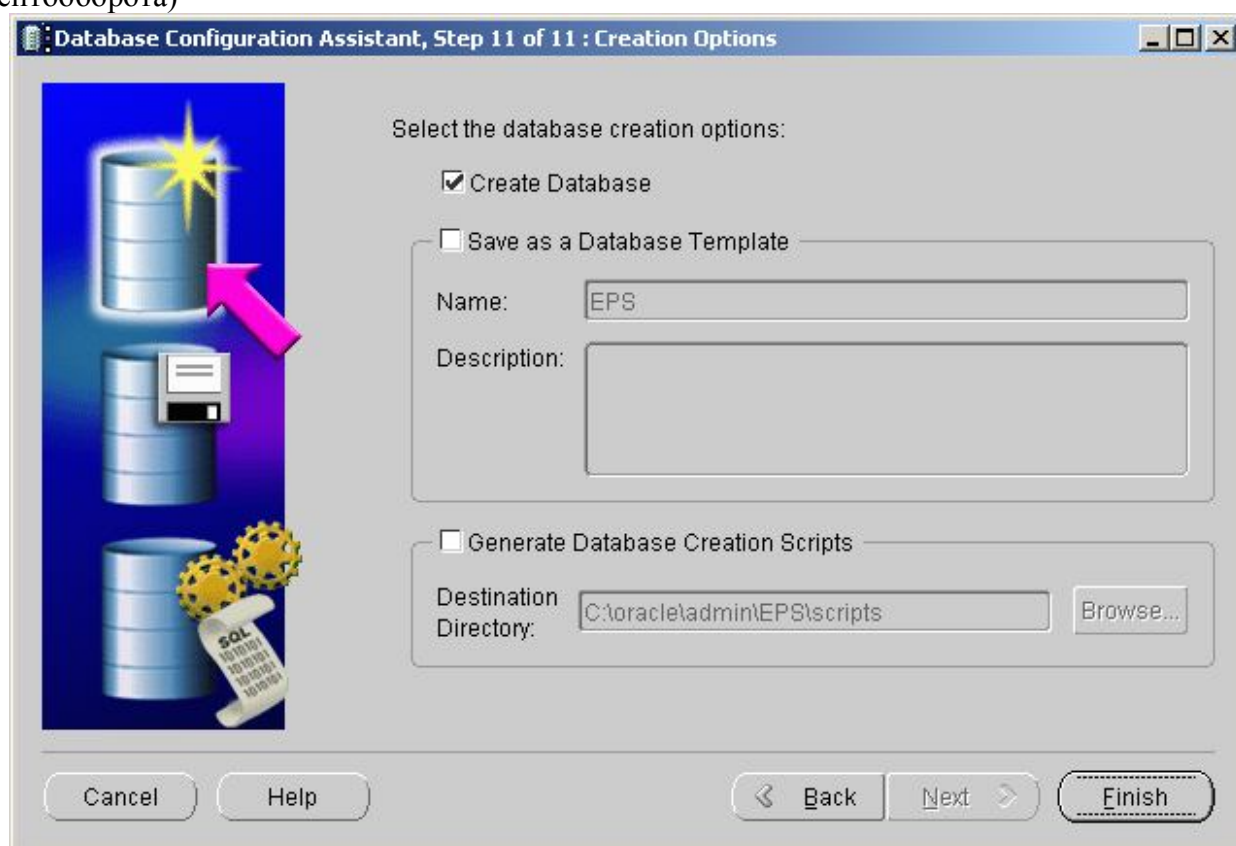
Database Configuration Details

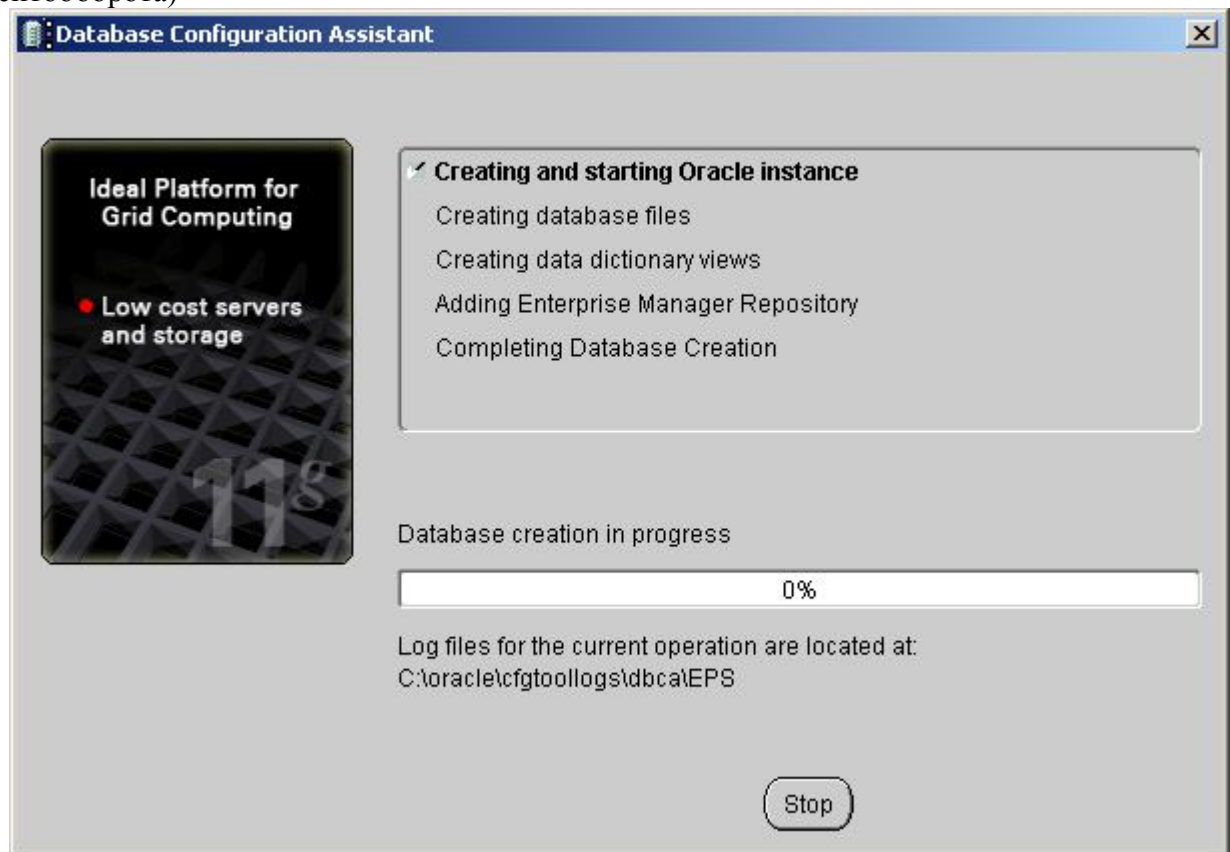
Database Components

Component	Selected
Oracle JVM	false
Oracle Text	false

Save as an HTML file...

OKCancelHelp





3 Задание на выполнение работы

1. Провести инсталляцию и настройку СУБД Oracle 11.2.
2. Продемонстрировать работоспособность установленного ПО.
3. Ответить на контрольные вопросы.

4 Контрольные вопросы

1. Какова роль клиентов и серверов в архитектуре клиент/сервер?
2. Назовите функциональные части модели разбиения распределённого приложения.
3. Перечислите классы приложений клиент/сервер
4. В чём отличие моделей «тонкого клиента» и «толстого клиента»?
5. В чём особенность трёхуровневой архитектуры клиент/сервер?
6. Что такое промежуточное программное обеспечение?
7. К какому классу приложений относится система документооборота OpenText Enterprise Process Services (EPS)?

5 ЛИТЕРАТУРА

1. Джин Бэкон, Тим Харрис. Операционные системы Параллельные и распределенные системы. СПб.: Питер, 2004. 800 с.: ил.
2. Х.М. Дейтел, П. Дж. Дейтел, Д.Р. Чофнес Операционные системы. Распределенные системы, сети и безопасность: Третье издание. Пер. с англ. – М.: ООО «Бином-Пресс», 2006 г. – 704 с.: ил.