

IBM Informix



IBM Informix Virtual Appliance User's Guide

Version 1.9

IBM Informix



IBM Informix Virtual Appliance User's Guide

Version 1.9

Note

Before using this information and the product it supports, read the information in "Notices" on page 43.

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 2008, 2009.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Welcome to the IDS Virtual Appliance . . .	1
Installing and Running the IDS Virtual Appliance . . .	4
Installing the IDS Virtual Appliance in a VMware Infrastructure Environment	5
First Steps	7
Configuring the IDS Appliance	8
MACH11 Cluster (IDS Enterprise/Developer Edition only)	13
OpenAdmin Tool (OAT) version 2.24 for IDS . . .	14
Scripts	16
Using the Informix JDBC Driver and the JDBC Common Client	17
Using PHP with IDS	18
The IDS Detective Game	20
Introduction to Data Studio	31
Enhanced Error Message Lookup Using the Basic Text Search (BTS) Extension	34
Using the IBM Informix Spatial DataBlade Module . . .	37
Appendix. Accessibility	41
Accessibility features for IBM Informix Dynamic Server	41
Accessibility Features	41
Keyboard Navigation	41
Related Accessibility Information	41
IBM and Accessibility	41
Notices	43
Trademarks	45
Index	47

Welcome to the IDS Virtual Appliance



IDS 11.50.xC4 / SLES10 SP2 Virtual Appliance v1.9

A virtual appliance is a pre-configured virtual desktop environment that is managed using tools from VMware. The virtual appliance runs either locally on your host computer or remotely on a VMware Infrastructure environment. The IBM® Informix® Dynamic Server (IDS) Virtual Appliance features IDS and an assortment of development and productivity tools that are used to:

- Develop new applications
- Create IDS demonstrations and presentations
- Develop and test IDS applications

Launching the IDS virtual appliance starts a SUSE Linux® Enterprise Server instance that is pre-configured with IDS 11.50.xC4, a fully configured high-availability cluster (MACH11) environment, and a host of productivity and development tools.

The IBM Informix Dynamic Server (IDS) 11.50 SUSE Linux Enterprise Server (SLES) 10 SP2 virtual appliance contains all the components that you need to start testing and developing software with IDS 11.50.

Post any questions or comments related to the virtual appliance in the IBM IDS DeveloperWorks Forum.

The edition of Informix Dynamic Server (IDS) included with the virtual appliance depends on the IDS virtual appliance you purchased. This documentation supports all IDS editions. Notes in the documentation indicate the IDS edition required for certain advanced features. The IDS editions supported by this documentation are:

Table 1. Informix versions

Informix Dynamic Server Express™ Edition	Full-function, object-relational data server. Includes important capabilities such as: high reliability, security, usability, manageability and performance.
Informix Dynamic Server Workgroup Edition	Includes all of the features of IDS Express plus features to handle high transaction volume.

Table 1. Informix versions (continued)

Informix Dynamic Server Enterprise Edition	Includes all of the features of IDS Workgroup Edition plus features required to provide the scalability to handle high user loads and provide 24x7x365 high availability, including Enterprise Replication (ER), High-Availability Data Replication (HDR), and global availability with HDR Multiple Remote Secondary Servers.
Informix Dynamic Server Developer Edition	A no-charge edition of IDS licensed for development use. It is ideal for application development and prototyping for developers and business partners delivering solutions to businesses of all sizes. The IDS Developer Edition includes all core features of the IDS enterprise edition, including the latest active-active cluster solution for high availability, but with some memory and usage limitations described in the license.

System Requirements

Depending on your hardware and operating system, the IDS 11.50/SLES10 SP2 virtual appliance requires certain VMware software:

- For UNIX®, Linux, or Windows®, either VMware Workstation 6.x or later or VMware Player 2.0.3, 2.5 or later must be installed on your host machine.
- For Mac OS X systems VMware Fusion is required.
- For remote access to VMware ESX or ESXi host machines, the VMware Infrastructure client is required.

You can obtain VMware Player here .

The recommended amount of VMware Player memory allocation is 768 MB.

The virtual appliance provides a pre-configured, compressed virtual disk of 12 GB (maximum size), mounted as */*. The appliance also comes with a compressed 4 GB virtual disk mounted as */data*. These disks will grow as needed during use of the appliance. See “Changing the VMware Memory Configuration” on page 11 if you experience performance problems.

System User IDs and Passwords

Three user accounts are predefined:

- user **root** (password: **root**): privileged SUSE Linux admin account
- user **informix** (password: **informix**): the dedicated Informix database administrator (DBA) account
- user **developer** (password: **developer**): regular, non-privileged user account

The command **su** or **su -** lets you temporarily obtain root (super user) permissions. To switch between the three user accounts, use the **su - username** command. For example, to start or stop one of the IDS demonstration instances as user **informix**, invoke a subshell as user **informix** by typing **su - informix**. Then run the appropriate scripts to create, start, stop, or remove IDS demonstration instances. When finished, exit the subshell by typing **exit**.

Installed Software

The following products are pre-installed, pre-configured, and ready to run:

Table 2. Installed software

Product	Description
IBM Informix Dynamic Server 11.50.xC4	Informix Dynamic Server Express Edition, Workgroup Edition, Enterprise Edition, or Developer Edition, depending on the product downloaded.
IBM Informix Client SDK 3.50.xC4	A single packaging of several application programming interfaces (APIs) for rapid, cost-effective development of applications for IBM Informix servers.
IBM Informix JDBC Driver 3.50.xC4	A Java™ database connectivity (JDBC) driver -- the JavaSoft specification of a standard API that allows Java programs to access database management systems.
IBM Informix Spatial DataBlade® 8.21.xC3	Brings all the significant features and benefits of the IBM Informix Dynamic Server (IDS) to location-based data. Many key business decisions involve location and proximity. The Spatial DataBlade enables organizations to transform both traditional and location-based data into important information to help gain a competitive advantage.
IBM Informix Web DataBlade 4.13.FC3X1	A collection of tools, functions, and examples that ease development of "intelligent," interactive, Web-enabled database applications.
IBM Data Server Driver for JDBC/SQLJ 9.5	Provides support for client applications written in Java™ using JDBC.
IBM Data Server Driver for ODBC/CLI 9.5	Provides support for open source drivers for IBM Data Servers.
IBM OpenAdmin Tool for IDS 2.24	Provides the ability to administer multiple database server instances from a single location. Some tasks you can perform with OpenAdmin include: gathering and analyzing performance statistics, monitoring system health, defining and managing automated tasks through the SQL Administration API, creating and displaying performance histograms for analysis and tuning, and monitoring high availability solutions that include HDR, shared disk secondary servers, and remote standalone secondary servers. You can easily plug in your own extensions to OpenAdmin to create the functionality you need.

Table 2. Installed software (continued)

IBM Data Studio 2.1	Provides a rich set of integrated functions that allows you to perform database design, development, deployment, and management tasks. Using the workbench, you can develop and test routines, deploy data-centric web services, create and run SQL and XQuery queries, and develop Java applications that access databases or in-memory data. In addition, you can perform database administration tasks such as creating and altering database objects and managing privileges. Using the administration console, you can monitor the health and availability of databases, view dashboards, view alerts, and troubleshoot problems using expert recommendations. You can also monitor Q replication and event publishing, generate replication health reports, and perform basic replication operations. Eclipse C/C++, PHP, and Ruby on Rails plug-ins are not included.
---------------------	--

See First Steps for information about the tools and demonstration programs included with the IDS virtual appliance.

Installing and Running the IDS Virtual Appliance

You must first install VMware Player, VMware Server, or VMware Workstation on your Windows or Linux host computer. For Macintosh systems, you must install VMware Fusion. Follow the manufacturer's instructions for downloading and installing the software. After installing the appropriate VMware product, follow these instructions to set up the IDS virtual appliance.

If you are using the Amazon Machine Interface, please refer to the instructions provided with that version of the appliance.

If you use Internet Explorer to download the UNIX or Linux version of the Virtual Appliance, the file extension will be renamed from **tgz** to **tar**. Rename the file by giving it a **tgz** extension before extracting the appliance.

Installing on Windows Systems

Follow the steps below to install the appliance on Windows systems:

1. Download and save the self-extracting appliance archive .exe file to a folder on your computer.
2. Run the .exe file to extract the archive.
3. Start VMware Player or VMware Workstation.
4. Select **Open an Existing Virtual machine**.
5. Navigate to the folder in which the archive was extracted, select the virtual appliance, and click **Open**.
6. Press **Ctrl+G** to control the virtual machine. To return control to the host computer, press **Ctrl+Alt**.
7. Log in as user **root** with password **root**.

8. Accept the software license agreements.
9. After the appliance is initialized, enter **developer** in the Username field and **developer** when prompted for the password.

Installing on UNIX or Linux Systems

1. Download and save the appliance archive file in a directory on your computer.
2. Enter the following command from a shell prompt to uncompress the file:

```
tar xvzf filename.tgz
```
3. Start VMware Player or VMware Workstation.
4. Select **Open an Existing Virtual machine**.
5. Navigate to the folder in which the archive was extracted, select the virtual appliance, and click **Open**.
6. Press **Ctrl+G** to control the virtual machine. To return control to the host computer, press **Ctrl+Alt**.
7. Log in as user **root** with password **root**.
8. Accept the software license agreements.
9. After the appliance is initialized, enter **developer** in the Username field and **developer** when prompted for the password.

Installing on Mac OS X Systems

You must first install VMware Fusion on your host computer. Follow the manufacturer's instructions for downloading and installing the software.

1. Download and save the appliance archive file in a directory on your computer.
2. Enter the following command from a shell prompt to uncompress the file:

```
tar xvzf filename.tgz
```
3. Start VMware Fusion.
4. Select **Open an Existing Virtual machine**.
5. Navigate to the folder in which the archive was extracted, select the virtual appliance, and click **Open**.
6. Click inside the VMware Fusion window to control the virtual machine.
7. Log in as user **root** with the password **root**.
8. Accept the software license agreements.
9. After the appliance is initialized, enter **developer** in the Username field and **developer** when prompted for the password.

Installing the IDS Virtual Appliance in a VMware Infrastructure Environment

The IDS Virtual Appliance is supplied in a format suitable for running in a VMware Infrastructure environment. If your organization is configured with VMware Infrastructure using host machines running VMware ESX, you can import the IDS Virtual Appliance image and run one or more virtual machine instances within your data center.

The following topics are discussed in detail on this page:

- "Downloading the ESX image" on page 6
- "Installing the VMware Infrastructure Client" on page 6
- "Importing the IDS Virtual Appliance image" on page 6

- “Administering the Virtual Machine”
- “Cloning the IDS Virtual Appliance image” on page 7

Downloading the ESX image

The IDS Virtual Appliance is supplied in 32-bit and 64-bit versions and is available by downloading or from installation media. Be sure to use the correct version for your application.

To download and extract the IDS Virtual Appliance:

1. Download the correct version of the IDS Virtual Appliance to your system.
2. Navigate to the directory in which the image was downloaded.
3. Extract the image by running the executable file.

The default location for the Virtual Appliance is a subdirectory immediately below the download directory. You can optionally modify the installation directory.

Installing the VMware Infrastructure Client

You use the VMware Infrastructure Client to access the VMware Infrastructure remotely. See your system administrator for the location of the installation media or the file to download and for instructions for installing the application.

Importing the IDS Virtual Appliance image

You import the IDS Virtual Appliance to copy it to a host machine in the data center. After importing the IDS Virtual Appliance, you can access it remotely using the VMware Infrastructure Client.

To clone the IDS Virtual Appliance:

1. Start the VMware Infrastructure Client.
2. Select **File > Virtual Appliance > Import**.
3. Click the **Import from file** radio button, then click **Browse**.
4. Navigate to the directory in which the IDS Virtual Appliance was extracted.
5. Select the file with the .ovf extension and click **Open**.
6. Click **Next**.
7. Verify the Virtual Appliance details and then click **Next**.
8. Type the name by which the IDS Virtual Appliance will be known, select a location, and click **Next**.
9. Select a host or cluster on which to store the Virtual Appliance, then click **Next**.
10. Select the datastore in which to store the Virtual Appliance files, then click **Next**.
11. Verify that the information you selected is correct. If it is, then click **Next**; otherwise, click the **Back** and correct the inaccurate information.

Administering the Virtual Machine

You use the VMware Infrastructure Client to power on or off the Virtual Machine, suspend the Virtual Machine, or edit the virtual machine settings.

To administer the IDS Virtual Appliance:

1. Start the VMware Infrastructure Client application, if necessary.
2. Select **View > Inventory > Virtual Machines and Templates**.
3. Navigate to the IDS Virtual Appliance.
4. Click the **Getting Started** tab, if necessary.
5. Select one of the following:
 - Power on the virtual machine
 - Power off the virtual machine
 - Suspend the virtual machine
 - Edit virtual machine settings

See the VMware documentation for information about editing the virtual machine settings.

Cloning the IDS Virtual Appliance image

You clone the IDS Virtual Appliance to make an identical copy of the image on another virtual machine. You can have several clones of the same virtual image depending on the resources in your data center.

To clone the IDS Virtual Appliance:

1. Start the VMware Infrastructure Client.
2. Select **Inventory > Virtual Machine > Clone**.
3. Type the name by which the new virtual machine will be known, select a location, and click **Next**.
4. Select a host or cluster on which to store the virtual machine, then click **Next**.
5. Select the datastore in which to store the virtual machine files, then click **Next**.
6. Choose the **Do not customize** radio button to continue cloning, or click the **Customize using the Customization Wizard** radio button.
See the VMware documentation for information about using the Customization Wizard.
7. Verify that the information you selected is correct. If it is, then click **Next**; otherwise, click the **Back** and correct the inaccurate information.

See “Administering the Virtual Machine” on page 6 for information about powering on or off the virtual machine.

First Steps

In order to get started quickly with IDS 11.50 software development, a special folder (`/opt/IBM/informix/FirstSteps`) is provided that contains topic-specific subfolders. The subfolders contain topic-specific labs and demonstrations. A symbolic link to the FirstSteps folder is included on the GNOME desktop. Information about the labs and demonstrations can also be found at the following links:

- How to use the Informix JDBC Driver and the JDBC Common Client
- Using PHP with IDS

Within each subfolder is a README file that provides a brief introduction of the objective of the demonstration or lab. As part of your development activities with IDS 11.50, you probably will need to execute SQL language scripts either interactively or as command scripts. The standard tool that comes with each IDS

installation is called **dbaccess**. You start the dbaccess utility either from a command line in a terminal window, or by double-clicking the dbaccess icon on the GNOME desktop. IBM Data Studio and the OpenAdmin Tool for IDS also provide interactive SQL features.

You can practice using the SQL language in IDS 11.50 by playing the IDS Detective Game. See The IDS Detective Game for more information.

The IDS 11.50 Single-Instance Demonstration

The **demo_on** IDS instance is preconfigured with databases named **stores** and **idsgame**. The **idsgame** database is used by the IDS Detective Game. The virtual appliance is configured to automatically start the **demo_on** instance each time the virtual appliance is powered on and to automatically stop the **demo_on** instance each time the virtual appliance is powered off. Use the following commands if you want to recreate the **demo_on** instance or if you want to manually set environment variables that affect the behavior of the **demo_on** instance. See the following links for a complete list of scripts and their descriptions.

- Scripts for use with the IDS demonstration environment
- Scripts for use with the IDS cluster demonstration environment (IDS Enterprise/Developer Editions only)
- Scripts for use with the ESQ/C demonstration environment

The **setDemo** script is used to set the environment variables for the **demo_on** instance. The convenience scripts that create, start, stop, and remove the **demo_on** instance use the **setDemo** script, so you do not need to explicitly source this script before running the convenience scripts. However, to run the dbaccess utility from a command prompt or to perform other actions in the context of the **demo_on** environment, you can explicitly source the **setDemo** script to set up the **demo_on** environment.

In addition to the basic configuration, the **demo_on** instance also provides these helpful enhancements for your development activities:

- A default temporary dbspace, called **tempdbs** (10 MB)
- A default and system smart blob space, called **sbospace** (50 MB)
- One external space to support the Basic Text Search (BTS) extension, called **bts_extspace** (located in \$INFORMIXDIR/demo/server/bts_extspace)
- One **bts** virtual processor (VP) to use the BTS extension
- One **idsxmlvp** virtual processor (VP) to enable additional built-in XML capabilities in IDS 11.50
- One **jvp** virtual processor to enable Java user-defined routines in IDS

Configuring the IDS Appliance

Use the information on this page to configure the IDS virtual appliance to work with your host operating system.

- Keyboard Layout and Settings
- Network Configuration
- Exchanging Data Between the Host and the Virtual Appliance
- Setting the subnet mask on Mac OS X systems
- Changing the VMware Memory Configuration
- Changing the Screen Resolution

- Checking File System Consistency
- Enabling support for Java in IDS
- Starting and Stopping the Apache Server

Keyboard Layout and Settings

GNOME desktop (X11)

To adjust the keyboard layout in the GNOME desktop:

1. Click the **Computer** menu (on the lower left corner of the desktop), and then select **YaST**.
2. At the prompt, enter the user root password.
3. In the YaST Control Center **System** Group section, choose the **Keyboard Layout** applet.
4. From the **Keyboard Properties** window, select your keyboard layout and type from the menus.

Network Configuration

The IDS virtual appliance is configured with two network connections.

- eth0** Use this connection to access the IDS virtual appliance from your host operating-system by specifying the hostname `ids1150srvr` and the IP address `192.168.179.100`.
- eth1** Use this connection to access the Internet from the SUSE server. This connection uses the NAT protocol and is useful if you need to download components to update your image.

Configuring a different static IP address: If the IDS virtual appliance is not accessible using the static IP address `192.168.179.100`, it could mean that your VMware Workstation or VMware Player has been set up to use a different subnet for host-only networking.

To adjust the VMware Workstation subnet settings on Windows systems:

1. Select **Edit > Virtual Network Settings > Host Virtual Network Mapping**.
2. Find VMware Network Adapter VMnet1, and click the > button on the right side.

If you cannot modify the VMnet1 settings, you could instead add a new VMware Host Virtual Adapter. To add a new adapter in VMware Workstation:

- Select **Edit > Virtual Network Settings > Host Virtual Adapters > Add...**
 - Click the **Host Virtual Network Mapping** tab.
3. Select **Subnet**, and change the subnet setting to **192.168.179.0** (mask: **255.255.255.0**).
 4. Confirm the changes.
 5. Restart your VMware image.

If you are using VMware Player instead of VMware Workstation, run the **vmnetcfg.exe** file located in the VMware Player installation directory to apply the changes shown above.

Setting the Subnet Mask on Mac OS X Systems

Configure your Mac OS X system to support network communication between your host system and the IDS virtual appliance by following the steps below:

1. From a terminal window on your host system, change directories to the following location:
`cd /Library/Application Support/VMware Fusion`
2. Stop the network with the following command:
`sudo ./boot.sh --stop`
3. Modify the following three files to replace the existing subnet value with 192.168.179.100:
`config`
`locations`
`vmnet/dhcpd.conf`
4. Restart the network:
`sudo ./boot.sh --start`

After reconfiguring the subnet, verify that you can ping 192.168.179.100 from the host.

Exchanging Data Between the Host and the Virtual Appliance

The IDS virtual appliance is configured to allow for data exchange between the virtual appliance and the host operating system using a variety of methods.

ssh and scp support

You can communicate between the Virtual Appliance and the host machine using **ssh** and **scp**. See the Virtual Appliance man pages for additional information.

FTP support

You can exchange data between the IDS virtual appliance and the host operating system. You can ftp into the appliance after you first configure the ftp service.

Enabling the ftp service

1. On the appliance desktop, select **Computer**.
2. Click **YaST**.
3. When prompted, enter the Administrator (root) password.
4. Click **System Services (Run Level)**.
5. Click **vsftpd**.
6. Click **Enable**.
7. Click **Finish**.

The default ftp configuration is to allow anonymous logins only. When you use ftp to access the virtual appliance, log in with user ID **anonymous** and press Enter when prompted for a password. You can disable the password prompt and you can enable ftp for normal users by modifying the vsftpd.conf file. See the ftp man pages for additional information.

The appliance supports two Ethernet cards. You can either use the fixed IP address **192.168.179.100** or the dynamic IP address that has been assigned by the DHCP client.

To determine the dynamic IP address, run the command **ifconfig** from a terminal window and find the **inet addr** value in the eth1 section of the output.

Note: Depending on the execution environment and the network configuration of the host operating system, the IDS Virtual Appliance may be visible using one IP address or the other but not necessarily both and may or may not be visible using the hostname.

Shared Folder Access

VMware supports a shared folder that can be accessed both from the IDS virtual appliance and from the host operating system. The IDS virtual appliance is configured with two shared folders:

Table 3. Shared Folders

Shared Folder Name	Use
c:\Temp	For use with Windows systems
/tmp	For use with UNIX/Linux systems

If the host operating system folder does not exist on your host machine, you must create it. From within the appliance, shared directories are accessed from the **/mnt/hgfs** directory. For example, if you create a directory named **C:\Temp** on your Windows host system, you can access files from the appliance using **/mnt/hgfs/Temp**. For UNIX and Linux host systems, you can access files from the appliance using the **/mnt/hgfs/tmp** directory.

In order to use the shared folder functionality, you may have to enable the feature. For VMware Workstation: Select **VM > Settings > Options > Shared Folders** and change the default option from **Disabled** to **Always enabled**. For VMware Player, select **VMware Player > Shared Folders**, and change the default option from **Disabled** to **Always enabled**.

Changing the VMware Memory Configuration

The resources available on the host machine determine the performance of the virtual appliance. Adjust the amount of memory used by the appliance if you notice poor or slow performance.

To adjust memory settings when using VMware Player:

1. Click **VMware Player** at the top of the appliance screen, then select **Troubleshoot > Change Memory Allocation**.
2. Adjust the memory settings as required. The display shows the minimum, recommended, and maximum amounts of memory to allocate.
3. Click **OK** when finished. The change will take effect after the virtual appliance is restarted.

To adjust memory settings when using VMware Workstation:

1. Select **VM > Settings** to open the Virtual Machine Settings window.
2. On the Hardware page, select **Memory** and move the slider to adjust the setting.
3. Click **OK**. The change will take effect after the virtual appliance is restarted.

Changing the Screen Resolution

The default screen resolution for the appliance is 1024 x 768. You can change this resolution to suit your host computer display settings.

To change screen resolution:

1. Click **Computer** in the bottom left corner of the IDS virtual appliance screen.
2. Click **Control Center** below the **System** heading.
3. In the Control Center, click **Screen Resolution**.
4. Change the screen resolution as desired from the Resolution list and then click **Apply**.
5. Click **Keep Resolution** to make the changes that you specified.

Checking File System Consistency

You can enable the fsck utility so that it checks the file system for errors and inconsistencies when the virtual appliance boots. Initially, the appliance is pre-configured with fsck disabled to reduce the startup time.

You enable fsck by editing the `/etc/fstab` file. The number in the last (sixth) field specifies whether **fsck** is enabled and if so, the order in which the file systems are checked. To enable fsck, first create a backup of the `/etc/fstab` file, then edit the file and modify the last field of either the root or data file system line. For example, setting the last field of the root file system to 1 and the data file system to 2 causes the root file system to be checked first followed by the data file system.

The file system consistency check will run automatically the next time the virtual appliance is shut down and restarted.

Enabling Support for Java in IDS

While the virtual appliance and the IDS demonstration instances are already configured to support Java, you must set CLASSPATH to include the krakatoa java libraries before you can run the krakatoa examples. These instructions assume that you are at a shell prompt and that your current working directory is `$INFORMIXDIR/extend/krakatoa/examples`.

The **demo_on** IDS instance must be enabled. The virtual appliance is configured to automatically start the **demo_on** instance each time the virtual appliance is powered on. Use the **createDemo**, **startDemo**, **stopDemo**, and **removeDemo** scripts to create, start, stop, or remove the demo_on environment. The scripts must be run as user **informix**.

1. Edit all of the files that have an sql extension (such as: Circle.sql, Env.sql, JDBC.sql, LO.sql) by replacing `$INFORMIXDIR` with `/opt/IBM/informix` in the file contents.
2. For 32-bit systems, modify the Makefile by removing **-fwritable_strings** from the definition of CFLAGS. This step is not necessary for 64-bit systems
3. Set the CLASSPATH environment variable:

```
export CLASSPATH="${INFORMIXDIR}/extend/krakatoa/krakatoa.jar:
${INFORMIXDIR}/extend/krakatoa/jdbc.jar:${CLASSPATH}"
```
4. Build the example:

```
make
```

- If you are logged in as user `informix`, run the example. If you are logged in as developer, then you must first modify your server to configure the `EXTEND` role; otherwise the SQL statements that create the UDR's will fail.

```
./run.sh
```

Starting and Stopping the Apache Server

The IDS virtual appliance is preconfigured with Apache web server, which starts automatically when the virtual appliance is started. Follow the steps below to start or stop the server:

- Log in as root.
- Run either of the following commands:

```
/opt/IBM/OpenAdmin/StartApache  
/opt/IBM/OpenAdmin/StopApache
```

MACH11 Cluster (IDS Enterprise/Developer Edition only)

The IDS virtual appliance editions that support Enterprise Replication (ER) and High Availability clusters provide a demonstration that shows how to configure and use ER and High Availability cluster capabilities. You can easily create a demonstration cluster environment that is separate from the demonstration instance. To find out more about MACH11 and configuring MACH11 clusters, see IDS 11.5 Info Center

The **cluster demonstration** is a set of pre-configured instances that demonstrate MACH11 capabilities. All demonstration instances are independent of one another, so you do not have to stop or remove any of the other pre-configured instances prior to creating or starting the cluster demonstration instances. You must explicitly create and start the **cluster demonstration** instances.

Convenience scripts are provided for you to create, start, stop, and remove the **cluster demonstration** instances. All convenience scripts must be executed as user `informix`.

To use the cluster demonstration, you must run the **createCluster** script as user `informix`. The **createCluster** script creates and starts the cluster demonstration instances. You can stop the cluster demonstration instances using **stopCluster** and start them again with **startCluster**.

The cluster environment does *not* start automatically after a re-boot.

To create, remove, start, or stop the demonstration cluster as user `informix`, use one of the following commands:

Table 4. MACH11 Cluster Start and Stop Scripts

<code>createCluster</code>	Creates and starts an IDS 11.50 demonstration cluster.
<code>removeCluster</code>	Stops the demonstration cluster and removes all database files associated with the cluster instances.
<code>startCluster</code>	Starts a previously created demonstration cluster.
<code>stopCluster</code>	Stops a running IDS demonstration cluster.

The MACH11 Cluster demonstration creates multiple **cluster demonstration** instances. Use the scripts described in the previous table to manage all instances. Separate scripts that set the environment for each individual cluster demonstration instance are included. These scripts are useful if you want to perform actions in the context of one of the cluster demonstration instances (to look at the instance's server log, for example, or to run the **dbaccess** utility against the instance, and so on.) When using the scripts in the following table, source the appropriate environment setup script from a command prompt.

Table 5. MACH11 Cluster Manipulation Scripts

. setPrimary	Sets the environment variables for the primary instance (cheetah2).
. setSDS1	Sets the environment for the sds1 instance.
. setSDS2	Sets the environment for the sds2 instance.
. setHDR	Sets the environment for the hdrsrv instance.
. setRSS1	Sets the environment for the rss1 instance.
. setRSS2	Sets the environment for the rss2 instance.

Connection Manager Log File

The tailCM script runs the **tail** command to continuously display the last few lines of the Connection Manager log file. When using the script, you may notice the following warning in the log:

```
Warning: Password Manager failed; working in trusted node mode
```

The warning indicates that no Connection Manager password file exists and that the Connection Manager can access all of the servers in the MACH11 cluster without using a password. You can prevent this warning from appearing by creating a password file and then encrypting it using the **onpassword** utility. See *The onpassword Utility* chapter of the *IBM Informix Dynamic Server Administrator's Reference* located here: <http://publib.boulder.ibm.com/infocenter/idshelp/v115/index.jsp>

Web-based Demonstrations for the IDS 11.50 Demonstration Cluster

The IDS virtual appliance includes some pre-configured demonstrations that illustrate some of the high-availability features provided by IDS 11.50. These demonstrations are written as PHP-based Web pages and they require the **cluster demonstration** instances to be started.

OpenAdmin Tool (OAT) version 2.24 for IDS

Starting OAT for the First Time

1. Click the link below to access the OpenAdmin Tool (OAT):
<http://localhost/openadmin>
2. In the **Quick Login** section, click **Get Servers**, choose the **demo_on** instance in the **Select a Connection** box, and click **Login**.

The following server details are filled in automatically:

Table 6. OAT Login Information

Informix Server	demo_on
hostname	ids1150srvr
Port	9088
User	informix
Password	informix

Installing the Missing Plugin

The first time you use OAT and navigate to a page that requires the Adobe® Flash Player plugin, a note is displayed at the top of the page indicating that additional plugins are required to display all the media on the page. Internet connectivity is required to download the plugin. Follow these steps to install the missing plugin:

1. Click **Install Missing Plugins**.
2. Click **Next** when the Plugin Finder Service window is displayed.
3. Click **I agree** to accept the license agreement.
4. In the Firefox window, select **.rpm** from the dropdown list and then click **Agree and install now**.
5. When asked whether to Open or Save the file, select **Open with** and click **OK**. If you receive a message indicating that no plugins were installed, then click **Manual Install**.
6. You will be prompted to enter the root password in order to install the plugin. Enter **root** for the password and then click **Continue**.
7. Click the **Install** button when the Software Installer window is displayed.
8. Click the **Close** button when the Software Installer indicates a successful installation.
9. In the Firefox OpenAdmin tool session, click **Reload** one or two times to refresh the display.

The single demonstration instance **demo_on** belongs to the **Default** connection group.

OAT has been pre-configured as follows:

- There is no OAT Group password
- The **demo_on** instance has been added to the **Default** group
- For IDS Enterprise Edition and IDS Developer Edition only:
An OAT Group called **IDS11Cluster** has been created. All cluster demonstration instances have been added to this OAT Group. The cluster demonstration instances are:
 - **cheetah2**
 - **hdrsrv**
 - **sds1**
 - **sds2**
 - **rss1**
 - **rss2**

The PHP files for OpenAdmin are located in various directories under: `/opt/IBM/OpenAdmin` in case you want to modify them.

Scripts

This page describes the scripts included with the virtual appliance. The following topics are included:

- Scripts for use with the IDS cluster demonstration environment (IDS Enterprise/Developer Editions only)
- Scripts for use with the IDS demonstration environment
- Scripts for use with the ESQL/C demonstration environment

Scripts for Use with the IDS Cluster Demonstration Environment (IDS Enterprise/Developer Editions only)

Use these scripts to create, delete, and configure the MACH11 cluster environment. These scripts are located in `$INFORMIXDIR/bin` and must be run as user **informix**.

Table 7. Cluster Demonstration Scripts

Script Name	Description
<code>createCluster</code>	Creates and starts the IDS cluster. The script creates a primary server, an HDR secondary server, two SD (shared disk) secondary servers, and two RS (remote standalone) secondary servers. The script also configures and starts the Connection Manager.
<code>removeCluster</code>	Stops the demonstration cluster and removes all database files associated with the cluster instances.
<code>setHDR</code>	Sets the environment for the <code>hdrsrv</code> instance.
<code>setPrimary</code>	Sets the environment for the primary instance (<code>cheetah2</code>).
<code>setRSS1</code>	Sets the environment for the <code>rss1</code> instance.
<code>setRSS2</code>	Sets the environment for the <code>rss2</code> instance.
<code>setSDS1</code>	Sets the environment for the <code>sds1</code> instance.
<code>setSDS2</code>	Sets the environment for the <code>sds2</code> instance.
<code>startCluster</code>	Starts a previously created demonstration cluster.
<code>stopCluster</code>	Stops a running IDS demonstration cluster.
<code>tailCM</code>	Runs the <code>tail</code> command to continuously display the last few lines of the Connection Manager log file.

Scripts for Use with the IDS Demonstration Environment

Use these scripts to create, remove, and configure the demonstration database environment. These scripts are located in `$INFORMIXDIR/bin` and must be run as user **informix**.

Table 8. IDS Demonstration Scripts

Script Name	Description
createDemo	Creates and starts the demonstration instance with the default settings.
removeDemo	Deletes all of the database files that are associated with the demonstration instance.
setDemo	Sets the correct environment variables for the demonstration instance.
startDemo	Starts the demonstration instance.
stopDemo	Stops the demonstration instance.

Scripts for Use with the ESQL/C Demonstration Environment

These scripts are used to start, stop, and configure the ESQL/C demonstration database environment. Use these scripts to run any of the ESQL/C demonstration programs that are included with the IDS server. The demonstration programs are located in \$INFORMIXDIR/demo/esqlc.

Table 9. ESQL/C Demonstration Scripts

Script Name	Description
createEsqclDemo	Creates and starts the ESQL/C demonstration environment.
removeEsqclDemo	Deletes all files that are associated with the ESQL/C demonstration environment.
setEsqclDemo	Sets the environment variables for the ESQL/C demonstration environment.
startEsqclDemo	Starts the ESQL/C demonstration environment.
stopEsqclDemo	Stops the ESQL/C demonstration environment.

Using the Informix JDBC Driver and the JDBC Common Client

The objective of this tutorial is to show you how to connect to an IDS instance using the JDBC driver.

This example assumes that you are logged into the developer account. Modify the instructions appropriately if you are logged into the informix account.

The sample code assumes that the **demo_on** instance is up and running and that the stores database is available. To make sure the server is online, type the following in the terminal window:

```
onstat -
```

If the server is not on line, start it by entering the following command as user informix

```
startDemo
```

In the steps below, you will compile and run a java program named **gentest.java**

1. From a terminal window, change to the tutorial directory:

```
cd ~/Desktop/IDS11_FirstSteps/02_Java
```

2. Compile the Java source:

```
javac gentest.java
```

3. Now, run the program using the JDBC Common Client:

```
java gentest 0
```

The output of the program should look like the following:

```
Try connecting to stores
Connected ...
Driver name: IBM DB2 JDBC Universal Driver Architecture
Driver version: 3.52.95
Database product name: IDS/UNIX32
Database product version: IFX11500
Prepare and Execute: SELECT * FROM INFORMIX.CUSTOMER
WHERE CUSTOMER_NUM = 101
Getting data for row: 1
customer_num: 101
fname: Ludwig
lname: Pauli
company: All Sports Supplies
address1: 213 Erstwild Court
address2: null
city: Sunnyvale
state: CA
zipcode: 94086
phone: 408-789-8075
Done
```

- 4.

To run the program using the Informix JDBC Client:

```
java gentest 1
```

```
Try connecting to stores
Connected ...
Driver name: IBM Informix JDBC Driver for IBM Informix
Dynamic Server
Driver version: 3.50.JC2DE
Database product name: Informix Dynamic Server
Database product version: 11.50.UC2DE
Prepare and Execute: SELECT * FROM INFORMIX.CUSTOMER
WHERE CUSTOMER_NUM = 101
Getting data for row: 1
customer_num: 101
fname: Ludwig
lname: Pauli
company: All Sports Supplies
address1: 213 Erstwild Court
address2: null
city: Sunnyvale
state: CA
zipcode: 94086
phone: 408-789-8075
Done
```

Using PHP with IDS

PHP is a scripting language that can be embedded into HTML documents. When a browser detects the PHP script tag, it hands the script to the PHP processor. HTML is dynamically generated by the PHP script to provide dynamic Web page content.

Objective

The objective of this tutorial is to show you how to access and interact with IDS using PHP.

Prerequisites

The sample code assumes that the **demo_on** instance is up and running and that the stores database is available. To make sure the server is online, enter the following command in a terminal window:

```
onstat -
```

If the server is not online, start it by typing the following command as user **informix**:

```
startDemo
```

The IDS virtual appliance is pre-configured with an Apache Web server. The document root for this Web server is `/opt/IBM/OpenAdmin/Apache_2.2.4/htdocs`.

This PHP demonstration is called `HelloWorld.php` and has already been installed in the Apache Web server document root at `/opt/IBM/OpenAdmin/Apache_2.2.4/htdocs`. To install a PHP program, simply copy it to the Apache document root. Then access it through the Firefox browser.

Tasks

The `HelloWorld.php` PHP demonstration accesses the stores database in the **demo_on** instance.

To run the `HelloWorld.php` demonstration program, click the link below:

<http://localhost/HelloWorld.php>

Open `HelloWorld.php` using a text editor.

`HelloWorld.php` is an HTML document that embeds a PHP script that interacts with the IDS server. Everything between the `<?php` and `?>` tags is PHP code.

This example uses the PHP PDO class to represent the connection between PHP and the IDS server. The database connection is opened when the PDO class is instantiated.

```
<html>
<body>
<?php
try {
$db = new PDO("informix:host=ids1150srvr; service=9088;
database=stores; server=demo_on; protocol=onsoctcp;
EnableScrollableCursors=1;", "informix", "informix");
print "Hello World!</br></br>";
print "Connection Established!</br></br>";
$stmt = $db->query("select * from customer");
$res = $stmt->fetch( PDO::FETCH_BOTH );
$rows = $res[0];
echo "Table contents: $rows.</br>";
} catch (PDOException $e) {
    print $e->getMessage();
}
}
```

```
?>  
</body>  
</html>
```

The statement that begins with **\$db = new PDO** instantiates the PHP PDO module. This argument establishes a connection to the IDS database, as specified by the arguments to the PHP PDO constructor. You can confirm this information using the dbaccess utility and the information in your SQLHOSTS file.

Make sure that your environment is set correctly before running dbaccess from the command prompt. You can do this with any one of the set* convenience scripts. The dbaccess utility's **Database** menu lists the databases that are available in the IDS instance. Once your environment is set up, you can see the contents of your SQLHOSTS file by typing

```
cat $INFORMIXSQLHOSTS
```

The HelloWorld.php example instantiates the PHP PDO module with the argument **EnableScrollableCursors=1**. This argument is required because the PHP PDO module does not enable scrollable cursors by default.

The last two arguments to the PHP PDO constructor specify the username and password to use when establishing a connection to the IDS server. In this example, the username is **informix** and the password is **informix**.

Feel free to modify the program. You can simply display data or you can create code that interacts with the user and generates dynamic Web page content based on the user's response.

After adding or modifying the code, test it by copying the file to **/opt/IBM/OpenAdmin/Apache_2.2.4/htdocs**. Run the program by opening Firefox and navigating to your new Web content.

For example, if the new script is called myTest.php, then navigate to:

```
http://localhosts/myTest.php
```

The IDS Detective Game

The information in this section provides an overview of what a database is, a brief history of databases, and instructions on how to start and play the IDS detective game. In the game, you will be given a problem and a series of clues about how to solve the problem using SQL queries. The queries are run against a preinstalled database and get progressively more complex as the game proceeds. Answers are provided at the end of this section.

Play the IDS Detective Game to help you get used to the Informix SQL language and to familiarize yourself with the dbaccess SQL tool. And oh yes, to have fun!

What is a Database?

A database is a collection of data (values) stored in tables, where each table consists of columns (fields) and rows (records)

A database can be compared to a filing cabinet:

- Database = filing cabinet
- Table = file folder within the cabinet
- Value = data stored in each cell of a spreadsheet of rows (records) and columns (fields) within the file folder

Both the filing cabinet and database provide you with a way to manage and organize data so that it can be quickly retrieved. However, because the database is automated it shortens the time required to search for data, sort data, add to the data, delete from the data, and edit the data.

With a relational database, such as IBM Informix Dynamic Server, data is organized and accessed according to relationships defined between data items within the various tables. Data in the tables can be managed using the SQL (Structured Query Language) programming language.

The Apple iPod is an example of a database. The Apple iPod uses a database to store all of your Music, Photos, etc.:

- Database: Entire Music Folder
- Tables: Artist Folders (grouped by artist)
- Values: Album Names, Song Names, etc.

The History of Databases

- 1970 - IBM invents the database E. F. Codd of IBM Research publishes a paper entitled "A Relational Model of Data for Large Shared Data Banks," leading to a new way for computers to manage information.
 - 1974 - IBM invents the database programming language SEQUEL (or SQL for short).
 - Don Chamberlin and Ray Boyce publish "SEQUEL: A Structured English Query Language."
 - 1970 to the present - IBM offers a complete family of relational database management systems (RDBMS) software
- Databases are used across all industries to manage everything from your credit card use, to bank accounts, to car insurance, to store purchases.

Playing the IDS Detective Game

Today you are going to learn about database technology by playing a fun and interactive game called "The IDS Detective Game." This game will enable you to:

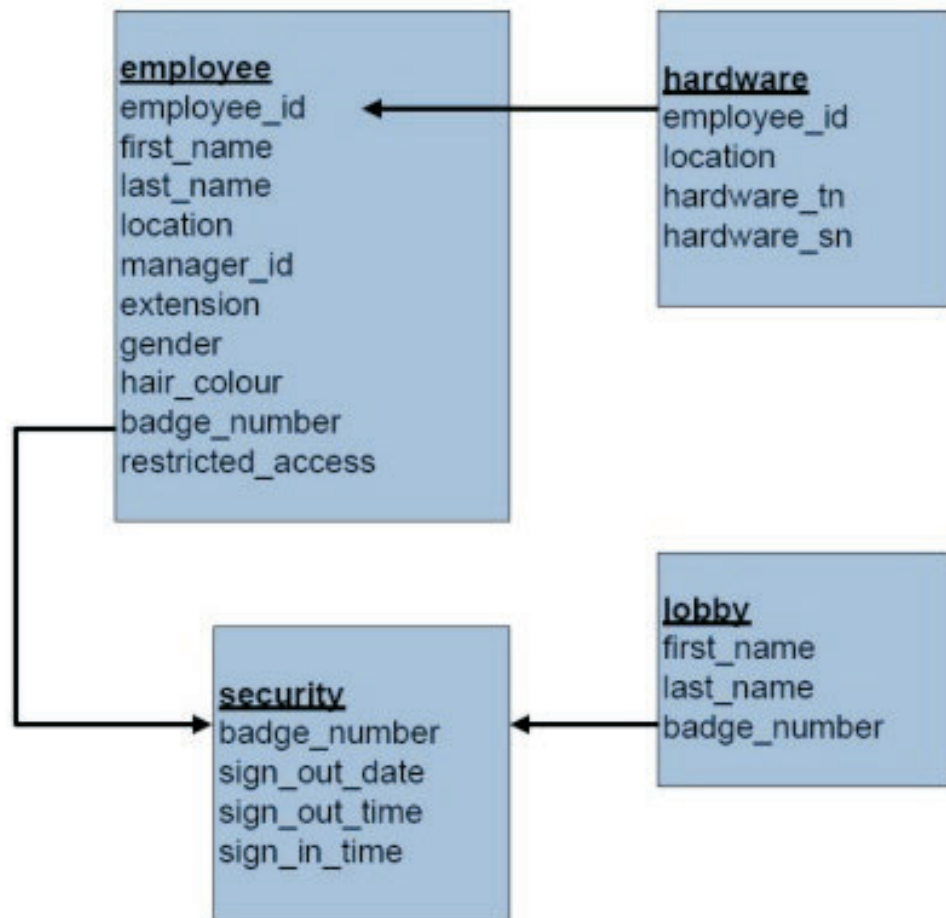
- Understand how database data is stored and to gain an understanding of database concepts such as tables, rows (records), columns (fields), and values
- Learn about relational databases and table joins
- Gain a working knowledge of some of the primary SQL statements:
 - SELECT
 - UPDATE
 - INSERT
 - DELETE

The IDS Detective Game consists of four tables:

- **employee** table stores information about each employee such as their employee number, name, manager, office number, phone number, and badge number, and so on.

- **security** table stores information as to when each employee or visitor badged in and out of the building (date and time).
- **lobby** table stores the badge number assigned to each visitor by name.
- **hardware** table stores the serial number and type number for every computer along with its location and the ID of the employee who is its primary user.

Relationships between tables in a relational database are established using **keys**. The relationships defined between the tables of the IDS Detective Game are shown in the diagram. Arrows indicate the fields that have keys associated with them and the relationships that were defined between the tables. You will need to use these keys when joining the tables.



Instructions

Follow the steps below to play the IDS detective game:

1. Double-click the **IDS11_FirstSteps** icon on the IDS virtual machine desktop, then double-click the **01_IDS_Detective_Game** folder.
2. Start IDS Detective clue program by double-clicking the **IDS11 Detective Game** icon.
3. Start the DBAccess utility by double-clicking the **dbaccess SQL Editor** icon.

Try performing a simple query to get used to the **dbaccess** program:

1. Select **Query-language** from the DBAccess main menu.
2. Select **New** from the Query-language menu and press Enter.
3. Try a test command by typing the following: `SELECT * FROM lobby`
4. Press the Esc (Escape) key then press Enter to run the query. You are now ready to play the game!

Tips

- Use DBAccess to execute SQL statements.
- Enter SQL commands in the **Query-language** tab from the DBAccess main menu. From the Query-language menu:
 - **New** starts a new SQL statement,
 - **Run** executes the SQL statement,
 - **Modify** modifies your last SQL statement
- When viewing your results, press **Next** to scroll through the entire list of results.
- To see the count of the number of records for an SQL statement, select **Next** until the number of rows retrieved is displayed.
- If the width of the fields to be displayed is less than 80 characters wide (including the field name), the results are displayed in a table format.
- If the width of the fields to be displayed is greater than 80 characters wide (including the field name), the results for each record is displayed over multiple lines, as defined by the table structure.
- Note the format of data in the tables
 - gender field contains a single uppercase letter (M or F)
 - hair_colour field contains lowercase entries only (brown, black, blonde, or red)
 - restricted_access field contains a single uppercase letter (Y or N)
 - sign_out_time and sign_in_time fields use a 24-hour format (e.g., 14:00:00)
 - sign_out_date field uses year-month-day format (e.g., 2007-11-07)
- The * in place of the column name(s) of the SQL query is equal to identifying all column names


```
SELECT *
FROM lobby
```

The above query selects all columns from the lobby table.
- **AND** is used to create compound conditions in a **SELECT** statement


```
SELECT *
FROM employee
WHERE employee.manager_id = '278354' AND employee.gender = 'M'
```

The Employee Table

In the employee table, the columns (or fields) are: **employee_id**, **first_name**, **last_name**, **location**, **manager_id**, **extension**, **gender**, **hair_colour**, **badge_number**, and **restricted_access**.

There are 140 rows (records) in the employee table but only a subset of the results table is shown. You need to page down (using **Next** menu option) to see additional records.

The Security Table

In the security table, the columns (or fields) are: **badge_number**, **sign_out_date**, **sign_out_time**, and **sign_in_time**.

There are 151 rows (records) in the security table but only a subset of the results table is shown on each page. You need to page down (using **Next** menu option) to see additional records.

The Lobby Table

In the lobby table, the columns (or fields) are: **first_name**, **last_name**, and **badge_number**.

There are 10 rows (records) in the lobby table.

The hardware table

In the hardware table, the columns (or fields) are: **employee_id**, **location**, **hardware_tn**, and **hardware_sn**.

There are 140 rows (records) in the hardware table but only a subset of the results table is shown on each page. You need to page down (using **Next** menu option) to see additional records.

SQL Statements

The language of relational database technology is the Structured Query Language (SQL). Invented by IBM in the 1970s, the SQL language continues to evolve and is the only way to access relational database data.

This tutorial will introduce you to the following primary SQL statements:

- SELECT - queries data from one or more tables
- UPDATE - changes existing rows in a table
- INSERT - adds data to a table
- DELETE - removes rows from a table

The following sections describe each of these SQL statements so you will have all of the information needed to play the IDS Detective Game.

The SELECT Statement

The SELECT statement is used to retrieve data. The format of this statement is:

```
SELECT column name(s)
FROM table name(s)
WHERE conditions for rows to meet (if any)
```

For example, to select all of the rows from the lobby table, issue the following command:

```
SELECT *
FROM lobby
```

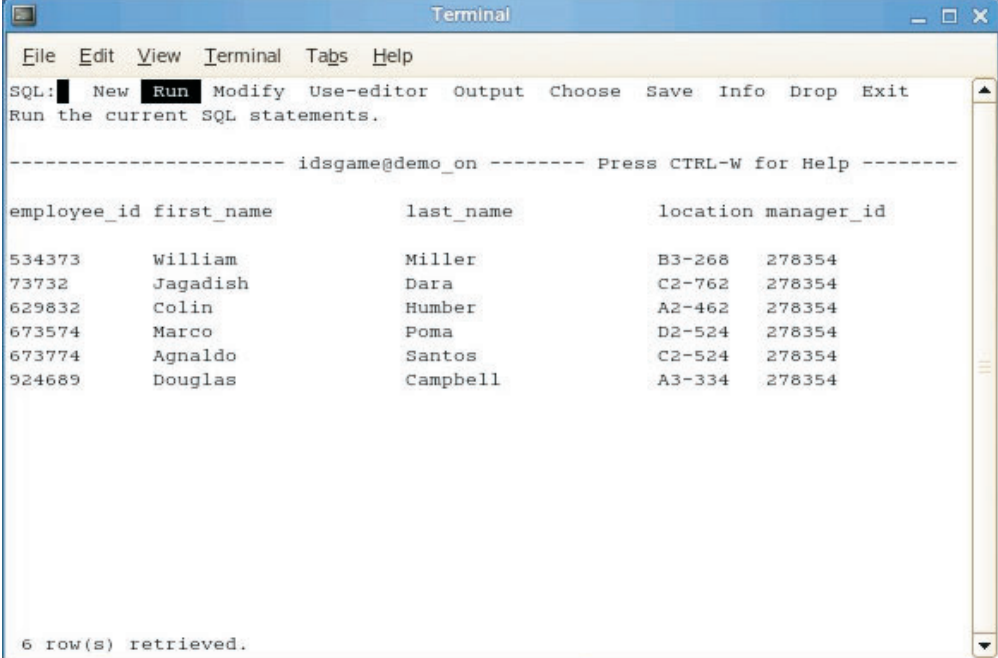
To select all of the rows and columns from the lobby table where the visitor's badge number is greater than 62, issue the following statement:

```
SELECT *
FROM lobby
WHERE lobby.badge_number > 'V0062'
```

To select all of the rows and columns from the employee table where the employee reports to Claudinei Santana (hint: employee id = 278354), and the employee is male:

```
SELECT *
FROM employee
WHERE employee.manager_id = '278354' AND employee.gender = 'M'
```

This statement will produce the following result:



```
Terminal
File Edit View Terminal Tabs Help
SQL: New Run Modify Use-editor Output Choose Save Info Drop Exit
Run the current SQL statements.
----- idsgame@demo_on ----- Press CTRL-W for Help -----
employee_id first_name      last_name      location manager_id
534373      William         Miller         B3-268      278354
73732       Jagadish        Dara           C2-762      278354
629832      Colin           Humber        A2-462      278354
673574      Marco           Poma          D2-524      278354
673774      Aginaldo        Santos         C2-524      278354
924689      Douglas         Campbell       A3-334      278354
6 row(s) retrieved.
```

Note: In order to see the output in a tabular format as shown here, use the following syntax:

```
SELECT employee_id, first_name, last_name, location, manager_id FROM employee
WHERE employee.manager_id = '278354' AND employee.gender = 'M'
```

The SELECT statement will take any number of relational operators including: =, >, <, >=, <=, <> (not equal to) The SELECT statement can also take a number of predicates including:

- **LIKE** and **NOT LIKE**
- **IS NULL** and **IS NOT NULL**
- **BETWEEN** and **NOT BETWEEN**
- **IN** and **NOT IN**

In the IDS Detective Game, you will use the BETWEEN predicate.

The BETWEEN Predicate

The BETWEEN predicate compares a single value to an inclusion range of values (that is, all values *between* a specified maximum value and minimum value).

For example, to select all of the rows and columns from the lobby table where the visitor's badge number is between 59 and 66, issue the following statement:

```
SELECT *
FROM lobby
WHERE lobby.badge_number BETWEEN 'V0059' AND 'V0066'
```

The UPDATE Statement

The UPDATE statement is used to change data in a table. With this statement, you can change the value of one or more columns for each row that satisfies the search condition of the WHERE clause. The format is:

```
UPDATE table name
SET column name = expression
WHERE conditions for rows to meet (if any)
```

Note: if you do not use the WHERE clause, all rows will be updated.

For example, to change the first name of the visitor having badge number 58 to "Joan", issue the following statement:

```
UPDATE lobby
SET first_name = 'Joan'
WHERE lobby.badge_number = 'V0058'
```

If you had omitted the WHERE clause on the previous SQL command and issued the following command instead, all records would have been updated:

```
UPDATE lobby
SET first_name = 'Joan'
```

The INSERT Statement

The INSERT statement is used to add data to a table. The format of this statement is:

```
INSERT INTO tablename (column name(s))
VALUES (value(s));
```

To add a visitor named "IDS Detective" to the lobby table, issue this command:

```
INSERT INTO lobby (first_name, last_name, badge_number)
VALUES ('IDS', 'DETECTIVE', 'V0062')
```

Note: This insertion is the 11th record of the lobby table and the table now has two records with badge_number = 'V0062'

The DELETE Statement

Use the DELETE statement to remove records (rows) from a table. The format is:

```
DELETE FROM table name
WHERE conditions for rows to meet (if any)
```

For example, to remove the record for the visitor with a last name of Detective from the lobby table, issue the following statement:

```
DELETE FROM lobby
WHERE lobby.last_name = 'Detective'
```

Note: Following this deletion there are only 10 records in the lobby table and only one record with badge_number = 'V0062'.

Joining Tables

The process of combining data from two or more tables is called joining tables. The columns involved in the join condition do not have to be identical; however, they must be compatible. To join the lobby table to the security table, issue the following command:

```
SELECT *
FROM lobby, security
WHERE security.badge_number = lobby.badge_number
```

Note: The fields shown are from both the lobby and security tables.

Recall: The Security table has 151 rows but only 10 that satisfy the join condition.

IDS Detective Game - Answers

IDS Detective: Solving the Mystery of the Missing Thinkpad

Following are the clues and solutions that you will need to solve the mystery of the missing Thinkpad.

Clue Number 1

Security has confirmed that only employees (that is, no visitors) left the building with any hardware on the date that the thinkpad went missing.

Solution to Clue Number 1:

Determine the total number of employees by issuing the following SQL command:

```
select * from <userid>.employee
```

If done correctly, the resulting employee query will show 140 employees (records).

Notes:

- To view the data in a table layout, use the following SQL statement:

```
SELECT employee_id, first_name, last_name, location, manager_id
FROM employee
```
- Use the "Next" menu option to view additional records
- The count of the number of records is shown along the bottom of the screen after the last record is displayed
- Use the Modify menu option to re-use the query statement for subsequent queries.

Clue Number 2:

Security has confirmed that the Thinkpad went missing on July 4, 2009. (Hint: Determine the complete list of potential suspects by joining two tables.)

Solution to Clue Number 2:

Part 1:

To determine which employees accessed the building on July 4, 2009, you need to join the Employee table and the Security table. Note that BADGE_NUMBER is the key for the table join and then issue the following SQL command:

```
select * from <userid>.employee as employee, <userid>.security as security
where employee.badge_number = security.badge_number
```

If done correctly, the resulting merged EMPLOYEE and SECURITY tables will show the appropriate values for 140 employees (records).

Note:

- To view the data in a table layout, use the following SQL statement:

```
SELECT employee.employee_id, employee.last_name, security.badge_number,  
security.sign_out_date  
FROM employee, security  
WHERE employee.badge_number = security.badge_number
```

Part 2:

Now you need to determine which employees signed out on July 4, 2009. To do this:

Revise the above query to also select all employees who signed out on July 4, 2009. The SQL command should be as follows:

```
select * from <userid>.employee as employee, <userid>.security as security  
where employee.badge_number = security.badge_number  
and security.sign_out_date = '07-04-2009'
```

If done correctly, this query will result in 44 suspects.

Note:

- To view the data in a table layout, use the following SQL statement:

```
SELECT employee.employee_id, employee.last_name, security.badge_number,  
security.sign_out_date  
FROM employee, security  
WHERE employee.badge_number = security.badge_number  
AND security.sign_out_date = '07-04-2009'
```

Clue Number 3:

The suspect signed out of the building between 4 p.m. and 7 p.m.

Solution to Clue Number 3:

To determine which of the remaining 44 suspects signed out between 4 p.m. and 7 p.m., issue the following SQL command:

```
select * from <userid>.employee as employee, <userid>.security as security  
where employee.badge_number = security.badge_number  
and security.sign_out_date = '07-04-2009'  
and security.sign_out_time between '16:00:00' and '19:00:00'
```

If done correctly, this query will result in 32 suspects.

To view the data in a table layout, use the following SQL statement:

```
SELECT employee.employee_id, employee.last_name, security.badge_number,  
security.sign_out_date  
FROM employee, security  
WHERE employee.badge_number = security.badge_number  
AND security.sign_out_date = '07-04-2009'  
AND security.sign_out_time BETWEEN '16:00:00' AND '19:00:00'
```

Clue Number 4:

Video surveillance tapes show that only males exited the building carrying Thinkpad cases between 4 p.m. and 7 p.m. on July 4, 2009

Solution to Clue Number 4:

To determine which of the remaining 32 suspects is male, issue the following SQL command:

```
select * from <userid>.employee as employee, <userid>.security as security
where employee.badge_number = security.badge_number
and security.sign_out_date = '07-04-2009'
and security.sign_out_time between '16:00:00' and '19:00:00'
and employee.gender = 'M'
```

If done correctly, this query will result in 23 suspects.

To view the data in a table layout, use the following SQL statement:

```
SELECT employee.employee_id, employee.last_name, security.badge_number,
security.sign_out_date
FROM employee, security
WHERE employee.badge_number = security.badge_number
AND security.sign_out_date = '07-04-2009'
AND security.sign_out_time BETWEEN '16:00:00' AND '19:00:00'
AND employee.gender = 'M'
```

Clue Number 5:

The suspect has brown hair.

Solution to Clue Number 5:

To determine which of the remaining 23 suspects has brown hair, issue the following SQL command:

```
select * from <userid>.employee as employee, <userid>.security as security
where employee.badge_number = security.badge_number
and security.sign_out_date = '07-04-2009'
and security.sign_out_time between '16:00:00' and '19:00:00'
and employee.gender = 'M'
and employee.hair_colour = 'brown'
```

If done correctly, this query will result in 11 suspects.

To view the data in a table layout, use the following SQL statement:

```
SELECT employee.employee_id, employee.last_name, security.badge_number,
security.sign_out_date
FROM employee, security
WHERE employee.badge_number = security.badge_number
AND security.sign_out_date = '07-04-2009'
AND security.sign_out_time BETWEEN '16:00:00' AND '19:00:00'
AND employee.gender = 'M'
AND employee.hair_colour = 'brown'
```

Clue Number 6:

The call reporting the missing thinkpad was placed from extension 5555. (Hint: This is not the person who took the missing Thinkpad, but you will need information from the results later in the game.)

Solution to Clue Number 6:

Issue the following SQL command:

```
select * from <userid>.employee as employee, <userid>.security as security
where employee.badge_number = security.badge_number
and security.sign_out_date = '07-04-2009'
and security.sign_out_time between '16:00:00' and '19:00:00'
and employee.gender = 'M'
and employee.hair_colour = 'brown'
and employee.extension = '5555'
```

Hint: This clue is a Red Herring. It will not help you find the sole suspect. However, later in the game you will need some of the values from this record. Note that the EMPLOYEE_ID is 924578.

Remove the following from the SQL command:

```
employee.extension = '5555'
```

The SQL command should now appear as follows:

```
select * from <userid>.employee as employee, <userid>.security as security
where employee.badge_number = security.badge_number
and security.sign_out_date = '07-04-2009'
and security.sign_out_time between '16:00:00' and '19:00:00'
and employee.gender = 'M'
and employee.hair_colour = 'brown'
```

Hint: You must enter 10 as the answer for this clue.

Clue Number 7:

A thinkpad with the same serial number as that of the missing machine was found during an inventory audit in room X-2111 which is a restricted access area.

Solution to Clue Number 7:

To determine which of the remaining suspects has access to restricted access rooms, issue the following SQL command:

```
select * from <userid>.employee as employee, <userid>.security as security
where employee.badge_number = security.badge_number
and security.sign_out_date = '07-04-2009'
and security.sign_out_time between '16:00:00' and '19:00:00'
and employee.gender = 'M'
and employee.hair_colour = 'brown'
and employee.restricted_access = 'Y'
```

If done correctly, this query will result in only one remaining suspect: Marc Ost (EMPLOYEE_ID: 924580).

To view the data in a table layout, use the following SQL statement:

```
SELECT employee.employee_id, employee.last_name, security.badge_number,
security.sign_out_date
FROM employee, security
WHERE employee.badge_number = security.badge_number
AND security.sign_out_date = '07-04-2009'
AND security.sign_out_time BETWEEN '16:00:00' AND '19:00:00'
AND employee.gender = 'M'
AND employee.hair_colour = 'brown'
AND employee.restricted_access = 'Y'
```

Closing out the investigation

Marc Ost's manager was contacted. The manager indicated that he had given Marc the Thinkpad to use but had forgotten to update the hardware table to indicate that Marc was the new owner.

Update the hardware table to reflect Marc's ownership of the missing thinkpad.

Solution

Delete the SQL command currently in the command field

In order to update the serial number for Marc Ost's machine, we need to get the machine information from the EMPLOYEE table for the employee who owns extension 5555:

```
select * from <userid>.employee as employee
where employee.extension = '5555'
```

Note that the EMPLOYEE_ID is 924578.

Now query the HARDWARE table to determine the machine owned by this employee:

```
select * from <userid>.hardware as hardware
where hardware.employee_id = '924578'
```

Note that this employee owns the machine with HARDWARE_TN = 6627-425 and HARDWARE_SN = 74-22265

Now delete this record from the HARDWARE table (because this employee no longer owns any hardware).

```
delete from <userid>.hardware as hardware
where hardware.employee_id = '924578'
```

You can verify that the record was deleted, if you like, by issuing the following SQL command to search for EMPLOYEE_ID 924578 in the HARDWARE table:

```
select * from <userid>.hardware as hardware
where hardware.employee_id = '924578'
```

Update Marc Ost's hardware information to make him the owner of the Thinkpad with type number 6627-425 and serial number 74-22265:

```
update <userid>.hardware as hardware
set hardware.hardware_tn = '6627-425', hardware.hardware_sn = '74-22265'
where hardware.employee_id = '924580'
```

You can verify that the record was updated, if you like, by issuing the following SQL command to search for EMPLOYEE_ID 924580 in the HARDWARE table:

```
select * from <userid>.hardware as hardware
where hardware.employee_id = '924580'
```

Introduction to Data Studio

In this tutorial, you will use IBM Data Studio to learn about the Java and Data perspectives. You will also see how to connect to a database and sample the contents of tables in the database.

Starting Data Studio

To start Data Studio:

1. Double-click the **IBM Data Studio** icon on the desktop. You can also start Data Studio by clicking **Computer > More Applications > IBM Software Development > Platform > IBM Data Studio**.
2. In the Workspace Launcher window, enter one of the following:
 - If you are logged in as user developer, enter:
`/home/developer/pureQueryLabs`
 - If you are logged in as user informix, enter:
`/home/informix/pureQueryLabs`
3. Click **OK** to launch Data Studio. The working directory is automatically created.
4. Click the **Workbench** icon at the top of the screen.

Working with Perspectives

A perspective is a group of views and editors. One or more perspectives can exist in a single Workbench window. Within a window, each perspective may have a different set of views, but all perspectives share the same set of editors.

By default, Data Studio is opened in the Data Perspective. The Java perspective is an initial set of views in which the pureQuery code is saved. A Java perspective combines views you would normally use to edit Java programs. For example, the view called Package Explorer is where the pureQuery projects are stored. For this tutorial, the default Java Common Client driver is used. To set Data Studio to the Java Perspective:

1. Select **Window > Open Perspective > Java**
2. Add the Data Source Explorer View to this perspective by selecting: **Window > Show View > Other**
3. Expand the **Data Management** folder and select **Data Source Explorer**.
4. Click **OK**.

Within a perspective, you can drag and drop the various views and place them where they are most useful. To drag and drop a view, left click the tab of that view, hold down the mouse button, and drag the view to where you want it to be positioned within that perspective.

Connecting to a Database and Exploring Tables

In this section, you will use Data Studio to connect to an IBM Informix database called **stores**. You will access the contents of a table called **customer** and view other information about the stores database.

Connect to the stores database by creating a new connection to the database using the Data Source Explorer.

1. Return to the Java perspective by clicking the Java button in the upper right corner of the Data Studio window.
2. In the Data Source Explorer window, right-click the **Connections** folder and select **New**.
3. In the Select a database manager on window, select **Informix**.
4. In the **Drivers** section, select **Informix 11.5 - Informix JDBC Driver Default**.
5. In the Properties window, change the values for Database, Host, Port number, Server, User name, and Password as shown.
 - Database: stores

- Host: ids1150srvr
 - Port number: 9088
 - Server: demo_on
 - User name: informix
 - Password: informix
6. Click **Test Connection**. If the connection is successful, click **OK**; otherwise, go back and check the values you entered.
 7. Click **Finish** to establish the connection to the stores database.

You can now use Data Studio to explore the stores database objects.

1. Return to the **Data** perspective by clicking the **Data** button in the upper-right portion of the Data Studio window. Click the **Workbench** icon (if necessary).
2. In the Data Source Explorer window, expand the **Database Connections** tab and navigate to the customer table by selecting: **Database Connections > stores > stores > Schemas > Informix > Tables**.
3. Right click the **customer** table and select **Data > Sample Contents**. The contents of the customer table are displayed in the Data Output view. Adjust the size of the view as necessary to view more or less of the data.

Data Studio has many powerful database object capabilities. For example, you can view the column statistics for a column in a table, and display the number of occurrences of each data value. Table statistics information is stored in the system catalog. For example, to view sample contents of the city table, perform the following steps:

1. Right click **City**, then select **Data > Sample Contents**. The city column statistics are view by clicking the **Result1** tab.

You can also view stored procedure code.

1. Expand the **Stored Procedures** twistie.
2. Double-click the **cdrcmdproc** stored procedure.
3. On the **Properties** tab, click the **Source** option to view the stored procedure code.

DDL Creation using the Data Source Explorer View

This section shows how to create DDL.

1. Return to the **Data Source Explorer** view in the **Data** perspective.
2. Navigate to the **customer** table.
3. Right click **customer** then select **Generate DDL**
4. Click **Next** then click **Next** again.

You can now see the schema for the customer table. You can modify the DDL in this window. You can modify the DDL, execute it, save it, and so on.

5. Click **Cancel** to close the Generate DDL window.

Working with Value Distributions

Now you will display value distributions in the orders table.

1. Return to the **Data Source Explorer** view in the **Data** perspective
2. Navigate to the **orders** table.
3. Right click **orders**, then select **Value Distributions > Multivariate**.

Find the **orders** tab where each column of the table is displayed. This tabbed window is split horizontally; the top part of the window displays the columns and the bottom part displays the column value statistics in graphical form. View the distribution graph for each column by highlighting it.

Creating Tables

In this section, you will create a new table.

1. Return to the Data Source Explorer view in the **Data** perspective.
2. Right click **Tables**, then select **Create > Table**.
3. Select the **Table1** tab.
4. Enter a table name and label.
5. Click the **Columns** tab.
6. Add or delete columns using the **Add** and **Delete** icons.
7. Click the **Preview DDL** button and inspect the DDL that is used to create the table.
8. Click the Run DDL button to run the DDL and create the table.

Feel free to continue exploring the stores database. However, do not change or drop existing tables and/or data as these will be needed for subsequent exercises.

Closing the Database Connection

To close the connection to the database:

- Right click **stores** then select **Disconnect**

You can also delete the tabbed windows (views) that you created during this exercise, but do not delete the Data Source Explorer view. Also, do not delete the either the Data Perspective or the Java Perspective.

Enhanced Error Message Lookup Using the Basic Text Search (BTS) Extension

The BTS extension is a fast and flexible way to make free-form text searches available in applications.

In this tutorial, you will use the Basic Text Search (BTS) extension to create a searchable database of Informix error messages.

The usual method for looking up an Informix error message is to use the **finderr** utility. The **finderr** utility is a command-line utility that takes an error number as an argument. Suppose you want to do more than just find an SQL error by number; perhaps you are troubleshooting a problem and want to find which errors are related to referential key constraints or virtual-index interface (VII) indexes.

A simple way to provide a more sophisticated search for Informix error messages is to load the error numbers and their definitions into a database and create a Basic Text Search (BTS) index on the text. The Basic Text Search extension is an IDS 11.50 feature that uses the power of the cLucene open source text search engine to support searching for words and phrases in unstructured text.

What follows is a step-by-step guide to creating a database to index the Informix error messages and search them using the BTS extension.

Create a smart blobspace

The Basic Text Search extension makes use of a logged sbospace to store index information. To create a smart blobspace use the **onspaces -c** parameter.

1. Open a terminal window and enter the following commands to create the directories and set permissions:

```
touch /home/informix/chunks/bts_sbospace
chown informix:informix /home/informix/chunks/bts_sbospace
chmod 660 /home/informix/chunks/bts_sbospace
```

2. Use the **onspaces** command to create the blobspace:

```
onspaces -c -S bts_sbospace -p /home/informix/chunks/bts_sbospace
-o 0 -s 20480 -Df LOGGING=ON
```

Create a BTS Virtual Processor

The Basic Text Search engine runs in its own Virtual Processor class, so to use BTS add a **VPCLASS** entry to the **onconfig** file.

1. Open the **onconfig** file using a text editor.
2. Add the following information to the **onconfig** file:

```
# for BTS search
VPCLASS bts,noyield,num=1
```

3. Restart IDS for the **onconfig** parameter to take effect.

The maximum number of BTS VPs is limited to 1 in IDS 11.

Create the Error Message database

The **finderr** database uses a single LVARCHAR column table with each row representing one error message (including number and text). In the example below, no dbspace is specified in the CREATE DATABASE statement, which means that the database is created in the root dbspace:

1. In a terminal window, enter the following commands:

```
dbaccess <<EOF
create database finderr with buffered log;
create table errtable(errcol lvarchar(10000));
EOF
```

Load the error message data

The list of Informix error messages is supplied with the IDS media in a file called **errmsg.txt**. In distributions using the standard US locales the file is found here on Linux or UNIX systems: **\$INFORMIXDIR/msg/en_us/0333/errmsg.txt**.

In order to convert the **errmsg.txt** file into a format that can be loaded into the **errtable** table add a pipe "|" character between each message, and escape every end-of-line with a backslash "\" character. This conversion involves more than a simple **sed** script because the number of blank lines between messages is not always consistent. A formatted error message file can be found in:

/opt/IBM/informix/IDS11-FirstSteps/04_BTS/errmsg.unl

1. In a terminal window, enter the following command to load the database:

```
echo load from '/opt/IBM/informix/IDS11-FirstSteps/04_BTS/errmsg.unl'
insert into errtable | dbaccess finder
```

Register the BTS Extension

Before you can create a BTS index, register the BTS DataBlade module with the **finderr** database. Registering the BTS DataBlade module is done by running the **blademgr** command line utility as user informix:

1. In a terminal window, start the **blademgr** utility:
blademgr
2. Type the following to register the **finderr** database:
register bts.2.00 finderr
3. Type "y" when prompted with: "Register module bts.1.00 into database finderr? [Y/n]."
4. After the database is created, exit from the **blademgr** utility:
quit

Create the BTS Index

With the BTS extension registered and an external dbspace in place, you can create a BTS index on the **errcol** column.

```
dbaccess finderr <<EOF
create index err_bts on errtable (errcol bts_lvarchar_ops)
  using bts (delete='immediate') in bts_sbspace;
EOF
```

In this example, **bts_lvarchar_ops** is an operator class that corresponds to the LVARCHAR data type. An operator class represents a set of functions for IDS to associate with the BTS access method. A similar operator class exists for every data type that can be represented as text (BLOB, CHAR, CLOB, LVARCHAR, NCHAR, NVARCHAR, VARCHAR).

The (**delete='immediate'**) clause is an optional clause that tells IDS to remove index information automatically after every delete operation. Without this clause, indexes on dynamic data will need to be manually optimized using the **bts_index_compact()** function. For a static database, such as the one in this example, this clause effectively makes no difference.

The BTS index can take a few minutes to create, and once created, the database is ready for queries.

Running Queries

With the index in place, queries on the **errcol** column would take the form:

```
select errcol from errtable where bts_contains(errcol,"xxx");
```

where "xxx" represents a search string that conforms to the Lucene search syntax. For example, suppose you wish to search for error messages that contain the words "unlock" and "table" but not "transaction", the query would be:

```
select errcol from errtable where bts_contains(errcol,
  "unlock AND table AND NOT transaction");
```

You can replace the command-line **finderr** utility by writing a shell script that constructs a similar query based on the command-line arguments:

```
dbaccess finderr <<EOF
select errcol from errtable where bts_contains(errcol,"$*");
EOF
```

Here is example output from the script:

Database selected.

```
errcol -12020 Smart Large Objects: Cannot unlock table entry.
```

If this error recurs, note all circumstances and contact Technical Support at tsmail@us.ibm.com.

```
errcol -291 Cannot change lock mode of table.
```

The current LOCK TABLE statement cannot be executed because you have already locked the same table using a different mode (EXCLUSIVE or SHARE). To change the lock mode, arrange to unlock the table before you lock it again.

```
errcol -153 ISAM error: not in ISMANULOCK mode.
```

The ISAM processor has been asked to lock or unlock the current file (table), but the file was not opened in the appropriate mode. For C-ISAM programs, review the uses of `isopen`, and make sure that the ISMANULOCK flag is passed when the program opens a table for manual locking. If the error recurs, note all circumstances and contact Technical Support at tsmail@us.ibm.com.

3 row(s) retrieved.

To use this script in production it would also need to escape characters that are treated as special characters by the BTS engine. For example with the script in its current form, to look up an error number such as -291, it would be necessary to escape the negative sign twice; once for the command shell and once for the BTS parser:

```
btserr \\-291
```

A good way to use an index like this would be as part of a PHP or Java web application that could parse any special characters before constructing the query, and be available to all users. Another method is to make the error message database available through an instant message chat bot.

Summary

In this tutorial you created and loaded the **finderr** database. The Blade Manager was then used to register the BTS module to the **finderr** database, and the index was created. Finally, scripts were used to read from the database.

Using the IBM Informix Spatial DataBlade Module

The objective of this tutorial is to show you how to register the IBM Informix Spatial DataBlade module for an existing database and how to create and use geometry objects using ESQL/C and the Client-side Java API. The Client-side Java API consists of Java classes that enable you to work with Geodetic objects from Java client-side programs.

The following programs are included in the ESQL/C demonstration directory:

Table 10. ESQL/C Demonstration Programs

Program Name	Description
load_shapes.ec	Creates a table with an ST_Geometry column and inserts examples of all geometry types in ESRI shape format, using the SE_GeomFromShape function.
load_wkb.ec	Creates a table with an ST_Geometry column and inserts examples of all geometry types in OGIS WKB format, using the ST_GeomFromWKB function.
geomfromshape.ec	Inserts a polygon into a table in ESRI shape format using the SE_GeomFromShape function.
geomfromwkb.ec	Inserts a polygon into a table in OGIS WKB format using the ST_GeomFromWKB function.
asbinary.ec	Retrieves data in OGIS WKB format.
asshape.ec	Retrieves data in ESRI shape format.

The Java API demo includes the following three example programs:

Table 11. ESQL/C Demonstration Programs

Program Name	Description
GeometryToWKT	Reads geometry objects from an IBM Informix spatial table and converts them into WKT (well-known text) strings. The program first creates a table, inserts geometry data, then queries the table. The output of this program shows the well-known text representation of sample geometries.
GeometryToArray	Creates new geometries using a geometry factory, then reads coordinate data out of the geometry objects. The program first creates several geometry objects, stores them in a Java vector, then uses two techniques to read coordinate data out of the geometry objects: <ul style="list-style-type: none"> • <code>IfxGeometry.toCoordArray()</code> • <code>IfxGeometry.toPointArray()</code>
CoordRefCreate	Creates new coordinate reference objects and inserts them into the SPATIAL_REFERENCES table. The program creates a new CoordRef object, then serializes it as a new SRS row into the SPATIAL_REFERENCES table. The output of this program shows the result of a query to retrieve the new spatial reference systems from the database.

The examples are included with IBM Informix Spatial DataBlade in the following directory:

`$INFORMIXDIR/extend/spatial.8.21.UC2/examples.`

Both the ESQL/C and the Java API examples assume that you are logged in to the informix account. If you are logged in as user developer, switch to informix account using the **su** command.

For these examples, it is assumed that the **demo_on** instance is up and running. To make sure the server is on line, type the following in a terminal window:

`onstat -`

If the server is not on line, start it by entering the following command as user informix:

```
startDemo
```

In the steps below, you will create a new database with logging, register the IBM Informix Spatial DataBlade and R-tree module, and compile and run the java programs:

1. Create a database named **geodemodb** with logging enabled by entering the following from a command prompt:

```
dbaccess <<EOF
create database geodemodb with log;
EOF
```
2. Register the Spatial DataBlade and R-Tree modules:
 - a. Start the Blade Manager utility by entering the following at a command prompt:

```
bladmgr
```
 - b. Enter the following command at the Blade Manager prompt to register the Spatial module:

```
register spatial.8.21.UC2 geodemodb
```
 - c. Press Enter when Blade Manager prompts you about whether to register the module.
 - d. Enter the following command at the Blade Manager prompt to register the R-Tree module:

```
register ifxrltree.2.00 geodemodb
```
 - e. Press Enter when Blade Manager prompts you about whether to register the module.
 - f. Enter the following command to list the registered DataBlade modules in the geodemodb database:

```
list geodemodb
```
 - g. Exit the Blade Manager utility:

```
exit
```

Building and Running the ESQL/C Examples

You build the sample programs by running the make utility.

1. Change to the directory in which the ESQL/C programs reside:

```
cd /opt/IBM/informix/extend/spatial.8.21.UC2/examples/esqlc
```
2. Build the examples:

```
make
```
3. Run the examples by entering the example program followed by the database name (geodemodb). For example:

```
./load_shapes geodemodb
```

Building and Running the Java Examples

You run the Java examples by first setting the CLASSPATH environment variable and then running the example programs.

1. Export the CLASSPATH environment variable:

```
export CLASSPATH=/opt/IBM/informix/extend/spatial.8.21.UC2/spatial.jar:
$CLASSPATH
```
2. Change to the directory in which the Java example programs reside:

```
cd $INFORMIXDIR/extend/spatial.8.21.UC2/examples/java
```

3. Build the examples:

```
javac -Xlint:unchecked *.java
```

4. Run the examples by using the following syntax:

```
java program_name connection_URL
```

In the example, *program_name* is the name of the example program to run and *connection_URL* specifies the JDBC driver, the server name and port number, the database to use, and the user name and password for the database. For example:

```
java CoordRefCreate jdbc:informix-sqli://ids1150srvr:9088/  
geodemodb:informixserver=demo_on;user=informix;password=informix
```

Summary

In this tutorial, saw a sample of the Java and ESQL/C demonstration programs, you created a new database with logging enabled, registered DataBlade modules using the Blade Manager utility, and compiled and ran both Java and ESQL/C demonstration programs.

Appendix. Accessibility

IBM strives to provide products with usable access for everyone, regardless of age or ability.

Accessibility features for IBM Informix Dynamic Server

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility Features

The following list includes the major accessibility features in IBM Informix Dynamic Server. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers.
- The attachment of alternative input and output devices.

Tip: The IBM Informix Dynamic Server Information Center and its related publications are accessibility-enabled for the IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

Keyboard Navigation

This product uses standard Microsoft® Windows navigation keys.

Related Accessibility Information

IBM is committed to making our documentation accessible to persons with disabilities. Our publications are available in HTML format so that they can be accessed with assistive technology such as screen reader software. The syntax diagrams in our publications are available in dotted decimal format.

You can view the publications for IBM Informix Dynamic Server in Adobe Portable Document Format (PDF) using the Adobe Acrobat Reader.

IBM and Accessibility

See the *IBM Accessibility Center* at <http://www.ibm.com/able> for more information about the commitment that IBM has to accessibility.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licenses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating

platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Index

A

- accessibility 41
 - keyboard 41
 - shortcut keys 41
- Accessibility 41
- Apache Server
 - starting and stopping 8

B

- Basic search text extension
 - introduction 34

C

- Client SDK 3
- Cluster
 - configuration 13
- Common Client 17

D

- Data Server Driver for JDBC/SQLJ 3
- Data Server Driver for ODBC 3
- Data Studio 4
 - introduction 31
- Database
 - Connecting using Data Studio 31
 - defined 20
- DataBlade 3
- Demonstration programs 7
- Detective Game 20
- disability 41

E

- ESX
 - Installation 5
- Exchanging data 8

F

- File system 8

H

- High-Availability Cluster
 - configuration 13

I

- IDS
 - Developer edition 1
 - Enterprise edition 1
 - Express edition 1, 3
 - Workgroup edition 1
- IDS Detective Game 20

- Installation
 - ESX 5
- Installed software 1, 3

J

- Java 8
- JDBC Driver 3, 17

K

- Keyboard layout and settings 8

L

- Linux 4

M

- Mac OS X 4
- MACH11
 - demonstration scripts 13
- Memory 8

N

- Network configuration 8

O

- OAT (Open Admin Tool) 3
 - starting 14

P

- Passwords 2
- PHP 18
- Plugin
 - installing 14

S

- Screen resolution 8
- Scripts
 - installed with Virtual Appliance 16
 - MACH11 13
- shortcut keys
 - keyboard 41
- Software 3
 - system requirements 2
- Spatial DataBlade 3
- Spatial DataBlade Module
 - introduction 37
- System requirements 2

U

- User ID 2

V

- Virtual Appliance
 - configuring 8
 - definition 1, 2
 - installed software 1
 - installing 4
- VMware
 - ESX 5

W

- Windows 4



Printed in USA