

# **Ingres® 2006 Release 3 for Linux**

## **Quick Start Guide**

**INGRES®**

July 2007

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Ingres Corporation ("Ingres") at any time.

This Documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of Ingres. This Documentation is proprietary information of Ingres and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this Documentation for their own internal use, provided that all Ingres copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. The user consents to Ingres obtaining injunctive relief precluding any unauthorized use of the Documentation. Should the license terminate for any reason, it shall be the user's responsibility to return to Ingres the reproduced copies or to certify to Ingres that same have been destroyed.

To the extent permitted by applicable law, INGRES PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL INGRES BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USER OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF INGRES IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

The use of any product referenced in this Documentation and this Documentation is governed by the end user's applicable license agreement.

The manufacturer of this Documentation is Ingres Corporation.

For government users, the Documentation is delivered with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013 or applicable successor provisions.

Copyright © 2007 Ingres Corporation. All Rights Reserved.

Ingres, OpenROAD, and EDBC are registered trademarks of Ingres Corporation. All other trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Contents

---

<b>Chapter 1: Installing Ingres</b>	<b>5</b>
Installing Ingres for Linux .....	5
Install Ingres for Linux Using the Installation Wizard.....	5
Installing Ingres for Linux at the Command Line .....	6
How You Access the Instance on Linux and UNIX .....	7
Start the Ingres Installation on Linux and UNIX .....	8
 <b>Chapter 2: Creating a Database</b>	 <b>9</b>
How You Create a Database.....	9
How You Create a Database Using Ingres Commands .....	9
Create a Database Using Createdb Command .....	10
Create a Table Using the CREATE TABLE Statement .....	11
Populate a Table Using the COPY Statement .....	12
 <b>Chapter 3: Using the Ingres Demonstration Application</b>	 <b>13</b>
Ingres Demonstration Application .....	13
What You Should Know .....	13
Start the Ingres Demonstration Application.....	14
Get Help on the Demonstration Application .....	14
 <b>Chapter 4: Connecting to Ingres from Eclipse</b>	 <b>15</b>
Required Software.....	15
Install Plug-ins .....	15
How You Connect to Ingres from Eclipse.....	16
Create a Connection Profile.....	17
Define an Ingres Driver.....	19
Create a New Database Connection .....	22
Connect to a Database.....	23
Create an SQL Scratch File .....	24
 <b>Chapter 5: Connecting to Ingres from Python</b>	 <b>27</b>
Requirements .....	27
Ingres Python DBI Driver and the Ingres ODBC Driver .....	27
Create a Connection to Ingres.....	28
Example—Connect to Ingres and Select from a Table .....	28

---

<b>Chapter 6: Connecting to Ingres from PHP</b>	<b>29</b>
Requirements .....	29
Download the PECL Extension .....	29
Install the Extension on UNIX or Linux .....	30
Configure Apache for UNIX/Linux .....	31
ingres_connect() Function—Connect to Ingres .....	32
Code Examples .....	33
 <b>Chapter 7: Connecting to Ingres from Perl</b>	 <b>35</b>
Requirements .....	35
Ingres Perl DBI Extension .....	36
How You Can Build and Install the Ingres Perl DBI Extension .....	36
Build the DBD::Ingres Extension .....	37
Test and Install the DBD::Ingres Extension .....	37
How You Can Use the DBD::Ingres Extension .....	38
Example—Connect to Ingres and Select from a Table .....	38
 <b>Appendix A: Frequently Asked Questions on Ingres for Linux</b>	 <b>39</b>
Does Ingres Shut Down Automatically? .....	39
What Character Set Should I Use on Japanese SUSE Linux 9? .....	39
Which Linux Shell Should I Use? .....	40
Why Is Createdb Issuing Error Messages? .....	40
How Do I Get the Function Keys to Work? .....	40
How Do I Map Function Keys? .....	41
Should I Modify System Kernel Parameters? .....	41
What Switches Do I Use to Compile C Applications? .....	42
Do I Need to Change Permissions? .....	42
Why Does the Fortran Compiler Fail When Using G77? .....	42
How Do I Get Characters to Display Properly? .....	43
 <b>Index</b>	 <b>45</b>

# Chapter 1: Installing Ingres

---

This section contains the following topics:

[Installing Ingres for Linux](#) (see page 5)

[How You Access the Instance on Linux and UNIX](#) (see page 7)

[Start the Ingres Installation on Linux and UNIX](#) (see page 8)

## Installing Ingres for Linux

Ingres is installed on Linux using RPM Package Manager. You can install Ingres using an installation wizard or from the command line.

**Note:** Installation of Ingres requires root access. If you are logged in as another user, you will need the root password.

For details on installation considerations and more information on using RPM to install and upgrade Ingres, see the *Installation Guide*.

### Install Ingres for Linux Using the Installation Wizard


The installation wizard easily installs Ingres without your having to know RPM commands.

#### To start the installation wizard

1. Run the `ingres_install` script located in the root directory of the Ingres distribution.

The first page of the Ingres Installation Wizard is displayed.

2. Respond to the Installation Wizard dialogs.

If you need more information about a specific item, click the item's information button .

After you provide the required information in the Installation Wizard dialogs, the installation program installs Ingres.

## Installing Ingres for Linux at the Command Line

You can install the Ingres RPM packages at the command line by doing either of the following:

- Using the `ingres_express_install` command
- Using RPM commands

### `ingres_express_install` Command—Install Ingres

The `ingres_express_install` command quickly installs Ingres RPM packages.

This command has the following format:

```
ingres_express_install [instance_ID] [instance_location]
```

#### ***instance\_ID***

Specifies the Ingres instance ID to use for the instance.

#### ***instance\_location***

Identifies the directory into which you want to install Ingres.

### **Examples: `ingres_express_install` Command**

This command installs all RPM packages in the current working directory with the default configuration:

```
ingres_express_install
```

This command installs all RPM packages with the default configuration, but with an instance ID of A1:

```
ingres_express_install A1
```

## How You Access the Instance on Linux and UNIX

When the installation is complete, the instance is running.

To access your instance, you must source the environment file that was created during installation.

During installation, an environment file is written to the home directory of the operating-system user ID that was defined during installation (the default is "ingres"). The name of the environment file depends on the value of `II_INSTALLATION`.

To source the environment file created during installation, issue the following command.

**Note:** The following examples assume an operating-system user ID of ingres.

For bash:

```
. ~ingres/.ingXXbash
```

For tcsh:

```
source ~ingres/.ingXXtsch
```

where XX is the instance ID of the instance.

For other users to have access to the instance and the Ingres tools, they must have access to the `.ingXXbash` and `.ingXXtsch` scripts. The scripts can be copied to the home directory of any user.

## Start the Ingres Installation on Linux and UNIX

Use the `ingstart` command to start Ingres.

**Note:** If you are using a raw device for your transaction log file, you must configure the log file before starting your instance.

**Note:** If your operating system has shadow passwords, you must install the password validation program before starting a networked DBMS Server instance.

### To start the Ingres instance

1. Log on to your system through the system administrator account for the instance.
2. Enter the following command:

```
% ingstart
```

The `ingstart` command checks whether you have sufficient operating system resources to run the Ingres components, and whether the raw log file (if used) is configured. If these conditions are met, `ingstart` starts all servers that are part of your instance.

### To stop the Ingres instance

Enter the following command:

```
% ingstop
```

The instance is stopped.



# Chapter 2: Creating a Database

---

This section contains the following topics:

[How You Create a Database](#) (see page 9)

[How You Create a Database Using Ingres Commands](#) (see page 9)

## How You Create a Database

A database can be created using Visual DBA, if available in your environment, or by using Ingres commands and SQL statements.

The process for creating a database is as follows:

1. Create a database.
2. Create base tables for the database.
3. Populate the tables with data.

**Note:** Examples used in this guide are from the Ingres demonstration database, demodb.

## How You Create a Database Using Ingres Commands

The process for creating a database using Ingres commands and SQL statements is as follows:

1. Create a database using the createdb command.
2. Create tables for the database using the CREATE TABLE statement.
3. Populate the tables with data using the COPY statement.

## Create a Database Using Createdb Command

Use the createdb command to create a database. The user who creates a database becomes its database administrator. For a complete description of the createdb command, see the *Command Reference Guide*.

**Note:** You must have the createdb privilege to create a database. For details, see the *Database Administrator Guide*.

### To create the demodb database using the createdb command

Issue the following command:

```
createdb -i demodb
```

This command creates a public database named demodb on the local node in the default locations. The database is Unicode-enabled using Normalization Form C.

## Create a Table Using the CREATE TABLE Statement

The CREATE TABLE statement creates a new base table that is owned by the user issuing the statement. The CREATE TABLE statement creates entries in the system catalogs for the table created.

You can create tables using:

- The Ingres/Terminal Monitor
- Interactive SQL
- An Embedded SQL program
- Ingres/Applications-By-Forms (ABF) and Ingres/4GL (referred to as ABF/4GL)
- Tables utility

For a complete description of the CREATE TABLE statement, refer to the *SQL Reference Guide*.

### To create the airport table using the CREATE TABLE statement

The following statement creates the airport table with columns ap\_id, ap\_iatacode, ap\_place, ap\_name, and ap\_ccode:

```
CREATE TABLE airport(  
    ap_id integer not null not default,  
    ap_iatacode nchar(3) not null not default,  
    ap_place nvarchar(30),  
    ap_name nvarchar(50),  
    ap_ccode nchar(2)  
);
```

## Populate a Table Using the COPY Statement

The SQL statement COPY can be used to copy the contents of a file into a table.

**Note:** The input file for the following COPY statement is a binary file. To create this file, use this statement:

```
COPY TABLE airport () INTO 'airport.in';
```

### To load the airport table from a binary file

Issue the following statement:

```
COPY TABLE airport (  
    ap_id=integer,  
    ap_iatacode=nchar(3) ,  
    ap_place=nvarchar(30),  
    ap_name=nvarchar(50),  
    ap_ccode=nchar(2))  
FROM 'airport.in';
```

### To load the airport table from a CVS file

Alternatively, if the data file is in comma-separated value (.csv) format, you must specify the character types and the delimiter on the COPY statement. This method allows more flexibility and lets you skip fields. The COPY statement would look like this:

```
COPY airport (  
    ap_id=char(0) comma,  
    ap_iatacode=char(0) comma,  
    ap_place=char(0) comma,  
    ap_name=char(0) comma,  
    ap_ccode=char(0) nl)  
FROM 'airport.csv';
```

For more information on the COPY statement, see the *SQL Reference Guide* and the *Database Administrator Guide*.

# Chapter 3: Using the Ingres Demonstration Application

---

This section contains the following topics:

[Ingres Demonstration Application](#) (see page 13)

[What You Should Know](#) (see page 13)

[Start the Ingres Demonstration Application](#) (see page 14)

[Get Help on the Demonstration Application](#) (see page 14)

## Ingres Demonstration Application

Ingres provides a sample transactional based application that demonstrates common programming tasks required to develop an application using Ingres.

The application shows how information is retrieved from Ingres and presented in a user interface control. Novice users can "borrow" code from the application to use in their own application development.

The demonstration application provides a working example of the main tasks required to insert, update, and retrieve data from an Ingres database, including:

- SELECT queries that require parameters
- Execution of a row producing procedure
- INSERT, UPDATE, and DELETE queries
- Transaction commit and rollback
- Use of Unicode character types
- Handling of binary large objects (BLOBs)

## What You Should Know

The application developer should be familiar with the development language and environment, and somewhat familiar with SQL.

## Start the Ingres Demonstration Application

You can access the demonstration application if you opted to create and populate a demonstration database during the installation process.

When the application is started, it automatically connects to the database, demodb, on the local Ingres instance.

### To start the demonstration application

Run IngresDemoApp.exe, located in the following directory:

`$II_SYSTEM/ingres/demo/xxxxxx/travel/app`

where xxxxxx is the language, IDE, or software framework that the application is developed in. For example:

**Eclipse Java application:** `$II_SYSTEM/ingres/demo/Eclipse/travel/app`.

**Note:** The source code for the application resides in:  
`$II_SYSTEM/ingres/demo/xxxxxx/travel/solution`.

## Get Help on the Demonstration Application

Each user interface control that displays data retrieved from an Ingres database has contextual help that includes instructions, source code, and query excerpts.

### To display contextual help

The procedure for displaying contextual help depends on the application. For example:

**Eclipse Java Application:** Press the help tab in the results part of the window, or press F1.

# Chapter 4: Connecting to Ingres from Eclipse

---

This section contains the following topics:

[Required Software](#) (see page 15)

[Install Plug-ins](#) (see page 15)

[How You Connect to Ingres from Eclipse](#) (see page 16)

[Create a Connection Profile](#) (see page 17)

[Define an Ingres Driver](#) (see page 19)

[Create a New Database Connection](#) (see page 22)

[Connect to a Database](#) (see page 23)

[Create an SQL Scratch File](#) (see page 24)

## Required Software

To connect to Ingres from the Eclipse SDK, you must download and install the following software and necessary plug-ins, which you can obtain from the Eclipse Data Tools Project page at <http://www.eclipse.org/datatools/> (<http://www.eclipse.org/datatools/>).

- Eclipse Software Development Kit (SDK)
- Eclipse Modeling Framework (EMF)
- Graphical Editing Framework (GEF)
- Data Tools Platform (DTP)
- Ingres DTP plug-in

The required versions are described in the Ingres readme.

## Install Plug-ins

You can check which plug-ins your copy of Eclipse has by accessing the Eclipse Help menu. If your copy of Eclipse does not have the required versions, you must download and install them.

### To install a plug-in

1. Extract the downloaded plug-in into your Eclipse plug-in area.
2. Re-start Eclipse to load the plug-ins.

## How You Connect to Ingres from Eclipse

The process for connecting to Ingres from the Eclipse SDK is as follows:

1. Create a Connection Profile.
2. Define an Ingres driver.
3. Create a new database connection.
4. Connect to a database.
5. Create an SQL scratch file.

**Note:** Before you connect to Ingres, ensure that your Ingres instance is running.



## Create a Connection Profile

To connect to Ingres, you must create a Connection Profile. You can create a connection file from Eclipse's Database Development perspective.

**Note:** Depending on the development language you are using, you must use the appropriate connection method. The following procedure instructs how to create a JDBC connection profile. For more information on JDBC connectivity, see the *Connectivity Guide*.

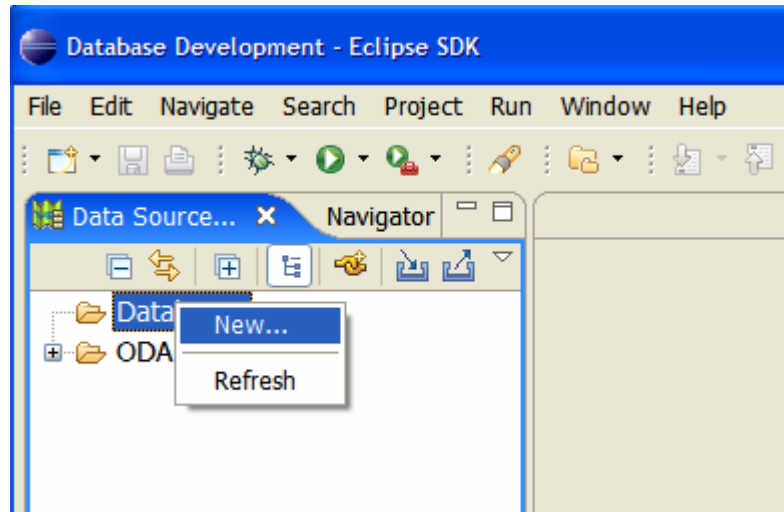
### To create a JDBC connection profile

1. Click Window, Open Perspective, Other in the Eclipse SDK.

The Open Perspective dialog appears.

2. Select the Database Development perspective and click OK.

The Data Source Explorer tab is added to the SDK.



3. In the Data Source Explorer, right-click the Database folder and select New from the context menu.

The New Connection Profile dialog appears.

4. Select the "Ingres JDBC Connection" wizard and click Next.

The Create connection profile panel is displayed.

5. Enter a name and description for your connection.

For example:



Click Next.

The Specify a Driver and Connection Details panel is displayed.

If you have already defined an Ingres driver, proceed to Create a New Database Connection. If you need to define a driver, continue with Define an Ingres Driver.

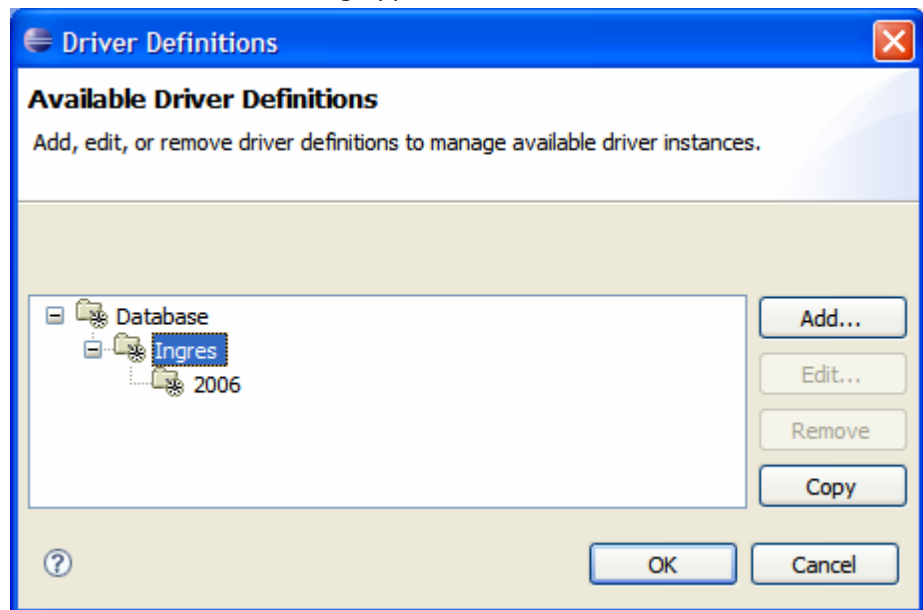
## Define an Ingres Driver

**Note:** If you have not defined an Ingres Driver, you must define one. If you have already defined an Ingres driver, skip to Create a New Database Connection.

### To define an Ingres driver

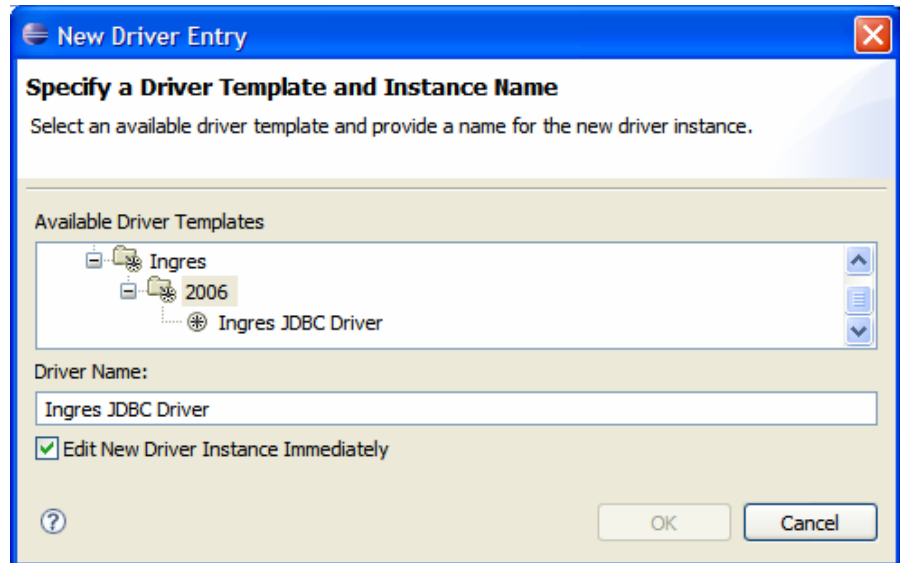
1. On the Specify a Driver and Connection Details panel of the wizard, click the Browse button next to the Select a driver drop-down menu.

The Driver Definitions dialog appears.



2. If no driver is listed in the Ingres 2006 branch, select the Ingres branch and click Add.

The New Driver Entry dialog appears.



3. In the Available Driver Templates list, select Ingres JDBC Driver.

The Driver Name field will contain: **Ingres JDBC Driver**.

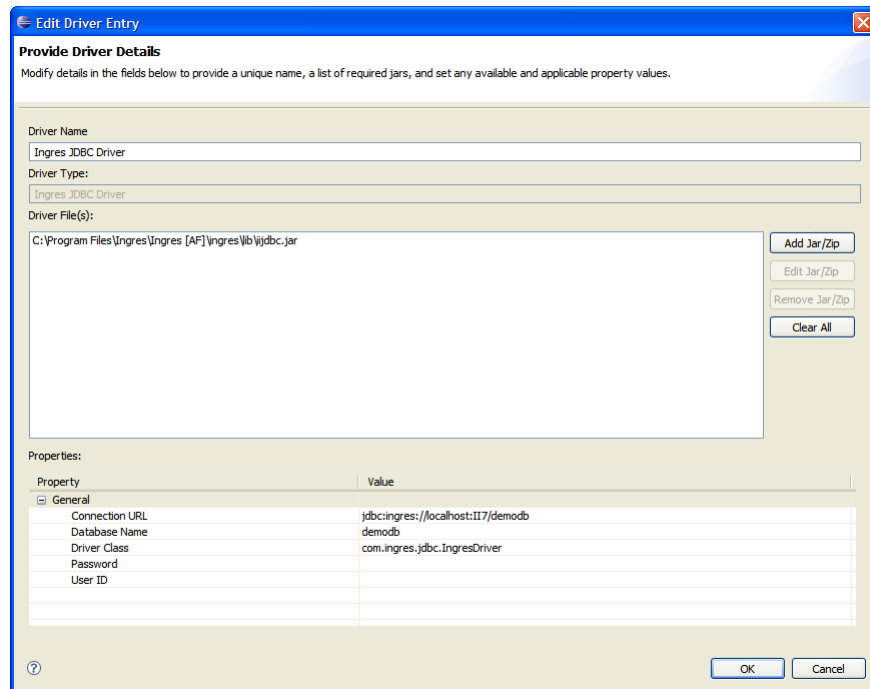
4. Ensure that the Edit New Driver Instance Immediately option is selected, and then click OK.

The Edit Driver Entry dialog appears.

**Note:** In most cases Eclipse will be unable to locate the driver file.

5. Select the Driver File iijdbc.jar and click Edit Jar/Zip. Browse to your Ingres Instance system area (defined in the Ingres system variable II\_SYSTEM) then to the subdirectories: ingres, lib to locate and select the driver file.

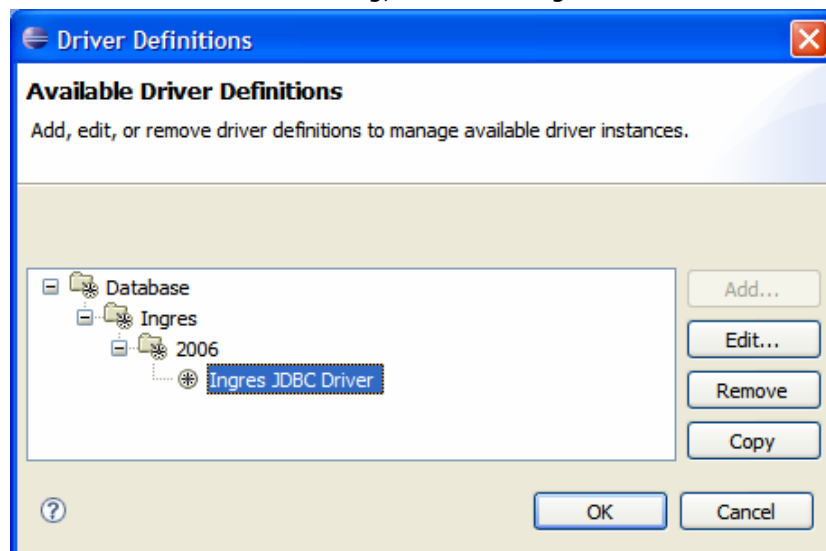
6. Ensure that the other driver details, including the Driver Class (**com.ingres.jdbc.IngresDriver**) and the instance ID (for example, **II**) are correct. For example:



**Note:** You can specify User credentials at connect time, rather than on the generic driver form.

Click OK.

7. In the Driver Definitions dialog, select the Ingres JDBC Driver from the list.



Click OK.

Continue with Create a New Database Connection.

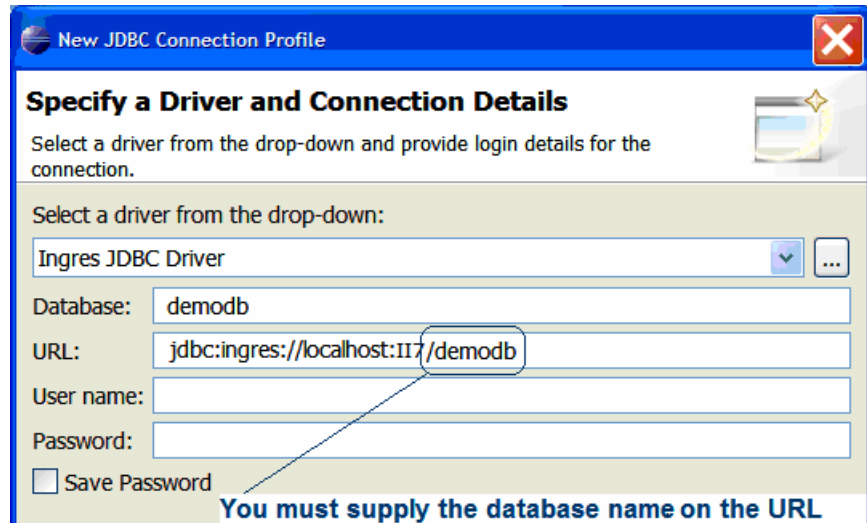
## Create a New Database Connection

After you have created an Ingres JDBC Driver, you can create a new database connection using this driver.

### To create a new database connection

1. Select the Ingres Driver from the list of Available Driver Definitions.
2. Specify the database name in the New JDBC Connection Profile dialog.

**Note:** You must specify the database name in the URL field. For example:  
`jdbc:ingres://localhost:II7/demodb`

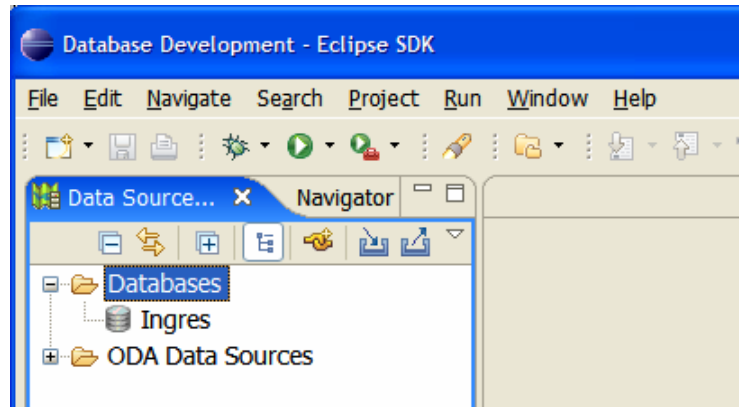


If your current user ID is not a valid Ingres user you must also specify User name and Password values. We recommend using the user credentials of the Ingres instance owner.

3. Test the connection by clicking Test Connection.

4. Click Finish to close the profile wizard.

An Ingres database is listed in the Data Source Explorer. For example:

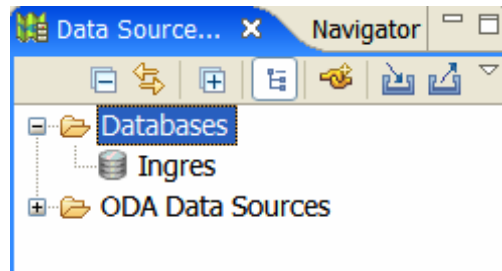


## Connect to a Database

After you have created a database connection profile, you can connect to the database from the Data Source Explorer.

### To connect to a database

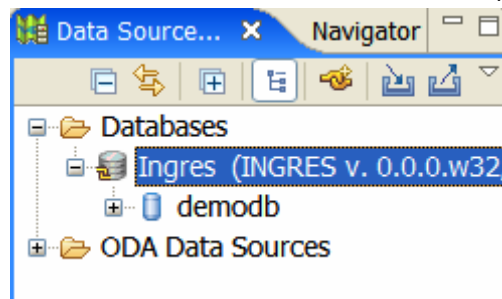
1. Open the Databases root folder in the Data Source Explorer.



Available databases are listed.

2. Right-click on the Ingres database and select Connect from the context menu.

The connection is established. For example:



You can now explore the objects in your database.

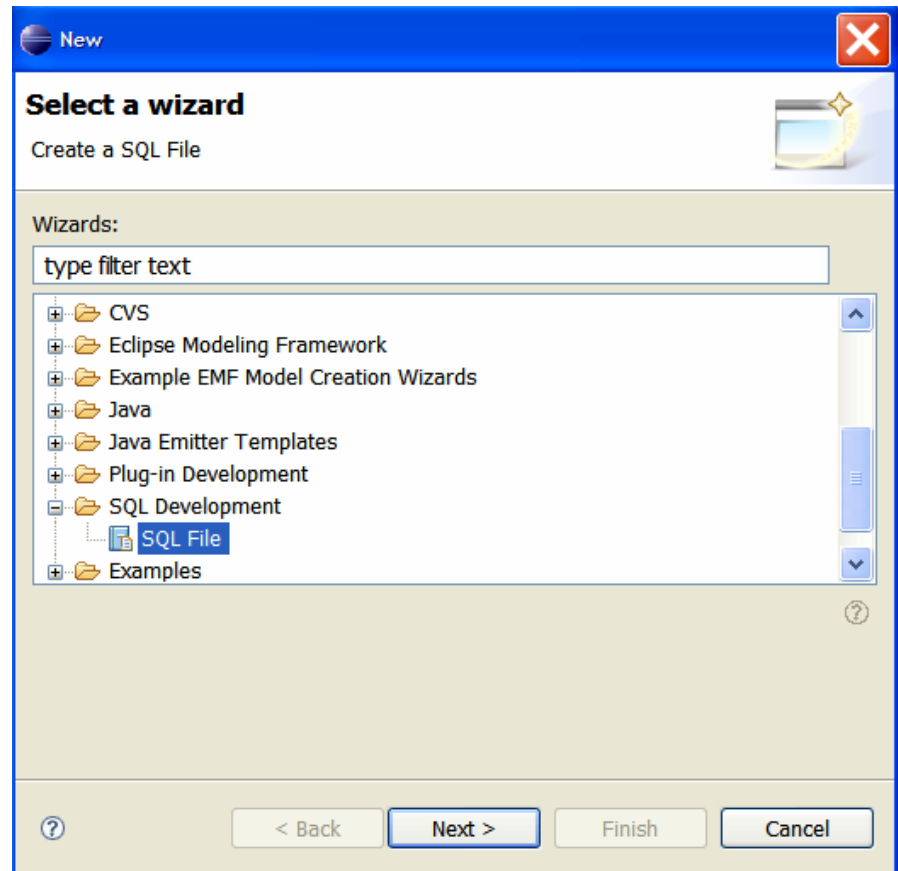
## Create an SQL Scratch File

An SQL scratch file is required to work with the database, and the file must belong to an Eclipse Project. If you have not created an Eclipse project, you must create one by clicking File, New, General Project. You can then create an SQL scratch file in the Eclipse SDK and connect it to a database.

### To create an SQL scratch file

1. Click File, New, Other.  
The New dialog appears.
2. Open the SQL Development folder and select the SQL File wizard.

For example:



Click Next.

The SQL File wizard appears.

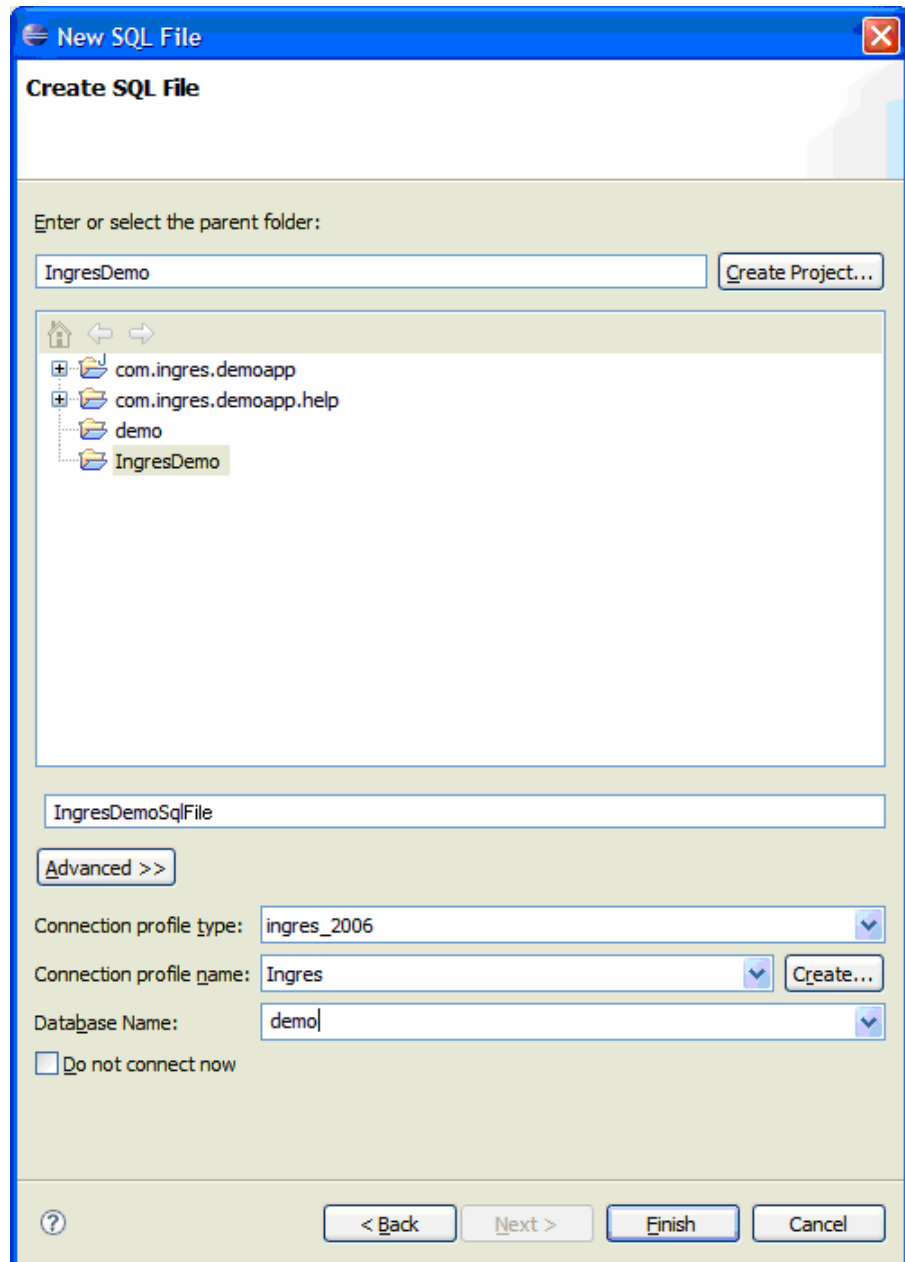


3. Select the parent folder.

The folder name is the same as the project name.

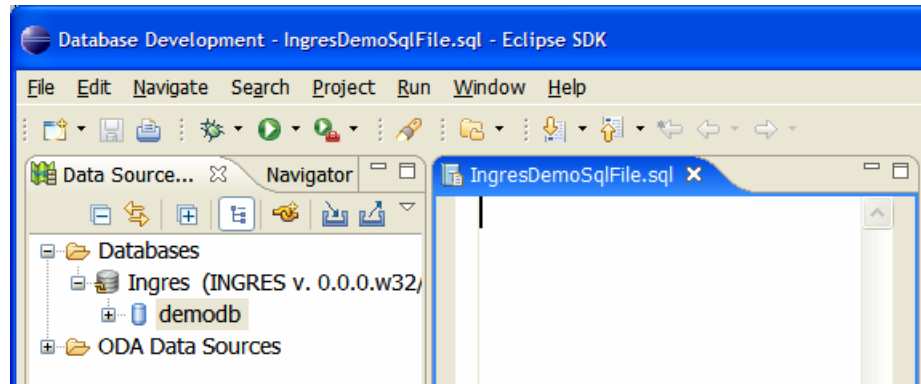
4. Enter a file name.

For example, "IngresDemoSqlFile":



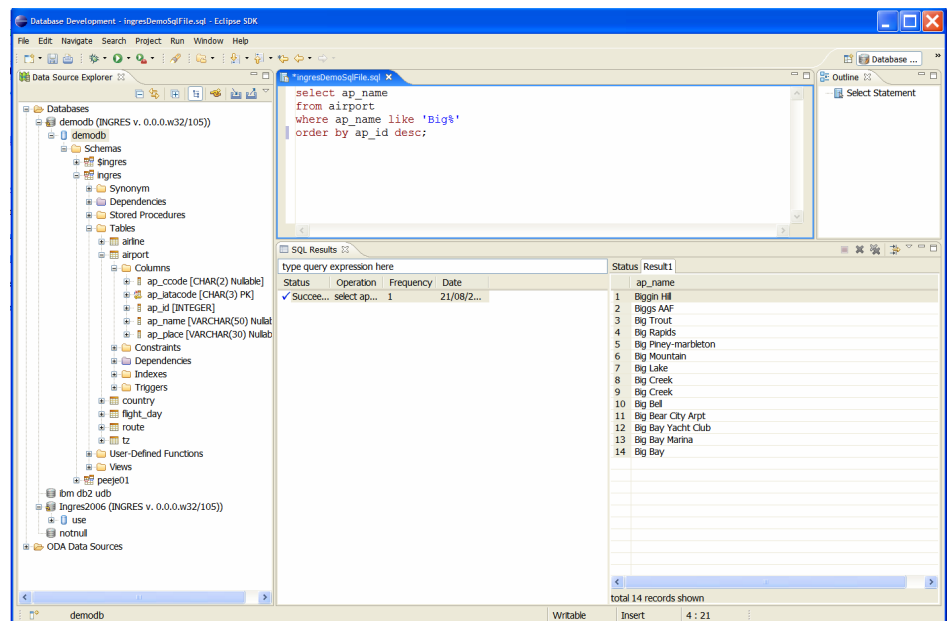
5. Set the connection details as shown, and then click Finish.

The Eclipse editor opens the new SQL file you created.



The SQL scratch file is connected to the database.

You can now issue SQL requests. Use the context menu or shortcut keys to execute your request. For example:



# Chapter 5: Connecting to Ingres from Python

---

This section contains the following topics:

[Requirements](#) (see page 27)

[Ingres Python DBI Driver and the Ingres ODBC Driver](#) (see page 27)

[Create a Connection to Ingres](#) (see page 28)

[Example—Connect to Ingres and Select from a Table](#) (see page 28)

## Requirements

To connect to Ingres from Python, you must meet the following prerequisites:

- Have Python installed in your development environment
- Build or download the Ingres Python DBI driver
- Create a connection from Python to Ingres

You can download the latest version of Python from <http://www.python.org/> (<http://www.python.org/>).

## Ingres Python DBI Driver and the Ingres ODBC Driver

To connect to Ingres from Python using the Ingres Python DBI driver, you must build and install the driver. Instructions are included in the Ingres Python DBI Driver readme, available on <http://www.ingres.com/downloads/prod-download-drivers.php> (<http://www.ingres.com/downloads/prod-download-drivers.php>).

The Ingres Python DBI driver requires that you have installed Ingres and the Ingres ODBC driver. For more information about the Ingres ODBC driver, see the chapter “Understanding ODBC Connectivity” in the *Connectivity Guide*. The Ingres ODBC driver is located in the following Ingres directory: `ingres\lib` (shared library).

## Create a Connection to Ingres

To connect to Ingres, you must instantiate an instance of the Connection class using the `ingresdbi.connect()` function.

If you have created a DSN, you could use the following code:

```
conn = ingresdbi.connect(dsn="myDSN")
```

Or you could connect directly to an Ingres database:

```
conn = ingresdbi.connect(database="myDB")
```

The `ingresdbi.connect()` function has other keywords that you can specify. For complete information, see the Ingres Python DBI Driver readme.

## Example—Connect to Ingres and Select from a Table

The following is a simple segment of code that connects to the Ingres database `iidbdb` and selects from `iitables`.

```
import ingresdbi

conn = ingresdbi.connect(database='iidbdb')
curs = conn.cursor()
curs.execute('select table_owner, table_name from iitables')
for x in curs:
    print x
curs.close()
conn.close()
```

# Chapter 6: Connecting to Ingres from PHP

---

This section contains the following topics:

[Requirements](#) (see page 29)

[Download the PECL Extension](#) (see page 29)

[ingres\\_connect\(\) Function—Connect to Ingres](#) (see page 32)

[Code Examples](#) (see page 33)

## Requirements

To connect to Ingres from PHP, you need to download or build the PECL extension and install it. (*PECL* stands for PHP Extension Community Library, which is a repository for PHP extensions.)

## Download the PECL Extension

Download the source code for the PECL extension for Linux from the following page:

<http://pecl.php.net/package/ingres> (<http://pecl.php.net/package/ingres>)

Or you can build the extension from the latest source code, available from the PHP project's Concurrent Versions System (CVS).

The source code for the Ingres PECL library is stored in the PHP project's CVS server, *cvs.php.net*. A CVS client is required to access this CVS server. Most Linux distributions have a CVS client available for installation from the installation media an online repository.

## Install the Extension on UNIX or Linux

The installation process for the PECL Ingres library requires a PHP development environment.

**Note:** You must first obtain the source code from CVS (see Download the PECL Extension) or <http://pecl.php.net/get/ingres> (<http://pecl.php.net/get/ingres>).

### To install the extension on UNIX or Linux

1. Examine your PATH statement to determine if it contains one of the following commands:

- phpize
- php-config

**Note:** Certain distributions of Linux rename these files to avoid version conflicts. To build the PECL Ingres library you need a C compiler.

2. Ensure that the environment variable II\_SYSTEM is defined; use the following command to confirm that it is set correctly:

```
echo $II_SYSTEM
```

3. Move to the directory that contains the source code for the extension.
4. Generate the configuration file for the extension by issuing the following command:

```
phpize
```

5. Generate the Makefile needed to build the extension:

```
./configure --with-ingres
```

6. Build the extension:

```
make
```

7. Install the extension:

```
make install
```

**Note:** Root access may be required to issue this command.

8. Edit the php.ini file, adding the following line:

```
extension=ingres.so
```

9. Ensure that the web server process owner is a valid Ingres user:

**Note:** If you use Apache, the Apache directive, *User*, found in httpd.conf, defines which operating system user runs Apache.

## Configure Apache for UNIX/Linux

If you are using Apache, you must configure it for use with Ingres and the PHP extension. Because the Apache web server does not make available operating system environment variables automatically, additional steps are needed.

**Note:** The following procedure assumes that Ingres is installed to `/opt/Ingres/II`.

1. The shared library `mod_env` must be loaded to pass variables from the operating system to Apache. Generally, this can be done using the Apache configuration directive:

```
LoadModule env_module modules/mod_env.so
```

**Note:** Certain Linux distributions use alternative mechanisms for configuring the modules available to Apache. Check the documentation for your operating system to see what steps are required.

2. Add the variables `II_SYSTEM` and `LD_LIBRARY_PATH` to Apache. Edit the configuration file, `httpd.conf`, adding the following to the bottom of the file:

```
SetEnv II_SYSTEM /opt/Ingres/II
SetEnv LD_LIBRARY_PATH /opt/Ingres/II/ingres/lib
```

3. Restart the Apache server to initiate the changes.

## ingres\_connect() Function—Connect to Ingres

You connect to Ingres using the `ingres_connect()` function and disconnect using `ingres_close()`. `ingres_connect()` returns an Ingres II link resource on success, or `FALSE` on failure.

This function has the following syntax:

```
resource = ingres_connect([database[,username[,password]]])
```

### **database**

Specifies a database with the following syntax:

```
[vnode_id:] dbname[/svr_class]
```

### **vnode\_id**

Specifies the virtual node name used to connect to a remote machine

### **dbname**

Specifies the database name

### **svr\_class**

Specifies the Ingres server class, which defaults to `INGRES` if not specified. It is used when connecting to different server classes.

### **username**

Specifies an Ingres user name to use for the connection

### **password**

Specifies the password for the user name

**Note:** If any parameters are missing, `ingres_connect()` uses the values in `php.ini` for `ingres.default_database`, `ingres.default_user`, and `ingres.default_password`.

### **Example: ingres\_connect()**

```
<?php
$link = ingres_connect("mydb", "username", "password");
    or die("Could not connect");
echo "Connected successfully";
ingres_close($link);
?>
```

**Note:** You can use `ingres_pconnect()` function to create a persistent connection.



## Code Examples

The following are examples of PHP code you can use for various Ingres operations.

### Example: Error checking

```
<?php
$link = ingres_connect("mydb", "username", "password");
if (ingres_errno($link) != 0) {
    echo ingres_errno($link) . " : " . ingres_error($link) . "<BR/>\n";
}
?>
```

### Example: Simple query

```
<?php
$link = ingres_connect("mydb", "username", "password");
// Gives a list of the tables
$sql = "select * from iirelation order by relid asc";
$rc = ingres_query($sql,$link);
// Do some error checking...
while ( $iirelation = ingres_fetch_object($link) ) {
    echo $iirelation->relid "<BR/>\n";
}
?>
```

### Example: Query with parameters

```
<?php
$link = ingres_connect("iiddb", "ingres", "ingres");
// Gives a list of the tables based on a parameter
$sql = "select * from iirelation where relowner = ? order by relid asc";
$params["owner1"] = ("usrname");
$rc = ingres_query($sql,$link,$params);
// Do some error checking...
while ( $iirelation=ingres_fetch_object($link) ) {
    echo $iirelation->relid "<BR/>\n";
}
?>
```

### Example: Loading a BLOB

```
<?php
    // Fetch the image to be inserted
    $handle = fopen ("username.png","r");
    $login_image = stream_get_contents($handle);
    fclose($handle);
    // Set up the query
    $sql = "insert into login_images values (?,?)";
    // Type the parameters being passed
    $types = "vB"; // varchar, BLOB
    // Set up the parameter values
    $params["login"] = "username";
    $params["image"] = $login_image;
    // Execute
    $rc = ingres_query($sql,$link,$params,$types);
?>
```

# Chapter 7: Connecting to Ingres from Perl

---

This section contains the following topics:

[Requirements](#) (see page 35)

[Ingres Perl DBI Extension](#) (see page 36)

[How You Can Build and Install the Ingres Perl DBI Extension](#) (see page 36)

[Build the DBD::Ingres Extension](#) (see page 37)

[Test and Install the DBD::Ingres Extension](#) (see page 37)

[How You Can Use the DBD::Ingres Extension](#) (see page 38)

[Example—Connect to Ingres and Select from a Table](#) (see page 38)

## Requirements

To connect to Ingres from Perl, you must meet the following prerequisites:

- Install and test Perl in your development environment
- Build and test the Ingres Perl DBI (database interface) driver
- Create a connection from Perl to Ingres

You can download the latest version of Perl from <http://www.perl.org/> (<http://www.perl.org/>).

System requirements and other installation information are contained in the Ingres Perl DBI Extension readme, available from <http://www.ingres.com/downloads/prod-download-drivers.php> (<http://www.ingres.com/downloads/prod-download-drivers.php>).

## Ingres Perl DBI Extension

The Ingres Perl DBI is a database extension for the Perl DBI system that enables access to Ingres databases. It is built on top of the standard Perl DBI extension. The driver supports database access to Ingres 2006 and prior versions of Ingres.

The source code is available for download from the Comprehensive Perl Archive Network (CPAN) repository for Perl extensions at <http://search.cpan.org/dist/DBD-Ingres/> (<http://search.cpan.org/dist/dbd-ingres/>).

For more information, see the Ingres Perl DBI Driver readme, available from <http://search.cpan.org/~htoug/DBD-Ingres-0.51/Ingres.pm> (<http://search.cpan.org/~htoug/DBD-Ingres-0.51/Ingres.pm>).

## How You Can Build and Install the Ingres Perl DBI Extension

Before you can use the Ingres Perl extension, DBD::Ingres, you must first build and install it. Building the extension requires a C development environment and Ingres ESQL/C. The steps required to build the extension are common to all platforms except for a slight difference between the commands used.

## Build the DBD::Ingres Extension

Perform the following procedure to download and build the DBD::Ingres extension.

### To download and build the extension

1. Download the latest version of the DBD::Ingres extension from <http://search.cpan.org/dist/DBD-Ingres/> (<http://search.cpan.org/dist/dbd-ingres/>).

2. Use a suitable tool to extract the downloaded files.

A common command for UNIX or Linux would be:

```
gzip -cd DBD-Ingres-0.51.tar.gz | tar xvf -
```

3. Open a command prompt or shell and change the directory to the location of the extracted files.

4. Ensure the Ingres environment is properly set up.

Ensure that `II_SYSTEM` and platform-specific paths have `LD_LIBRARY_PATH` defined.

5. Enter the following command to generate the Makefile needed to build the extension:

```
perl Makefile.PL
```

6. Build the extension:

```
make
```

## Test and Install the DBD::Ingres Extension

Perform the following procedure to test and install the DBD::Ingres extension.

### To test and install the extension

1. Set the environment variable `DBI_DSN` to the name of a valid Ingres database, for example, **perl****ldb**:

```
DBI_DSN=perlldb #ksh, bash, sh
setenv DBI_DSN=perlldb #tcsh, csh
```

2. Run the tests using the following command:

```
make test
```

If everything is set up correctly, the message "All tests successful" will be displayed.

3. Install the extension using the following command:

```
make install
```

## How You Can Use the DBD::Ingres Extension

To use the Ingres Perl module, the Perl DBI module must be included using the **use** statement. The data source name supplied to the DBI->connect() method is used to determine the correct DBD module to load to make the database connection.

### Example—Connect to Ingres and Select from a Table

The following is a simple segment of code that connects to the Ingres database iidbdb and selects from iitables.

```
use DBI;

# Define the database to use
$dbname = "dbi:Ingres:iidbdb";

# Connect to the database
my $dbh = DBI->connect($dbname, "", "");

# Prepare a statement
$cursor = $dbh->prepare("SELECT table_name, table_owner FROM iitables order by
table_name asc");

# Execute the cursor
$cursor->execute;

# Fetch the results
while ($row = $cursor->fetchrow_arrayref) {
    print(DBI::neat_list($row), "\n");
}

# Close the cursor
$cursor->finish;
```

# Appendix A: Frequently Asked Questions on Ingres for Linux

---

This section contains the following topics:

[Does Ingres Shut Down Automatically?](#) (see page 39)  
[What Character Set Should I Use on Japanese SUSE Linux 9?](#) (see page 39)  
[Which Linux Shell Should I Use?](#) (see page 40)  
[Why Is Createdb Issuing Error Messages?](#) (see page 40)  
[How Do I Get the Function Keys to Work?](#) (see page 40)  
[How Do I Map Function Keys?](#) (see page 41)  
[Should I Modify System Kernel Parameters?](#) (see page 41)  
[What Switches Do I Use to Compile C Applications?](#) (see page 42)  
[Do I Need to Change Permissions?](#) (see page 42)  
[Why Does the Fortran Compiler Fail When Using G77?](#) (see page 42)  
[How Do I Get Characters to Display Properly?](#) (see page 43)

## Does Ingres Shut Down Automatically?

**I notice that Ingres starts automatically when the machine is started. Does it also stop automatically when the machine is shut down?**

Yes. The ingstop command is issued when the machine is shut down.

## What Character Set Should I Use on Japanese SUSE Linux 9?

**I installed Ingres on Japanese SUSE Linux 9. Japanese SUSE's default encoding is set to ja\_JP.UTF-8, which is not an Ingres-supported character set. What character set should I use?**

Use kanjieuc for Ingres non-Unicode data types (for example char and varchar). If you want to use the Ingres Unicode data types, you must convert your UTF-8 data to UTF-16 format for storage in Ingres (regardless of the Ingres character set setting). To convert to UTF-16 format, you can use, for example, iconv(3).

## Which Linux Shell Should I Use?

**Linux provides multiple sh-type shells. Which should I use with Ingres?**

On Linux systems, the file `/bin/sh` is a link to a shell such as `bash`, `ash`, `ksh`, or `zsh`. This shell is invoked when a Bourne shell script is run. Ingres was developed and tested on a Linux system using GNU `bash`, version 2.05b-50. Limited, successful testing has also been done with the `ksh` and `zsh` shells.

## Why Is Createdb Issuing Error Messages?

**I am having trouble creating databases with the `createdb` program. The program is issuing error messages. Why?**

Make sure that you are not running the “`createdb`” program that is provided by PostgreSQL. Make sure that the `PATH` setting for the shell from which you install and start Ingres includes Ingres executable directories before other executable directories.

## How Do I Get the Function Keys to Work?

**How do I get the function keys to work with the forms-based Ingres applications on Linux?**

The `TERM_INGRES` environmental variable defines the type of terminal you are using, so that the function keys will work when using the forms-based tools, such as `CBF` or `QBF`. On Linux, setting `TERM_INGRES` to `konsole1` will work for most terminals.

During installation, the value of `TERM_INGRES` is set to `konsole1`. You can change this value if it does not meet your needs.



## How Do I Map Function Keys?

### How can I map function keys PFK1 through PFK4 for an xterm?

Running Ingres in an xterm, set TERM to xterm and set TERM\_INGRES to vt100fx. Then use xmodmap to determine and set your function keys. To show current settings, use the command `xmodmap -pke`. (To determine the syntax for your version of xmodmap, use `man xmodmap`.)

You will probably find that there are no bindings for KP\_F1 through KP\_F4; you will need to bind them.

For example: to bind keys Shift+F1 through Shift+F4, create a file "mykeys" that contains:

```
keycode 67 = F1 KP_F1
```

```
keycode 68 = F2 KP_F2
```

```
keycode 69 = F3 KP_F3
```

```
keycode 70 = F4 KP_F4
```

Then issue the command:

```
xmodmap mykeys
```

Shift+F1 through Shift+F4 will now be defined as PFK1 through PFK4.

## Should I Modify System Kernel Parameters?

### Do I need to modify system kernel parameters before running Ingres?

No. Standard kernels and kernels compiled with default values (without modifying the Linux source headers) should provide adequate resources. For additional information, see the Readme file.

You might, however, need to increase the maximum allowable size for shared memory segments. You can do this by running `/sbin/sysctl -w kernel.shmmax=<new value>` as root. Change is immediate and does not require a reboot.

Ingres will fail to start if any of the kernel parameters do not meet required values. You can check these parameters using the syscheck utility in `$II_SYSTEM/ingres/utility`, as follows:

```
syscheck
```

If syscheck reports any potential problems, use the following to generate a list of suitable parameters:

```
syscheck -c
```

The output is in the format expected by /sbin/sysctl. If the output is written to a file, the new values can be applied as follows:

```
syscheck -c > out.file  
/sbin/sysctl -p outfile
```

## What Switches Do I Use to Compile C Applications?

**What compiler and compiler switches were used to create the Linux version of Ingres? Are additional switches needed for compiling C-language application programs?**

Ingres was compiled using the GCC compiler version 3.3.3 with the following switches:

```
-fsigned-char -fno-strength-reduce -D_REENTRANT -DLINUX -  
D_GNU_SOURCE -DXLIB_ILLEGAL_ACCESS -D_FILE_OFFSET_BITS=64 -  
D_LARGEFILE_SOURCE -fPIC -O
```

## Do I Need to Change Permissions?

**Do I need to change the permissions for /dev/kmem to run Ingres on Linux?**

No. While this step is required on some UNIX systems, it is not required for this version of Ingres for Linux.

## Why Does the Fortran Compiler Fail When Using G77?

**When I compile the Fortran code generated by the ESQLF pre-compiler using g77, it fails with 'Unrecognized statement name.....'. Why?**

The g77 compiler (which is bundled with many, if not all, Linux distributions) does not support some of the statements that the ESQLF pre-compiler generates. More information is available at [gnu.org](http://gnu.org).

## How Do I Get Characters to Display Properly?

**When using alternative Ingres character sets, must I do anything other than setting `II_CHARSETXX`, to get the characters to display correctly?**

If your terminal is using the same character set as Ingres, you should not have a problem. If characters are not displaying correctly, try starting the terminal with a specific character set.

For example, if you are using SHIFTJIS (Japanese Double Byte character set) as the `II_CHARSETXX` setting in a `kterm`, start the `kterm` with the following command:

```
kterm -km sjis
```

If you encounter problems using double byte character sets and `kterm`, try using the `rxvt` terminal instead. For example, to start the `rxvt` terminal with the SHIFTJIS character set, issue this command:

```
rxvt -km sjis
```



# Index

---

## A

access  
    the instance • 7  
automatic startup • 39

## C

character set • 39, 43  
connect to Ingres • 15, 27, 29, 35  
COPY statement • 12  
CREATE TABLE statement • 9, 11  
createdb command • 9, 10, 40

## D

databases  
    creating • 9, 10  
demonstration application • 13

## E

Eclipse • 14, 15

## F

function keys • 40, 41

## J

Japanese SUSE Linux • 39

## K

kernel • 41

## L

Linux  
    compiler switches • 42  
    installing Ingres on • 5  
    shell • 40

## P

Perl • 35  
PHP • 29  
Python • 27

## R

RPM • 5, 6

## S

start Ingres • 8

## T

table  
    create • 11  
    populate • 12