Ingres® 2006 for Linux

Getting Started

Updated March 15, 2006

INGRES®

Contents

Chapter 1: Introducing Ingres for Linux 7

Ingres Features 7
What You Need to Know 9
RPM Packaging 9
Documentation Set 9

Chapter 2: Preparing for Installation 11

How You Prepare for Installation 11
An Ingres Installation 11
    Installation Code 12
Major Configurations 12
Ingres Files 14
File Location Guidelines 15
Installer-defined Directory 17
DBMS Server Disk Configurations 17
    Four-Disk DBMS Server Configuration 18
    Three-Disk DBMS Server Configuration 18
    Two-Disk DBMS Server Configuration 20
    One-Disk DBMS Server Configuration 21
Complete the Installation Worksheet 22
Installation Parameters 22
    General Installation Parameters 22
    DBMS Server Installation Parameters 23
World Region and Time Zone 24
    Time Zone Names 24
Supported Character Sets 26
Ingres Servers 28

Chapter 3: Installing Ingres for Linux 29

Automatic Startup 29
Installation Methods 29
Default Installation Method 30
    Install Ingres with the Default Configuration 30
ingres_express_install Command—Install Ingres 31
Invoke RPM Directly 31
    Install Into a Different Location 32

Documentation RPM Package 32
Custom Installation Method 32
How You Install Ingres as a Custom Configuration 32
Response File—Define Configuration for Custom Install 33
Set the II_RESPONSE_FILE Variable 34
Multiple Instances 35
Rebuild RPM Package with Unique Name 35
How You Upgrade Using RPM 36
Ingres 2006 Patch Removal 37
How You Access the Instance 37
Source the Environment File 38
Stop the Instance 38
Start the Instance 39
Further Customization Tasks 39
How You Prepare Your Installation for General Use 40
How You Safely Uninstall Ingres 41
uninstall_ingres Command—Uninstall Ingres for Linux 41

Chapter 4: Configuring Ingres Cluster Solution for Linux 43

Requirements for the Ingres Cluster Solution 43
Ingres Cluster Solution Overview 44
Preparation for Installing the Ingres Cluster Solution 45
How You Install the Ingres Cluster Solution 46
How Client Applications Access an Ingres Cluster 47
Class Node Affinity (CNA) 48
Global File System and Database Creation 48

Chapter 5: Understanding the Interfaces 49

Ingres ODBC Driver 49
Ingres JDBC Driver 50
Ingres .NET Data Provider 50
Ingres OpenAPI 51
Embedded SQL 51
Ingres Web Deployment Option 52
Application Development Tools 52

Chapter 6: Frequently Asked Questions 53

Appendix A: Linux Installation Worksheet 57

Installation Worksheet 57

Appendix B: RPM Packages 59

RPM Package Names 59
RPM Package Changes Since Last Release 60

Index 61

Chapter 1: Introducing Ingres for Linux

Welcome to Ingres®, the complete open source information management solution! Ingres lets you develop and deploy robust, mission-critical applications in a variety of environments.

This guide provides the information you need to get started quickly and productively with Ingres for Linux.

Ingres Features

Ingres is a complete relational solution, providing the following features:

- A robust, high-performance relational database management system (RDBMS) on which you can build the best in mission-critical applications.
- Support for the latest technology, such as multiple processors and 64-bit architecture, and standards such as XML and Unicode. The extensive list of supported platforms includes Linux for zSeries and S/390, UNIX, and Windows.
- Easy to manage environment that requires far less maintenance than the competition. Ingres is hard to beat in the area of long-term cost of ownership.
- The Ingres Management Architecture (IMA) tools provide the framework that allows you to create system management applications using standard SQL tools and access multiple installations across Ingres Net.
- Ingres Web Deployment Option provides users with a secure and reliable foundation for Internet-based electronic commerce. Applications developed with Ingres Web Deployment Option can be used to provide full read/write access to enterprise-wide corporate data. Database-driven web sites provide much greater flexibility and can reflect changes in dynamic content as they happen.
- Open connectivity lets you integrate the entire enterprise—from PCs to mainframes. Ingres Net, coupled with the Ingres ODBC and JDBC drivers, let you take advantage of the technologies that drive the business environment.

  Through ODBC, Ingres allows transparent access to enterprise data repositories. Applications developed through ODBC are independent of hardware, networking protocols, and operating systems. The Ingres ODBC driver uses the native API (application programming interface), effectively making ODBC a native language for Ingres.

If you prefer, you can use the OpenAPI to provide access to all distributed databases and take advantage of an alternative to embedded SQL.

- Integration of data from multiple sources on different platforms from a single application.

  The Enterprise Access products provide a single, standard interface for both relational and pre-relational databases. Real-time access is provided for a variety of client applications running on Windows, OpenVMS, UNIX, and Linux workstations, as well as through web browsers. Popular application development tools can utilize this technology, meaning that your options are unlimited.

  Databases supported include all popular UNIX or Microsoft Windows-based database management systems such as Oracle, Informix, Sybase, and Microsoft SQL Server. On the OS/390 platform, a separate product called EDBC gives the same level of access to Advantage CA-IDMS Database, Advantage Datacom Database, DB2, VSAM, CICS/VSAM, and IMS — so you can truly take your data from anywhere.

- Ingres Distribution Option supports distributed databases across a wide variety of hardware, software, and networking architectures. Regardless of whether the data resides on desktops or mainframes, Ingres Distributed Option lets you treat all your enterprise data as a single, global Ingres database.

- Ingres Replicator Option provides fault-tolerant data replication and guaranteed data integrity. Two-way replication allows your business to carry on when the network link is down. This process holds the data and completes the replication as soon as the network becomes available. Data is readily available to users at anytime or anywhere, and can be administered using the graphical user interface of Visual DBA.

- The Ingres character-based tools including Embedded SQL and Vision are available for more traditional environments. Embedded SQL programming kits are available for C, COBOL, Fortran, and several other 3GLs.

- Additional platform-specific, TP monitoring, and cluster support options, provide the integrity and reliability you need in a distributed environment by ensuring that transactions do not get lost or damaged.

- The Ingres Cluster Solution on Linux provides support for Linux Cluster environments. Ingres exploits open source and third party software to provide extensions to the base operating system. This software includes cluster file systems such as Oracle Cluster File System (OCFS) and the Red Hat Global File System (GFS), and the OpenDLM Distributed Lock Manager (DLM) based on the AIX DLM released by IBM to the open source community. Linux Cluster support lets you achieve scalability and reliability on a low-cost clustering platform.

What You Need to Know

This guide is intended for all Ingres users, including the system administrator, the database administrator, application developers, and users. Some chapters assume you are already familiar with:

- Basic programming concepts
- Basic relational DBMS concepts

Note: Some procedures in this guide require operating system privileges (or permissions). If you do not have these privileges and the operating system expertise associated with them, have the operating system administrator perform these tasks.

RPM Packaging

Ingres for Linux is distributed as a collection of Red Hat Package Manager (RPM) packages. For more information on RPM, see rpm.org, your UNIX man page, or your Linux info program.

Documentation Set

The documentation package for Ingres for Linux installs into a specific directory (usr/share/doc) and can be easily accessed from the desktop. A link to the documentation is created in the $II_SYSTEM/ingres/doc directory for all Ingres for Linux 2006 installations.

Chapter 2: Preparing for Installation

This chapter describes general installation considerations. You must thoroughly understand the information in this chapter and make the required decisions before you install Ingres.

How You Prepare for Installation

Make sure you thoroughly understand the issues regarding system resources, storage locations, and other configuration parameters before installing Ingres. If you are not sure, have someone more knowledgeable in these areas perform the installation planning.

To prepare your system for installation, follow this process:

- Verify system requirements. For requirements, see the Readme file.
- Determine your installation configuration and the RPM packages to be installed.
- Choose file locations.
- Complete the Installation Worksheet, which is provided as an appendix in this guide.
- Choose the default or custom installation method.

An Ingres Installation

An Ingres installation consists of a set of product components that share a unique system-file location, ownership, and installation code.

You can configure an installation in many ways, depending on the products you install and the parameters you set. For example, you can configure an installation as a stand-alone system or as a client that accesses databases on a remote DBMS Server installation. A network can contain many different types of installations.

Installation Code

The installation code is a two-character code that identifies a specific installation on a node and allows all processes and images to be installed and shared successfully. The value of the installation code is stored in the II_INSTALLATION environment variable.

The first character must be a letter; the second character can be a letter or numeral. The default installation code is II.

If you have more than one installation on the same node, each installation on that node must have a unique installation code. For example, you can install and run a new version of Ingres under one installation code, while maintaining an existing older installation under a different installation code on the same computer or network node.

Major Configurations

The configuration is determined by the RPM packages you install. RPM package names for this release are listed in RPM Package Names (see page 59).

Note: The base package, ingres2006, is a prerequisite for all other packages and must be installed prior to, or at the same time as, the other packages.

After performing the initial installation, you can install other packages later to add more capabilities to your installation. For example, you can add networking capabilities to a stand-alone installation that is connected to a network. Alternatively, you can add client capabilities to a networked DBMS Server installation.

The major configuration types and their corresponding RPM packages are as follows:

Stand-alone DBMS Server

Provides only local access to local databases. This installation includes the Name Server, DBMS Server, and its own set of tools.

Packages to install:

ingres2006

ingres2006-dbms

Networked DBMS Server

Allows remote clients to access its databases through a network. (If tools are installed, local users also can access its databases.)

This installation includes the Name Server, DBMS Server, Communications Server, Data Access Server, ODBC driver, and JDBC driver.

By installing Ingres tools and then modifying the Ingres Net connection data, you can add outgoing client capabilities to a networked DBMS Server installation. Doing so enables it to act both as a client of a remote DBMS Server and as a DBMS Server to its own remote clients.

Packages to install:

ingres2006

ingres2006-dbms

ingres2006-net

ingres2006-jdbc

ingres2006-odbc

Networked DBMS Server with Ingres Distributed Option

Allows access to multiple databases—local and remote, Ingres and non-Ingres—simultaneously. This installation includes Ingres Distributed Option, Name Server, DBMS Server, Communications Server, Data Access Server, ODBC driver, and JDBC driver. It may also contain Ingres tools.

Packages to install:

ingres2006

ingres2006-dbms

ingres2006-net

ingres2006-jdbc

ingres2006-star

Client

Has its own set of Ingres tools and accesses databases on a networked DBMS Server installation on a remote node. This installation includes the Name Server, Communications Server, Data Access Server, ODBC driver, JDBC driver, and Ingres tools. It does not contain the DBMS Server.

You can configure a networked DBMS Server installation as a client of another DBMS Server on a remote node. To do so, install Ingres tools on the DBMS Server, and then add client capabilities by modifying the Ingres Net connection data.

You can set up a client in a different environment to access an Ingres database in your current environment. For example, a client in a Windows environment can access a database in a Linux environment. For more information, see the *Connectivity Guide*.

Packages to install:

ingres2006

ingres2006-net

ingres2006-jdbc

ingres2006-odbc

In addition, you can add the following components and development tools:

- Applications-By-Forms (ingres2006-abf)
- Ingres Web Deployment Option (ingres2006-ice)
- Ingres Replicator Option (ingres2006-rep)

Ingres Files

An Ingres installation includes the following files:

- System files (executables)
- Database files

  These files include the master database and user databases. The master database, iidbdb, stores information about all databases, their locations, and the users who can access them.

- Transaction log file

  This file stores uncommitted transactions and buffers committed transactions before they are written to the database. Ingres uses one logical installation-wide log file. The file is circular and wraps when it encounters the physical end-of-file.

  Each logical log file may consist of up to 16 physical disk files, which helps to alleviate I/O bottlenecks. To ensure that no committed transactions are lost if the primary transaction log devices fail, Ingres can maintain a backup of the primary transaction log file on the storage locations you specify.

- Checkpoint files, journal files, dump files

  Checkpoint, journal, and dump files provide for data recovery in case of a database disk failure. Checkpoints alone provide for data recovery up to the time of the checkpoint. Checkpoints and journals provide for recovery up to the time of failure. Checkpoint, journal, and dump files provide online checkpoints.

- Temporary work files

  Work files are temporary files created during external sorts and other DBMS Server operations that require large amounts of temporary file space.

File Location Guidelines

Before installing Ingres, you must decide on locations for the Ingres files. Choose these locations carefully because they cannot be easily changed once specified (except for the location of the transaction log).

During installation, each location is stored in an Ingres environment variable/logical (for example, the location for the system files is stored in II_SYSTEM).

Use the following guidelines when choosing file locations:

- System files (II_SYSTEM)

  The location you choose for Ingres system (executable) files will also contain the error log and configuration files. Choose a location that has adequate disk space.

- Database files (II_DATABASE)

  The location for your databases contains the master database (iidbdb), which stores information about all databases, their locations, and the users that can access them. By default, this location also contains all user databases, unless the database administrator specifies an alternate location for a database when creating it.

  When choosing this location, consider the following:

  – Place database files on a separate disk from checkpoint, journal, and dump files to maximize chances for data recovery.

  – Place database files on a separate disk from the transaction log files to improve system performance by optimizing disk I/O.

  – On systems with three or more disks, do not place the database files on the same disk as your operating system because it forces database scans to compete with system operations for I/O.

  - Do not place database files on a device that is likely to become full as databases are added. Full disks can become fragmented, causing slow disk retrievals and degraded system performance.

- Checkpoint, journal, and dump Files (II_CHECKPOINT, II_JOURNAL, II_DUMP)

  Checkpoint files, journal files, and dump files can reside on the same device because journals and dump files are useful in recovery only if the associated checkpoint is also available. By default, the install program places journal and dump files in the same location as the checkpoint files.

  Important! Do not place the checkpoint device on the same disk as the database files. Storing data and backups on the same device provides poor insurance against disk failure. On single-disk systems, we recommend checkpointing to magnetic tape. Checkpointing to disk provides little safety if the disk fails.

- Temporary work files (II_WORK)

  Work files can reside on the same device as the checkpoint files because work files are useful in recovery only if the associated checkpoint is also available. For highest performance, however, assign your temporary work files to a different physical device than your database, checkpoint, journal, or dump files.

  For more information about temporary files and sorting, see the *Database Administrator Guide*.

- Transaction and backup transaction logs

  Consider the following when choosing the locations for the transaction or backup transaction log file:

  - If possible, avoid installing the primary and backup transaction log files on I/O-bound disks, because this can increase data acquisition times of the Recovery Server and archiver process and slow down all users.

  - If possible, put the transaction log file on separate disks from your database, checkpoint, dump, and journal files so that you can recover your unsaved, committed transactions if any of these disks fail.

  Locations for these files are defined by the log_file (primary) and dual_log (backup) configuration parameters. You can add, modify, or delete locations for your primary and dual transaction log files using the Configuration-By-Forms utility.

Note: On Ingres for Linux, the backup transaction log is disabled by default. To enable the backup transaction log, set the environmental variable II_DUAL_LOG to the location of the backup log file (either in the response file or prior to installing RPM packages).

Warning! We strongly recommend that you do not use ReiserFS for database locations. Severe performance degradation was noticed when using this file system.

Note: The II_SYSTEM location should not be placed on an OCFS-1 CFS.

Installer-defined Directory

In the diagrams that follow, the installer-defined directory (“inst.-def. dir.”) indicates a location chosen by the installer. In the installer-defined directory, the install program creates an appropriate directory tree in which it stores the appropriate files.

When specifying a file location, use only the path name for the installer-defined directory at the top of the directory tree. Do not include the subdirectories that the install program creates in the installer-defined directory.

DBMS Server Disk Configurations

The following examples illustrate some typical disk configurations for a stand-alone or networked DBMS Server installation.

Tip: To avoid a single point of failure, you should set up at least a two-disk installation. Configurations with three or more disks can provide better performance and additional data recovery options. Although you can put all files on a single disk, you are strongly advised against doing so, because you could lose all of your data if the disk fails.

For any server configuration, you can set up an optional backup transaction log device on a separate disk from the primary transaction log. This setup enables recovery of unsaved, committed transactions if the primary transaction log device fails.

Four-Disk DBMS Server Configuration

A configuration with four or more disks provides the best performance and recovery options. For best performance, configure your system with the operating system on a separate disk and your Ingres files on three or more other disks, as follows:

Disk 1—Operating system files

Disk 2—Checkpoint, journal, and dump files

Disk 3—Product system files, and transaction log work files

Disk 4—Databases

If possible, put your backup transaction log on a separate disk from both your database and primary transaction log. Also, you can use more than one disk partition for your transaction log.

Three-Disk DBMS Server Configuration

The following three-disk configuration provides better performance and recovery than a two-disk system:

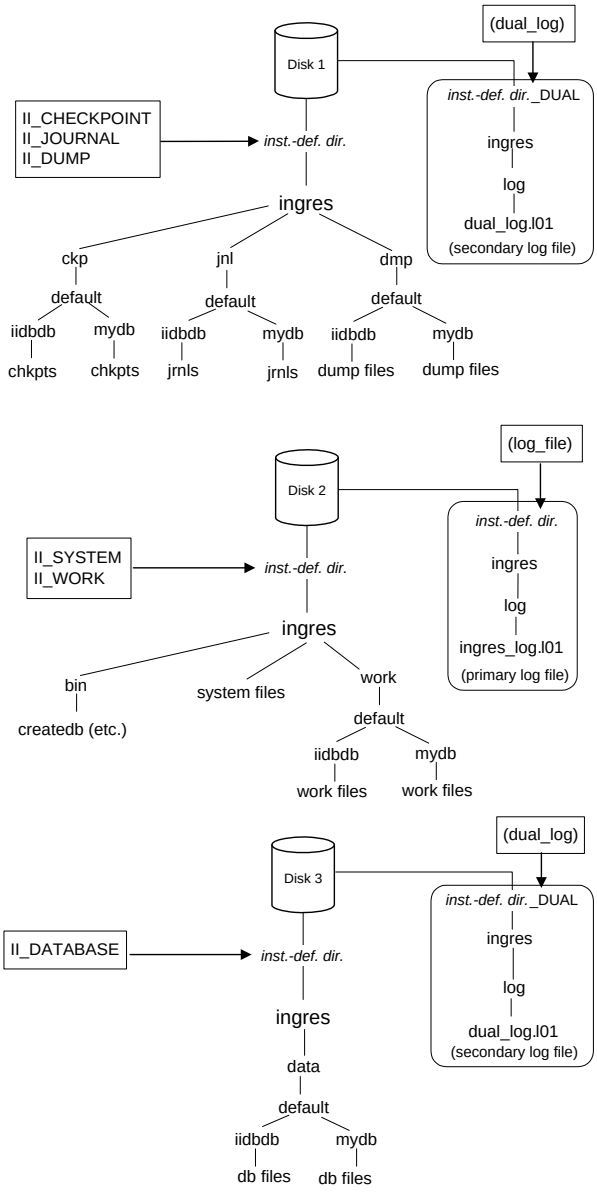
Disk 1—Operating system, checkpoint, journal, and dump files

Disk 2—Product system files, transaction log, and work files

Disk 3—Databases

Your backup transaction log can reside on either Disk 1 or Disk 3, as indicated by the oval surrounding its directory tree and the parentheses around the log_file and dual_log configuration parameters in the following diagram.

If Disk 3 fails, you can recover databases and committed transactions. If Disk 2 fails, you can recover your committed transactions if you have a backup transaction log; if you have no backup log, you will lose committed transactions that were not written to the database.



Two-Disk DBMS Server Configuration

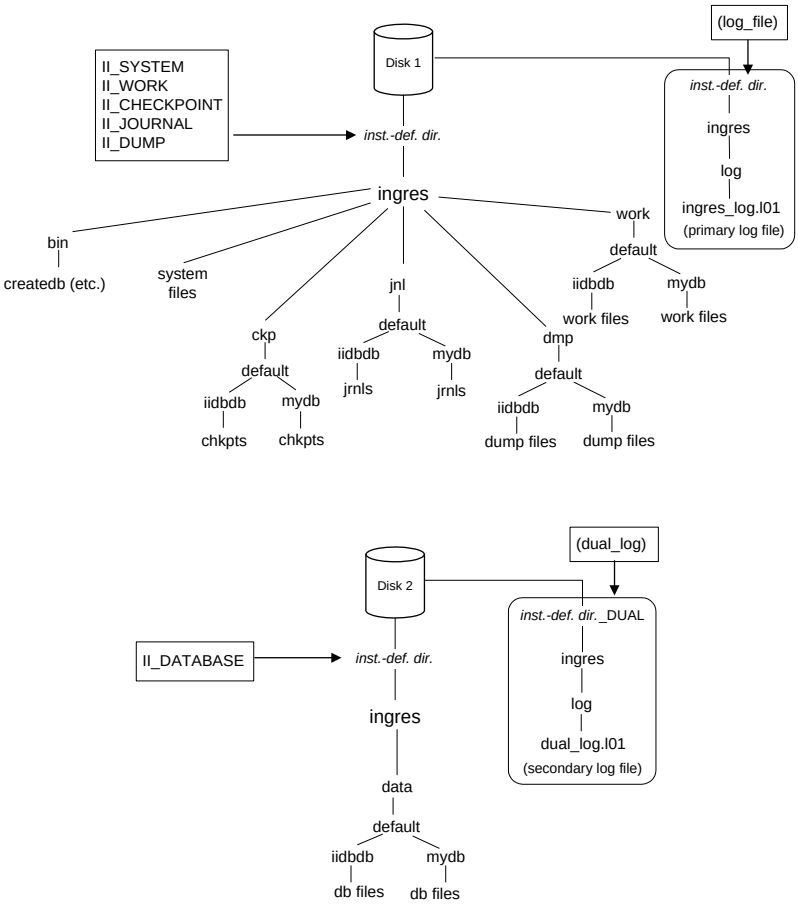
A two-disk system is the minimal recommended configuration:

Disk 1—All Ingres files except for those on Disk 2

Disk 2—Databases, optional backup transaction log file

If Disk 2 fails, you can recover databases and committed transactions. However, if Disk 1 fails and you do not have a secondary log device, you will lose unsaved, committed transactions.

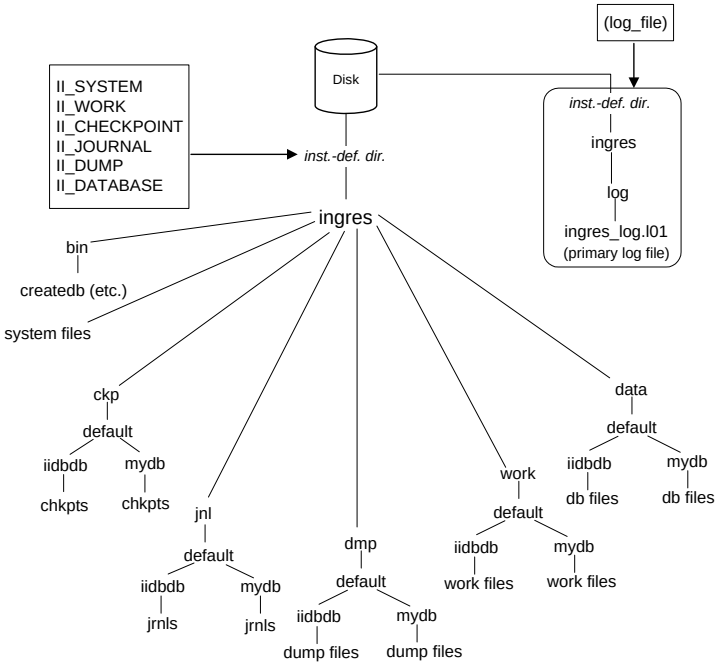
The following illustrates a sample two-disk DBMS Server configuration:



One-Disk DBMS Server Configuration

A single-disk system is a high-risk setup and not recommended. If the disk fails, you could lose all your data. On single-disk systems, you should checkpoint to magnetic tape.

The following illustrates a sample one-disk DBMS Server configuration:



Note: The default installation method installs Ingres as a one-disk system.

Complete the Installation Worksheet

Before installing Ingres, you should record file locations and values for other installation parameters on the installation worksheet (provided as an appendix in this guide). These entries will serve as ready responses to the installation program prompts.

To complete the worksheet

1. Make a copy of the blank Installation Worksheet for each installation you intend to create.
2. Fill in values for the required parameters.

   Note: In a cluster installation, a few of the worksheet values must be supplied for each node.
3. Keep the completed worksheet for future reference.

Installation Parameters

During the installation process, you must provide values for several installation and configuration parameters. Before installing Ingres, decide on these values and enter them on your Installation Worksheet.

General Installation Parameters

General installation parameters apply to all installations.

The general installation parameters are as follows:

Installation code (II_INSTALLATION)

Identifies the installation, as described in Installation Code (see page 12).

Ingres installation directory (II_SYSTEM)

Defines location of the product's system (executable) files.

User ID (II_USERID)

Defines the user ID of the system administrator that owns the Ingres installation. The ID is added to the system if it does not exist.

Group ID (II_GROUPID)

Defines the group ID to which the system administrator user ID belongs and that owns the Ingres installation. The ID is added to the system if it does not exist.

Silent install process (SILENT_INSTALL)

Quiets installation messages except those from RPM.

Start instance with OS (START_ON_BOOT)

Controls whether the instance will be started automatically when the operating system boots.

DBMS Server Installation Parameters

The parameter settings when installing the Ingres DBMS Server are as follows:

Database location (II_DATABASE)

Defines the location for the Ingres master database (iidbdb) and the default location for database files.

Checkpoint files location (II_CHECKPOINT)

Defines the location for the checkpoint files that serve as a static backup of the database.

Journal files location (II_JOURNAL)

Defines the location for the journal files, which provide a dynamic record of changes made to Ingres databases since the last checkpoint.

Dump files location (II_DUMP)

Defines the location for the dump files used to perform online backups.

Temporary work files location (II_WORK)

Defines the location for temporary files created during external sorts and other DBMS Server operations.

Default transaction log file size (LOG_KBYTES)

Defines the default transaction log file size. The default size (32768 KB per file) is adequate for most installations. You should change the file size only if you have an existing application that requires a larger transaction log file.

World Region and Time Zone

You must specify the time zone for your installation. This value is stored in the II_TIMEZONE_NAME environment variable.

On some systems, the default value for II_TIMEZONE_NAME is NA-PACIFIC. If you are in a different time zone, you must change the value of II_TIMEZONE_NAME.

Time zone names are organized by world region. In some cases, the time zone name is a positive or negative offset from Greenwich Mean Time (for example, GMT2 or GMT-2). If you are unable to locate the correct time zone within one of the designated world regions, use the GMT-OFFSET world region and specify one of the GMT offsets as your time zone.

The time zone parameter tells Ingres what adjustments to make for Daylight Savings Time. If you must make other adjustments for special time changes imposed in your area (such as for energy conservation purposes), you can use the iizic time zone compiler provided in the distribution. For details on the iizic compiler, see the *Command Reference Guide*.

Time Zone Names

The world regions and their time zone names are as follows:

Africa
GMT
GMT1
GMT2
GMT3
GMT4

Asia
INDIA
JAPAN
KOREA
HONG-KONG
PAKISTAN
PRC
ROC
GMT5
GMT6
GMT7
GMT8
GMT9
GMT10
GMT11

Australia
AUSTRALIA-LHI
AUSTRALIA-NORTH
AUSTRALIA-WEST
AUSTRALIA-SOUTH
AUSTRALIA-TASMANIA
AUSTRALIA-QUEENSLAND
AUSTRALIA-VICTORIA
AUSTRALIA-NSW
AUSTRALIA-YANCO

Middle East
EGYPT
IRAN
ISRAEL
KUWAIT
SAUDI-ARABIA
GMT2
GMT3
GMT4

North America
NA-PACIFIC
NA-MOUNTAIN
NA-CENTRAL
NA-EASTERN
NA-ALASKA
CANADA-ATLANTIC
CANADA-NEWFOUNDLAND
CANADA-YUKON
MEXICO-GENERAL
MEXICO-BAJANORTE
MEXICO-BAJASUR

North-Atlantic
EUROPE-WESTERN
EUROPE-CENTRAL
EUROPE-EASTERN
IRELAND
MOSCOW
POLAND
TURKEY
UNITED-KINGDOM
GMT
GMT1
GMT2
GMT3

South America
BRAZIL-EAST
BRAZIL-WEST
BRAZIL-ACRE
BRAZIL-DENORONHA
CHILE-CONTINENTAL
CHILE-EASTER-ISLAND
GMT6
GMT5
GMT4
GMT3

South Pacific
NEW-ZEALAND
US-HAWAII
GMT10
GMT11
GMT12
GMT-12
GMT-11
GMT-10

Southeast Asia
INDONESIA-WEST
INDONESIA-CENTRAL
INDONESIA-EAST
MALAYSIA
PHILIPPINES
SINGAPORE
THAILAND
VIETNAM
GMT7
GMT8
GMT9

GMT-Offset
GMT-12
GMT-11
GMT-10
GMT-9
GMT-9-and-half
GMT-8
GMT-7
GMT-6
GMT-5
GMT-5-and-half
GMT-4
GMT-3
GMT-3-and-half
GMT-2
GMT-2-and-half
GMT-1
GMT
GMT1
GMT2
GMT3
GMT3-and-half
GMT4
GMT5
GMT5-and-half
GMT6
GMT7
GMT8
GMT9
GMT9
GMT10
GMT10
GMT11
GMT12
GMT13

Supported Character Sets

During installation, select the character set carefully because changing this value later can corrupt the data in your databases. The installation program provides a default value.

Ingres-supported character sets for the non-Unicode character data types are as follows:

Character Set	Description	Format
ALT	Support of Cyrillic on DOS	Single byte
ARABIC	Arabic-449-Plus	Single byte
CHINESET	Traditional Chinese - Taiwan	Double byte
CHINESES	Simplified Chinese - PRC	Double byte
CHTBIG5	Traditional Chinese - Taiwan, BIG5	Double byte
CHTEUC	Traditional Chinese - Taiwan, EUC	Double byte
CHTHP	Traditional Chinese - Taiwan, HP ROC15	Double byte
CSGB2312	Simplified Chinese - GB2312	Double byte
CSGBK	Simplified Chinese - GBK	Double byte
CW	Cyrillic on Windows 3.1	Single byte
DECMULTI	DEC Multinational (superset of ASCII) and default for VMS	Single byte
DOSASMO	IBM DOS ASMO Arabic (cp708)	Single byte
ELOT437	Greek for PC/RS6000/SCO-UNIX	Single byte
GREEK	DEC Greek Elot	Single byte
HEBREW	DEC Hebrew	Single byte
HPROMAN8	HP Roman8 (superset of ASCII)	Single byte
IBMPC437	IBM PC Code Page 437 (US and English)	Single byte
IBMPC850	IBM PC Code Page 850 (Multilingual), includes accented characters	Single byte
IBMPC866	IBM PC 866 (Cyrillic for DOS)	Single byte
IS885915	ISO 8859/2 (Latin and some Greek). Identical to ISO 8859/1 Latin, except for eight characters, including the Euro currency symbol (€, Unicode U+20AC).	Single byte

Character Set	Description	Format
ISO88591	ISO 8859/1 Latin and default for UNIX (superset of ASCII)	Single byte
ISO88592	8859/5 (Latin and Cyrillic)	Single byte
ISO88595	8859/9 (Latin and some Turkish) CP 920	Single byte
ISO88599	ISO 8859/15 (Latin and Euro sign)	Single byte
KANJIEUC	Japanese, EUC	Double byte
KOI18	KOI 8-bit (ISO 6937/8), Russia	Single byte
KOREAN	Korean	Double byte
PC857	IBM PC Code page 857 - Turkish	Single byte
PCHEBREW	IBM PC / MSDOS Hebrew	Single byte
SHIFTJIS	Shift-JIS Japanese	Double byte
SLAV852	IBM PC Code Page 852 (Slavic)	Single byte
THAI	DEC Thai Tis	Single byte
WARABIC	Arabic	Single byte
WHEBREW	Microsoft Windows Hebrew	Single byte
WIN1250	Eastern Europe: Windows page 1250	Single byte
WIN1252	Windows code page 1252 - Latin 1 (Western Europe) and default for Windows	Single byte
WTHAI	IBM/Windows Thai (cp874)	Single byte

Note: The character set value must be entered in uppercase.

Note: ja_JP.UTF-8 on Japanese SUSE Linux is not supported.

Ingres Servers

The following server types are available, depending on your particular installation:

Bridge Server

Enables network communications between an Ingres client using one network protocol and an Ingres server using another network protocol.

Communications Server

Provides network communications that let users connect to databases belonging to other installations through Ingres Net.

DBMS Server

Allows users access to your installation's databases.

Data Access Server

Translates requests from the Ingres JDBC Driver and forwards them to the appropriate DBMS Server.

ICE Server

Allows developers to build World Wide Web applications that can access enterprise-wide corporate data, through the Ingres Web Deployment Option.

JDBC Server

Acts in tandem with the JDBC driver to provide JDBC access to Ingres.

Name Server

Keeps track of all the servers associated with an installation.

Recovery Server

Handles online recovery from server and system failures.

Remote Command Server

Allows the execution of remote operating system commands.

Star Server

Allows users to connect to multiple databases simultaneously through Ingres Distributed Option.

Chapter 3: Installing Ingres for Linux

This chapter explains how to prepare for installing Ingres on Linux, and how to install product components. It then discusses post-installation tasks.

Automatic Startup

By default, Ingres will be installed to start automatically at Linux system boot. During installation, the mkrc utility is run, which installs the appropriate RC script. To prevent Ingres from being added to the RC startup scripts, you can set the environment variable START_ON_BOOT=NO before installation.

For more information on the mkrc utility, see the *Command Reference Guide*.

Installation Methods

To install Ingres on Linux, you must choose one of following methods:

- **Default**—The default installation method uses default configuration parameter settings, and installs Ingres on one disk.
- **Custom**—The custom installation method lets you customize the configuration parameters by creating a response file.

Default Installation Method

The default installation method installs the product on one disk using the default configuration settings.

The installation process transfers all pertinent files from the distribution medium to the II_SYSTEM location for your Ingres instance on this node.

By default, the II_INSTALLATION variable is set to II, and all other variables default to the value of II_SYSTEM.

The value of II_SYSTEM is /opt/Ingres/IngresII, where II is the value of II_INSTALLATION. The default installation method installs all specified Ingres packages into this directory.

In addition to creating a system directory, the program creates an operating system user and a system group, both with the default name “ingres,” if needed.

The value for II_TIMEZONE_NAME is NA-PACIFIC until you change it after installation using the ingsetenv command.

Note: Before running the install program, make a complete backup of your system.

Install Ingres with the Default Configuration

To install all the RPM packages in the current working directory with the default configuration on one disk, issue this command:

```
ingres_express_install
```

ingres_express_install Command—Install Ingres

The ingres_express_install command quickly installs Ingres RPM packages.

If you set II_RESPONSE_FILE to point to a valid response file, then you can install Ingres with the configuration defined in that response file. The response file is described in Response File--Define Configuration for Custom Install (see page 33).

This command has the following format:

```
ingres_express_install [installation_ID]
```

installation ID

Specifies the Ingres installation ID to use for the installation.

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 installation ID of A1:

```
ingres_express_install A1
```

Invoke RPM Directly

As an alternative to using the express install script, you can install Ingres by invoking RPM directly.

To install a single package, invoke RPM with the appropriate installation flags, including the full path to the package you want to install, as follows:

```
rpm -Uvh full_path_to_file/Ingres_package.rpm
```

To install more than one package at a time, pass multiple file names, specifying the full path to each package, as follows:

```
rpm -Uvh full_path_to_file/Ingres_package.rpm
full_path_to_file/Ingres_package2.rpm full_path_to_file/Ingres_package3.rpm
```

To install all packages in the same directory, specify the following:

```
rpm -Uvh full_path_to_file/*.rpm
```

Install Into a Different Location

To install a package with an II_SYSTEM value other than the default, use the -prefix flag when invoking RPM. The following command installs the specified package with II_SYSTEM=/home/ingres/IngresII:

```
rpm –Uvh –-prefix=/home/ingres/IngresII path_to_file/Ingres_package.rpm
```

Note: All packages installed in a single installation must have the same value of II_SYSTEM. If you use the prefix flag to install the base package, you must install all subsequent packages with the same –prefix value.

Documentation RPM Package

The RPM package that contains the Ingres manuals is shared among all Ingres instances on one machine, and is installed into a standard location, /usr/share/doc. The Ingres documentation RPM package cannot be relocated. You cannot use the –prefix flag or the iirpmrename command with this package.

Custom Installation Method

The custom installation method lets you change the instance identifier, file locations, and other configuration parameters for the instance. You define the desired values in a response file, which is then used in the installation process.

Note: Before running the install program, make a complete backup of your system.

How You Install Ingres as a Custom Configuration

To install Ingres as a custom configuration, follow this process:

1. Create a response file that defines the configuration.
2. Specify the name of the response file on the II_RESPONSE_FILE environment variable.
3. Install the RPM packages by either using the express install script or invoking RPM directly.

Response File—Define Configuration for Custom Install

The response file defines the configuration for your Ingres installation. You can use any name for the response file.

Each entry in the response file must be on a separate line and have the following form:

variable=value

where *variable* is an Ingres environment variable.

For a list of valid variables, see the Installation Worksheet.

Example: Response File to Install Ingres as a Two-Disk System

The following response file shows Ingres installed on two disks, as follows:

- Disk 1—System files, checkpoint, journal, work, and dump locations, and transaction log
- Disk 2—Databases and backup transaction log

The machine has one CPU, is located in Tokyo, and requires the KANJIEUC character set.

Note: Because the checkpoint, journal, work, dump, and transaction log locations all reside with the system files, you do not need to specify a location because the default location is II_SYSTEM.

The response file for this configuration is as follows:

```
II_DATABASE=/disk2
II_DUAL_LOG=/disk2
II_TIMEZONE_NAME=JAPAN
II_CHARSET=KANJIEUC
```

Example: Response File to Install Ingres as a Four-Disk System

The following response file shows Ingres installed on four disks, as follows:

- Disk 1—Checkpoint, journal, and dump locations
- Disk 2—System files, transaction log, work files
- Disk 3—Databases
- Disk 4—Backup transaction log

The machine has two CPUs, requires a 500 MB transaction log, and is located in New York. You want the database to comply with the ANSI/ISO Entry SQL-92 standard.

The response file for this configuration is as follows:

```
II_DATABASE=/disk3
II_CHECKPOINT=/disk1
II_JOURNAL=/disk1
II_DUMP=/disk1
II_WORK=/disk2
II_LOG_FILE=/disk2
II_DUAL_LOG=/disk4
LOG_KBYTES=500000
II_NUM_OF_PROCESSORS=2
II_TIMEZONE_NAME=NA-EASTERN
SQL92=ON
```

Set the II_RESPONSE_FILE Variable

To install a custom configuration, you must set the Ingres environment variable II_RESPONSE_FILE to the name of response file that contains the configuration settings. Use the absolute path and file name.

To set the II_RESPONSE_FILE variable, use the following command:

For bash:

```
export II_RESPONSE_FILE=<full path to file>/<response file name>
```

For tcsh:

```
setenv II_RESPONSE_FILE <full path to file>/<response file name>
```

Note: The response file must be in a directory that is globally readable or the install process will fail.

Multiple Instances

You can have more than one installation on the same computer. Each installation on that computer must have a unique installation code. For example, you can install and run Ingres 2006 and Ingres tools under one installation code, while maintaining an existing 2.6 installation under a different installation code.

Tip: Do not delete or alter any files that the install process places into the $II_SYSTEM/ingres directory and its subdirectories. Also, do not place any user files or directories into this directory and its subdirectories. Lastly, do not alter the permissions on this directory and its subdirectories or on any files in this directory structure. Doing so compromises the security and integrity of your databases.

RPM does not allow multiple instances of the same package on one machine. To allow multiple Ingres instances on one machine, you must have a unique set of package names for each instance you want installed.

To do this, you must rebuild each RPM package to include an installation ID that is unique to the machine. You can then install this package using the instructions described in Invoke RPM Directly (see page 31).

Rebuild RPM Package with Unique Name

To install multiple Ingres instances on one machine, you need a unique set of package names for each instance. Each package must be rebuilt with the new name.

To rebuild the package, use the iirpmrename command. Multiple RPMs can be specified for renaming with the iirpmrename command.

For example, to rebuild the base package named Ingres2006-9.0.4.rpm to use an II_INSTALLATION of XX, issue the following command:

```
iirpmrename [full_path_to_file/]Ingres2006-9.0.4.rpm XX
```

A new RPM package is created in the current working directory called Ingres-XX-9.0.4.rpm. The package will install with II_SYSTEM=/opt/Ingres/IngresXX and II_INSTALLATION=XX.

Note: While you can still override the value for II_INSTALLATION using the response file or other methods mentioned above, we do not recommend it.

How You Upgrade Using RPM

Follow these instructions to safely upgrade any Ingres release using RPM.

To upgrade Ingres RPMs

1. Make sure the Ingres instance you want to upgrade is completely shut down and that you have taken a backup, as described in the *Migration Guide*.
2. Issue the following command to see the list of Ingres RPM packages installed for the instance to be upgraded:

   ```
   rpm -q --whatrequires core_pkg_name
   ```

   where *core_pkg_name* is the core package name for the release, for example, ca-ingres (for Ingres r3) or ingres2006 (for Ingres 2006).

   Note: If you are upgrading an instance that was installed using RPMs that were renamed to embed an installation ID, then use "*core_pkg_name-XX*" instead of the core package name.

   Note: All packages must be upgraded at once. Any attempt to perform an upgrade on a single or smaller group of packages will fail.
3. Log in as root and CD to the directory containing the RPM to be used to perform the upgrade. Make sure variable II_CONFIG is not set.
4. Run rpm -ivh to install the new RPM packages over the existing installation.

   If the installation was originally installed using --prefix to change II_SYSTEM from the default, then this parameter must also be used for the upgrade.

   For default installation location:

   ```
   rpm -ivh --replacefile --replacepkgs \
     complete list of RPM files to be installed
   ```

   For any other installation location:

   ```
   rpm -ivh --replacefiles --replacepkgs --prefix <II_SYSTEM> \
     complete list of RPM files to be installed
   ```
5. Remove the RPM package information for the old installation, as follows:
   a. Find the correct packages:

      ```
      rpm -q --whatrequires core_pkg_name
      ```

   b. If the list returned is satisfactory, remove the packages:

      ```
      rpm -q | --whatrequires ca-ingres | xargs rpm -e --justdb
      ```

Notes:

- If you used -rpm -Uvh with --prefix to upgrade an installation, binary and other installed files (for example, iimerge) may be removed. Should this occur, simply re-run the same command using the additional flags --replacefiles and --replacepkgs, and all the missing files will be replaced. Database or configuration files will not be affected.
- For the special case where you are upgrading from a non-NPTL build to a NPTL build with the same version number, you must upgrade using the following command:

  ```
  rpm -Uvh --replacefiles --replacepkgs new-package-list
  ```

Ingres 2006 Patch Removal

The only supported method for removing patches installed against Ingres 2006 is to reinstall the RPM from the base release. Therefore, make sure you keep the RPM used for your most recent install so that a patch can be backed out, if necessary.

How You Access the Instance

When the installation is complete, the instance is running.

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

Source the Environment File

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 installation 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.

Stop the Instance

To stop the instance:

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

   ```
   % ingstop
   ```

   The installation is shut down.

Start the Instance

To start the instance:

1. Log on to your system through the Ingres system administrator account for the instance.
2. Shut down components of the instance that are running, if any, using the following command:

   ```
   % ingstop
   ```

   The installation is stopped.
3. Enter the following command:

   ```
   % ingstart
   ```

   The installation is started.

Further Customization Tasks

The configuration installed by RPM is suitable for moderate hardware (about 512 MB of memory) and a moderate number of users (about 32).

After Ingres is installed and started, you may need to change default values for some configuration parameters to suit your particular needs or enhance performance. You may need to change the size of the transaction log file or the number of concurrent users. Use the Configuration-By-Forms utility or the ingsetenv command.

Note: Some changes, such as to the logging system, should not be made while the DBMS Server is running. Changes made with Configuration-By-Forms while servers are running are not effective until you restart the servers.

For information about DBMS Server or Ingres Distributed Option configuration options, see the *System Administrator Guide* and the *Distributed Option User Guide*.

How You Prepare Your Installation for General Use

After Ingres is installed, you must complete additional tasks so that users can access and use the new installation.

You must complete all or some of these tasks:

- Enable recovery of your master database.

  To help recover your master database (iidbdb) if it becomes corrupted, make sure iidbdb has been checkpointed and that journaling has been enabled. This is the automatic default. For details, see the *Database Administrator Guide*.

- Authorize users to start and stop servers.

  If you plan to have users other than the Ingres system administrator start and stop the Ingres servers, you must edit the config.dat file. For details, see the *System Administrator Guide*.

- Allow access on systems using shadow passwords.

- Allow users to access tools and databases.

  To enable users to access tools and databases, the system administrator must do the following:

  – Edit the user login files to facilitate user access to the tools needed to query the databases.

  – Use the accessdb command or create user statement to authorize users to access specific databases.

  For more information, see the *System Administrator Guide*.

- Create or upgrade databases.

  For each new installation, someone must create the needed databases using the createdb command, as described in the *Database Administrator Guide*. Only the user databases must be created; the iidbdb master database is created during installation.

  If upgrading from an earlier release, you must run the upgradedb program to upgrade your databases after the installation is complete. For details, see the *Migration Guide*.

How You Safely Uninstall Ingres

Removing Ingres is an irreversible event with pervasive effects. Any products or applications that shared the removed DBMS Server are affected, as follows:

- Any future attempt to connect to this database will fail.
- If you re-install Ingres, you may not be able to reference the data files.

If you know you want to remove an Ingres installation, follow this process:

1. If you want to keep the data files for later use, you must run unloaddb against each database before removing Ingres. Doing so will allow you to reference the data files easily if you re-install Ingres later. For details on the unloaddb command, see the *Database Administrator Guide* and the *Command Reference Guide*.
2. If you do *not* want to keep the data files, you can run destroydb against each database before removing Ingres. The destroydb command locates and deletes all data files. For details on the destroydb command, see the *Command Reference Guide*.
3. Uninstall Ingres.

uninstall_ingres Command—Uninstall Ingres for Linux

The uninstall_Ingres command removes any or all instances of Ingres for Linux on a machine. This command invokes RPM. If this command is run without any parameters, it generates a list of standard (non-renamed) installed Ingres RPM packages and prompts for confirmation before removing them.

This command has the following format:

```
uninstall_ingres {[installation ID] | [-a (--all)]} [-y (--yes)] [-c (--clean)]
```

installation ID

Identifies the installation ID embedded in the renamed package names to be removed.

-a

Removes all Ingres RPM packages for *all* Ingres installations. This parameter cannot be used if specifying an installation ID.

-y

Answers yes to all prompts (that is, does not prompt).

-c

Removes the $II_SYSTEM/ingres directory after the uninstall operation is complete.

Chapter 4: Configuring Ingres Cluster Solution for Linux

This appendix describes requirements for the Ingres Cluster Solution on Linux and how to install and configure it. The Ingres Cluster Solution for Linux provides support for Linux Cluster environments.

Requirements for the Ingres Cluster Solution

The Ingres Cluster Solution requires that you install Ingres enabled for Native POSIX Threads. To install Ingres enabled for Native POSIX Threads, you must be running a Linux distribution that provides Native POSIX Thread Library (NPTL) support. All Linux distributions based on the 2.6 kernel provide NTPL support, as does the Red Hat Enterprise Linux AS 3.0 2.4.21EL kernel.

In addition, you must have the following:

- A hardware cluster in which multiple machines can share the same physical storage. This cluster may take the form of a Storage Area Network (SAN), shared 1394b/“Firewire” drives, or shared Small Computer Systems Interface (SCSI) drives.

  All disk locations must be set up as cluster file systems on storage accessible from the same path from all members of the cluster.
- A separate isolated private network for use by the Distributed Lock Manager and Quorum process.
- A supported cluster file system (CFS). This software allows simultaneous update access to the shared storage from multiple machines. For the latest information on supported CFS products, see http://ingres.com.
- The OpenDLM distributed lock manager, which can be obtained from the Ingres Corporation web site or from SourceForge at http://sourceforge.net/projects/opendlm/.
- The iiquorumd quorum management daemon, which can be obtained from the Ingres Corporation web site.

Ingres Cluster Solution Overview

The Ingres Cluster Solution is a variation of a typical Ingres installation in which Ingres runs simultaneously on multiple host machines to provide cooperative and transparent access to one or more databases.

Ingres is installed in the typical manner, except most file locations must be on cluster file systems—that is, file systems using hardware and software that allow safe simultaneous update access from all machines intended to be part of the cluster. Certain other locations must *not* be on cluster file systems if the file system supports only block-oriented data transfer (for example, Oracle Cluster File System). Once installed, you run the iimkcluster utility to convert the initial installation into one of the cluster members (nodes), and then run the iisunode utility to add more nodes.

Use of the Ingres Cluster Solution requires specialized hardware and third-party software.

The Ingres Cluster Solution is incompatible with the following Ingres features:

- Row-level locking, except when accessing a database through a server class using “class node affinity”
- Update mode locks, except when accessing a database through a server class using “class node affinity”
- Two-phase commit (2PC)
- Replication
- C2 auditing of tables with long data types

On each node, you should make sure that Ingres and your applications perform as expected. Some of the restrictions on lock level and lock mode are handled internally by Ingres, but may result in increased contention or deadlocks.

Preparation for Installing the Ingres Cluster Solution

Before installing Ingres in a Linux Cluster environment, follow these steps:

1. Check the Ingres Technical Support web site for the latest cluster installation procedures and supported hardware and software. Download the required software.
2. Install any required non-Ingres cluster support software, and perform the procedures recommended by the provider for verifying correct installation. This software includes:
   - A cluster file system
   - OpenDLM
   - iiquorumd
3. Plan your storage location layout as described for a stand-alone Ingres installation, with the restriction that your data (II_DATABASE), transaction log (II_LOG_FILE, II_DUAL_LOG), checkpoint (II_CHECKPOINT), journal (II_JOURNAL), and dump (II_DUMP) locations must all reside on a cluster file system.

How You Install the Ingres Cluster Solution

Follow these steps to install and configure Ingres in a Linux Cluster environment:

1. Install Ingres in a stand-alone configuration using the locations decided upon under Step 3 in Preparation for Installing the Ingres Cluster Solution (see page 45).
2. Verify that the stand-alone Ingres installation operates correctly.

   Typically, it is easier to resolve any configuration issues at this stage because only one machine is in use. In addition, aside from the cluster file system support, Ingres is using operating system software only.
3. When you are confident that the stand-alone Ingres is operating correctly, shut down the installation. As the user that owns the installation, execute the iimkcluster utility, as follows:

   ```
   iimkcluster
   ```

   The utility prompts you for a node number and a nickname.

   Node numbers are unique integers in the range 1 through the maximum supported cluster members for your platform (currently 15). During a partial cluster failure, the surviving cluster member (node) with the lowest node number is responsible for recovering transactions on behalf of the failed nodes, so you should assign low numbers to the more powerful machines in the cluster.

   The nickname is an optional simple alias, which you can use in any context in which you could specify a *nodename* parameter. The nickname appears in the error log in lieu of the machine name.

   The iimkcluster utility renames the transaction logs and certain diagnostic log files (iircp.log, iiacp.log, and so on) by appending the host name of the machine on which the cluster member is running. Also created is a sub-directory in the $II_SYSTEM/ingres/files/memory directory with the name of the host machine, as well as directory *$II_SYSTEM/ingres/admin/hostname* to which the *symbol.tbl* is relocated.

   This step keeps entities that are normally operated upon by only one node separate from corresponding objects that will be created by the other nodes.
4. Restart Ingres. Confirm that all processes have started. Confirm the initial node is operational by performing a few sanity checks such as creating and destroying a scratch database.

   You should also perform application testing to confirm that certain Ingres Cluster Solution restrictions, such as lack of support for row-level locking, will not impact the usability of your applications.
5. Shut down Ingres.

6. For each additional cluster member (node), set up your Ingres environment by sourcing the environment file. This file (.ingXXbash or .ingXXtsch) is located in II_SYSTEM.
7. Run the iisunode utility on each node. As the user that owns the installation, enter the following:

   ```
   iisunode
   ```

   The utility prompts you for a unique node number and nickname. Once entered and confirmed, iisunode does the following:

   - Adds the same directories for the new node as iimkcluster created for the initial node.
   - Duplicates the configuration information from the initial node.
   - Creates a private symbol.tbl file based on the initial nodes symbol.tbl file.
   - Creates the transaction logs for this node.
8. Start Ingres individually on each node, and verify correct operation.

How Client Applications Access an Ingres Cluster

Applications can access Ingres configured for Ingres Cluster Solution by using any of the following methods:

- Running directly on one of the cluster member machines
- Connecting directly to a specific cluster member using a vnode defined with the network address of the cluster member
- Selecting a connection to any available member of the cluster, or a subset thereof, by using a vnode defined with multiple network addresses

Class Node Affinity (CNA)

Ingres allows the creation of server classes that function as regular DBMS servers but can be configured for specialized situations. The server parameter class_node_affinity, if set for a server class, allows servers in this class to be started on only one node at a time.

The configuration name and server class name for the default CNA classes generated is iicnann, where nn is the node number, zero padded.

While iimkcluster and iisunode set up a separate CNA class for each node they are run on, these classes are not bound to the node they were defined on, but can be started on any node. In addition, any database (except iidbdb) that is connected to a server class using CNA cannot be opened by any other server class, including the default class.

The advantage of these restrictions is that because all operations on the database are known to be being performed on one node with pages resident in one cache, operations on the database do not require distributed locks, and the pages for the database do not need to participate in Distributed Multi-Cache Management (DMCM) protocols. For an installation servicing multiple databases, this allows you to increase efficiency by grouping your database operations by node, which significantly increases cache hit rates and decreases the latency of lock resolution and the overhead associated with DMCM. In addition, Row Level Locking and Update Mode Locks are automatically supported for databases serviced by CNA classes, instead of being silently converted to lower granularity locks and stronger lock modes.

Global File System and Database Creation

The Red Hat Global File System (GFS) has poor performance using direct I/O and small page sizes, as required when using 2 KB pages with Ingres. (This problem was seen on GFS version 6.)

When creating databases with II_DATABASE on a GFS device, we recommend that you use the createdb –page_size option to specify a page size for your system catalogs of at least 4096 (4 KB). All tables residing in whole or in part on a GFS device should use page sizes of 4096 or higher. If you have work locations on GFS devices or want to avoid the need to explicitly specify a page size when creating a table, set the default page size (default_page_size) to 4096 or higher.

If the transaction logs reside in whole or in part on a GFS device, block_size should be at least 4.

Chapter 5: Understanding the Interfaces

This chapter gives an overview of the tools that Ingres provides for interfacing with the database.

Ingres ODBC Driver

The Ingres ODBC driver is compliant with Microsoft Open Database Connectivity (ODBC) interface specifications. ODBC is a specification for an application programming interface (API) that enables applications to access multiple database management systems using the SQL language.

ODBC permits maximum interoperability—a single application can access many different database management systems. This enables an ODBC developer to develop, compile, and deploy an application without targeting a specific type of data source. Users can then add the database drivers that link the application to the database management systems of their choice.

The Ingres ODBC driver is installed during the installation process. It includes the Ingres ODBC Call-level Interface (CLI), which provides access to the ODBC application environment without the need to use a third-party driver manager. The Ingres ODBC driver is supported on all platforms on which Ingres runs.

For important information and a free download of the latest version of the Ingres ODBC driver, see the Ingres web site.

For more information on the Ingres ODBC driver, including supported functions and how to configure and connect to a data source, see the *Connectivity Guide*.

Ingres JDBC Driver

The Ingres JDBC product includes three components:

- Data Access Server (DAS)
- JDBC driver
- Information utility

The Ingres JDBC driver is a pure Java implementation of the JDBC 3.0 API released with the Sun Java SDK version 1.4. The driver supports application, applet, and servlet access to Ingres data sources through the Data Access Server.

The information utility, JdbcInfo, loads the Ingres JDBC driver and displays its internal release information.

For more information on the Ingres JDBC driver, see the *Connectivity Guide*.

Ingres .NET Data Provider

The Ingres .NET Data Provider is a .NET component that enables high-performance native .NET access to Ingres data sources, and delivers Ingres data to the Microsoft .NET Framework.

The Ingres .NET Data Provider offers:

- A series of .NET types to describe the user's data
- .NET provider classes to manipulate the data
- Connection pooling to efficiently manage data connections

Integration with Visual Studio .NET allows an application programmer to drag and drop the Ingres .NET Data Provider design component onto a design surface. Integration also allows the programmer to use wizards and editors to aid application development.

For more information on Ingres .NET Data Provider, see the *Connectivity Guide*.

Ingres OpenAPI

Ingres OpenAPI is a C programming language interface for accessing an Ingres database. It enables you to develop applications using a set of functions that are called directly with normal function-call facilities. This interface provides an alternative to embedded SQL, which requires a preprocessor in addition to a C compiler.

For more information, see the *OpenAPI User Guide*, which provides:

- Requirements for creating an Ingres OpenAPI application
- A description of the header files, library, and environment variables used by an Ingres OpenAPI application
- How to use the synchronous and asynchronous sample code included with the Ingres OpenAPI product

Embedded SQL

The term embedded SQL (ESQL) refers to SQL statements embedded in a host language such as C or Fortran. The ESQL statements include most interactive SQL statements, plus statements that fulfill the additional requirements of an embedded program.

All ESQL statements must be processed by the ESQL preprocessor, which converts the ESQL statements into host-language source-code statements. The resulting statements are calls to a runtime library that provides the interface to Ingres. After the program has been preprocessed, you must compile and link it according to the requirements of the host language.

For more information, including details about using ESQL with a particular host language and instructions on compiling and linking, see the *Embedded SQL Companion Guide*. You can also see the *SQL Reference Guide*, which describes ESQL independently of any host language.

Ingres Web Deployment Option

The Ingres Web Deployment Option provides the foundation for Internet-based electronic commerce. It enables a Web client to retrieve data from or update an Ingres database.

You can specify actions to perform in your Web application using special HTML variables defined by the Web Deployment Option. For example, there are variables to:

- Execute dynamic SQL statements.
- Run Ingres database procedures.
- Run Report-Writer reports.
- Run client applications.

A page generated by setting HTML variables can contain data from only a single SQL statement. When you need data from more than one statement, you can create Web Deployment Option Macro XML documents. A macro document contains one or more special tags that define an SQL statement to be executed. Macro XML allows you to create Web pages that include data from several database tables and provides many formatting options that control the way the data is presented to the Web client.

Additionally, the Web Deployment Option provides macro tag extensions for use with an XML-aware editor, so that adding XML elements to your documents is easier than ever. You can simply point and click to automatically generate macro syntax within your working environment. For more information, see the *Distributed Option User Guide*.

Application Development Tools

Ingres provides an extensive set of application development tools, which include:

- Vision
- Applications-By-Forms (ABF)
- Visual Forms Editor (VIFRED)
- Forms Runtime System (FRS)

For information on these tools, see the *Forms-based Application Development Tools User Guide*.

Chapter 6: Frequently Asked Questions

This chapter answers common questions about Ingres for Linux. It provides valuable information to further your knowledge about your new software.

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.

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).

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.

I am having trouble creating databases with the createdb program. The program is issuing strange 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.

Where are function keys PFK1 through PFK4 on my PC keyboard?

Although key bindings and terminal settings vary with equipment and with character sets, if you are running Ingres on a Linux console in vt100 mode, the following should work: Set TERM to vt100 and set TERM_INGRES to vt100f. The keys PFK1 through PFK4 should be bound to the top row of the numeric keypad.

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.

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 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 3.2.2 with the following switches:

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

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

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

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.

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
```

Appendix A: Linux Installation Worksheet

This appendix provides a worksheet to aid in installing Ingres on Linux.

Installation Worksheet

The worksheet shows the environment variables in parentheses. You can use these variable names in a response file.

If you do not want to use the default value shown in the Default column, enter a new value in the Value column.

Important! You cannot change the character set from its current setting (II_CHARSET) at any time without risking the corruption of your data.

After installing the product, keep the completed worksheet for future reference.

Worksheet for Installation_____ on Host_____

Required Parameter	Default	Value
General Installation Parameters		
Installation code (II_INSTALLATION)	II **Note:** Value is II only if you have not renamed the RPM package.	
Location for system files for Ingres and Ingres tools (II_SYSTEM)	II_SYSTEM	
User ID (II_USERID)	ingres	
Group ID (II_GROUPID)	ingres	
Silent install process (SILENT_INSTALL)	no	
Start instance with OS (START_ON_BOOT)	yes	
Language (II_LANGUAGE)	english	
Region and time zone (II_TIMEZONE_NAME)	NA-PACIFIC	

Required Parameter	Default	Value
DBMS Server Parameters		
Database location (II_DATABASE)	II_SYSTEM	
Checkpoint files location (II_CHECKPOINT)	II_SYSTEM	
Journal files location (II_JOURNAL)	II_CHECKPOINT	
Dump files location (II_DUMP)	II_CHECKPOINT	
Temporary work files location (II_WORK)	II_SYSTEM	
Default transaction log file size (LOG_KBYTES)	32768	
Transaction log file location (II_LOG_FILE)	II_SYSTEM	
Backup transaction log file location (II_DUAL_LOG)	II_LOG_FILE	
Number of concurrent users (CONNECT_LIMIT)	32	
Region and time zone (II_TIMEZONE_NAME)	NA-PACIFIC	
Character set (II_CHARSET)	ISO88591	
Comply with ANSI/ISO Entry SQL-92 database standards? (SQL92)	no	
Region and time zone for this client (II_TIMEZONE_NAME)	NA-PACIFIC	
Listen address for the DBMS server that will handle this client's requests	II_INSTALLATION	

Appendix B: RPM Packages

This appendix lists Ingres for Linux RPM packages available for installation.

RPM Package Names

The RPM packages you can install for Ingres for Linux 2006 are listed here.

Note: Version numbers change frequently, so they are not shown as part of the package name in this guide.

RPM Package	Description	Comments
ingres2006	Core package	Pre-requisite of all other packages
ingres2006-abf	ABF and Vision	
ingres2006-dbms	DBMS Server	
ingres2006-ice	Web Deployment Option	Requires DBMS
ingres2006-jdbc	JDBC Server	
ingres2006-net	Communications servers	
ingres2006-odbc	ODBC driver	
ingres2006-rep	Replicator Option	Requires DBMS and Net
ingres2006-star	Distributed DBMS	Requires DBMS and Net
ingres2006-32bit	64-bit Linux only	
ingres2006-documentation	Ingres manuals	Independent of other packages. Do not rename or relocate. If you install multiple instances of Ingres, this package is installed only once, because all instances of a given version can share the same manuals.

RPM Package Changes Since Last Release

Ingres 2006 has a reduced number of RPM packages compared with Ingres r3. Some components that were RPM packages are now merged into other packages, as follows:

Ingres Component	Now In This RPM Package
Embedded SQL precompilers	ingres2006
Query and Reporting runtime	ingres2006
C2 Security	ingres2006-dbms
Tuxedo support	ingres2006-dbms
Object Management Extensions	ingres2006-dbms
Bridge Server	ingres2006-net
DAS Server	ingres2006-net
Vision	ingres2006-abf

Index

A

access
    to database and tools • 40
    to the instance • 37
accessdb command • 40
application development • 50, 52
Applications-By-Forms • 52
automatic startup • 29, 53

B

Bridge Server • 27, 59

C

character set • 25, 26, 53, 55
character-based tools • 7
checkpoint file location • 21
class node affinity • 48
client
    applications and Ingres Cluster • 47
    installation • 12
Cluster Solution • 7, 43
Communications Server • 12, 27
compiling application programs • 55
configuration
    default • 30, 39
    of installation • 12, 39
    parameters • 39
    types • 11, 12
createdb • 53
custom installation method • 32
customization • 39

D

Data Access Server • 12, 27, 50
data recovery • 17, 21
database
    access • 40
    creating • 40
    creating on a Global File System • 48
    file locations • 18, 20, 22
    master • 22
DBMS Server
    disk configurations • 17
    file locations • 17
    installation parameters • 22
default installation method • 30
disk configurations • 17
Distributed Option • 7
documentation RPM package • 9, 32, 59
dump files • 18, 23

E

Embedded SQL • 7, 51
Enterprise Access products • 7
environment file • 37, 38
environment variables
    II_CHARSET • 33, 55, 57
    II_CHECKPOINT • 22, 34
    II_DATABASE • 22, 33, 34, 45, 48, 57
    II_DUAL_LOG • 16
    II_DUMP • 22, 34, 57
    II_GROUPID • 22, 57
    II_INSTALLATION • 11, 12, 22, 37
    II_LOG_FILE • 34
    II_NUM_OF_PROCESSORS • 34
    II_RESPONSE_FILE • 32, 34
    II_SYSTEM • 9, 22, 30, 32, 35, 45
    II_TIMEZONE_NAME • 23, 30, 33, 34
    II_USERID • 22, 57
    II_WORK • 23, 57
    LOG_KBYTES • 22, 34, 57

F

file locations • 17
Forms Runtime System • 52
Fortran compiler • 55
function keys • 53, 54

G

g77 compiler • 55
Global File System • 48

I

ICE Server • 27
II_CHARSET • 33, 55, 57
II_CHECKPOINT • 22, 34, 57
II_DATABASE • 22, 33, 34, 45, 48, 57
II_DUAL_LOG • 16
II_DUMP • 22, 34, 57

II_GROUPID • 22, 57
II_INSTALLATION • 11, 12, 22, 37
II_JOURNAL • 23, 34, 57
II_LOG_FILE • 34
II_NUM_OF_PROCESSORS • 34
II_RESPONSE_FILE • 32, 34
II_SYSTEM • 9, 30, 32, 35, 45
II_TIMEZONE_NAME • 23, 30, 33, 34
II_USERID • 22, 57
II_WORK • 23, 57
iimkcluster utility • 46
iirpmrename command • 35
iisunode utility • 46
installation
  code • 11, 12, 22
  location, different • 32
  preparing for • 11
  worksheet • 57
installing
  a custom configuration • 29, 32
  Ingres Cluster Solution • 46
  multiple instances • 35
  the default configuration • 29, 30
instance
  accessing • 37, 38
  multiple • 32, 35

J

Japanese SUSE • 26, 53
JDBC driver • 50
journal files • 23, 57

K

kernel • 43, 54

L

location • 18, 20
LOG_KBYTES • 22, 34, 57
logging system, making changes to • 39

M

multiple instances • 32, 35

N

Name Server • 12, 27
Net • 7

O

OCFS • 17
ODBC driver • 49
OpenAPI • 51

R

recovering
  databases • 18, 20
  transactions • 17, 21
Recovery Server • 27
ReiserFS • 17
Replicator Option • 7
response file • 33, 57
RPM • 9, 11
  package, documentation • 9, 32, 59
  packages • 9, 12, 59

S

server types • 27
shell • 53
Star Server • 27
starting and stopping Ingres • 38, 39
system files • 18, 34, 57

T

time zone • 23, 57
transaction log files
  backup • 17, 18, 20
  recovery • 17

U

uninstall Ingres • 41
user
  access to databases and tools • 40
  access to instance • 38
  authorization to start servers • 40

V

Vision • 7
Visual Forms Editor • 52

W

Web Deployment Option • 7
web-based applications • 52
work files • 23, 57
worksheet for installing • 57