

The Ingres logo, consisting of the word "INGRES" in white, uppercase, sans-serif font, centered within a dark blue rectangular box.

Ingres® Code Architecture

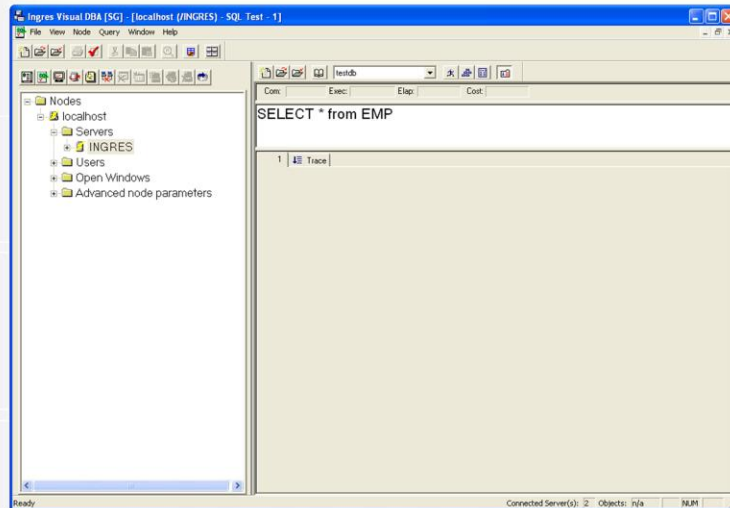
Shubhagam Gupta

Welcome to discussion on Ingres Code Architecture.

In this presentation we will discuss the

- Various components of Ingres database,
- Their interrelationship and
- Their code architecture.

Ingres Code Architecture



2

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

To an Ingres user, an Ingres database stores and retrieves data based on an SQL query.

For example here is a screen shot of Ingres Terminal monitor session connected to database testdb.

Here we wish to retrieve the details of all employees in a company.

We enter our SQL query <Click To Animate >

“Select * from emp”

The database server processes the query <Click to Animate>

Ingres Code Architecture

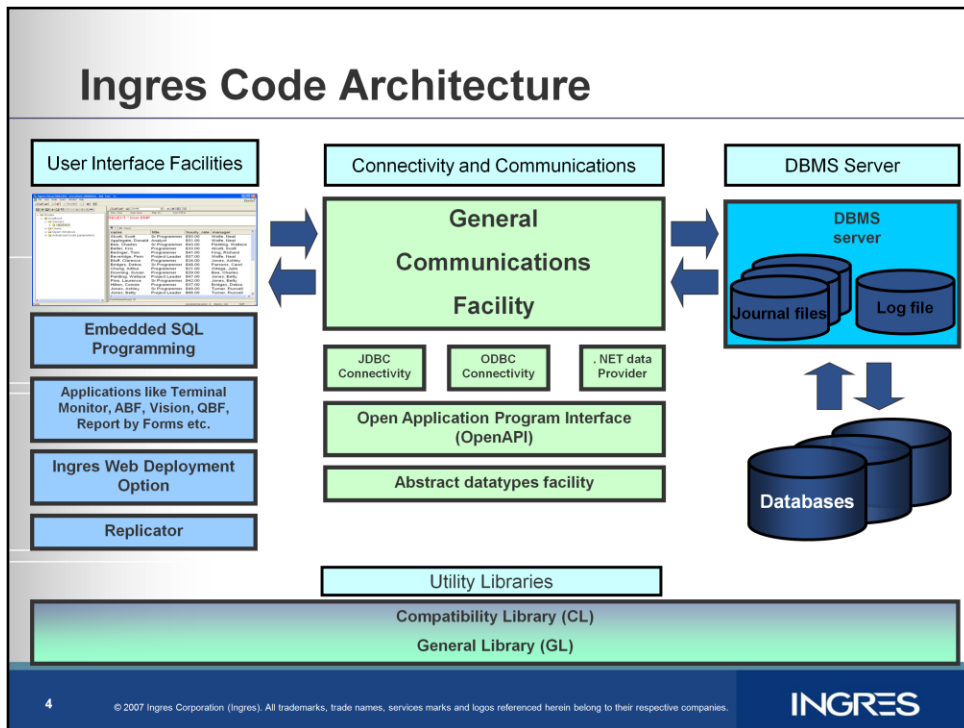
The screenshot displays the Ingres Visual DBA interface. The main window shows the results of a query: `SELECT * from EMP`. The results are presented in a table with four columns: `name`, `title`, `hourly_rate`, and `manager`. The table contains 15 rows of employee data. On the left, a tree view shows the database structure, including 'Users', 'Open Windows', and 'Advanced node parameters'. The status bar at the bottom indicates 'Connected Server(s): 2 Objects: n/a' and 'Ready'.

name	title	hourly_rate	manager
Alcott, Scott	Sr Programmer	\$50.00	Wolfe, Neal
Applegate, Donald	Analyst	\$51.00	Wolfe, Neal
Bee, Charles	Sr Programmer	\$43.00	Fielding, Wallace
Belter, Kris	Programmer	\$33.00	Alcott, Scott
Beringer, Tom	Programmer	\$41.00	King, Richard
Beveridge, Fern	Project Leader	\$57.00	Wolfe, Neal
Bluff, Clarence	Programmer	\$24.00	Jones, Ashley
Bridges, Debra	Sr Programmer	\$48.00	Parsons, Carol
Chung, Arthur	Programmer	\$21.00	Ortega, Julio
Downing, Susan	Programmer	\$29.00	Bee, Charles
Fielding, Wallace	Project Leader	\$47.00	Jones, Betty
Fine, Laurence	Sr Programmer	\$42.00	Jones, Betty
Hilton, Connie	Programmer	\$37.00	Bridges, Debra
Jones, Ashley	Sr Programmer	\$49.00	Turner, Russell
Jones, Betty	Project Leader	\$66.00	Turner, Russell

We obtain the results of our query.

To understand the code architecture of Ingres. [<Click Here>](#)

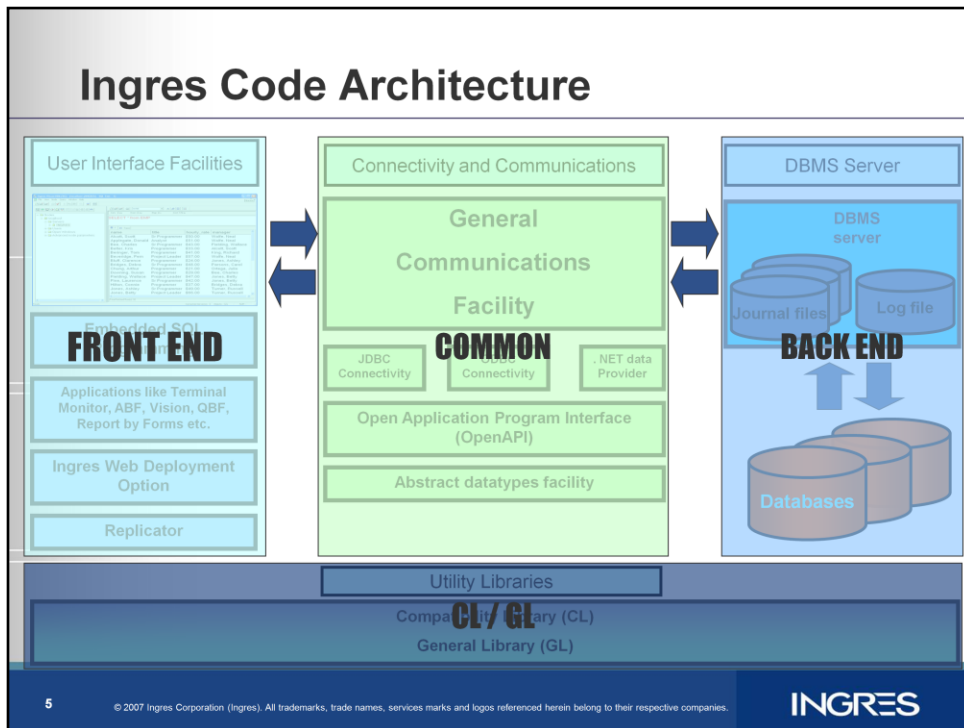
let us examine how Ingres processes this query [<Click Here>](#)



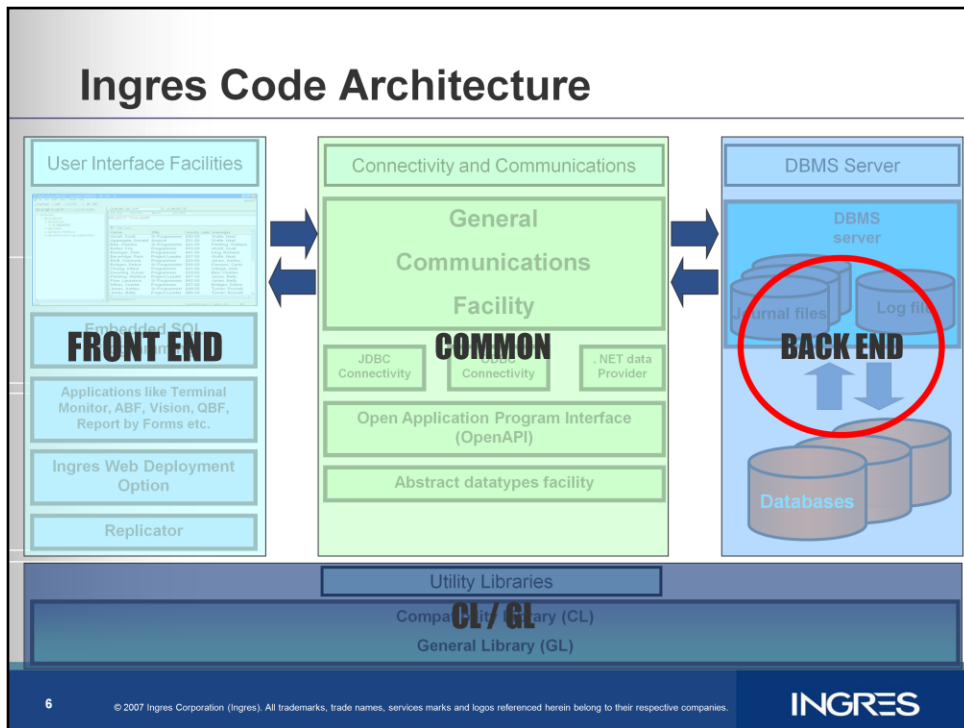
1. The Query is processed by Global Communications facility which communicates the data and results between the terminal monitor to the DBMS server.
<Click Here to Animate>
2. The DBMS server and its associated processes process the query and send the results to the client through GCF.
<Click Here to Animate>
3. Broadly categorizing we can summarize the code into
 - User interface facilities
 - Connectivity and Communications
 - DBMS server
 - And Utility libraries to interact with Operating system.
1. Ingres has several kinds of user interface for example
 1. Embedded SQL precompiler for programmatic operations of database.
 2. Ingres applications like, SQL Terminal Monitor, Query by forms (QBF), Report by Forms (RBF), VISION etc.
 3. We have Ingres Internet Commerce (ICE) for web deployment.

These user interfaces and others can be commonly grouped as Front End for Ingres.
- Similarly in connectivity tools we have
 - Java Database connectivity (JDBC) driver
 - Open Database connectivity driver (ODBC) driver
 - Dot Net Data provider
 - Open API a C api to Ingres etc.

These connectivity tools and others comprise the common facilities for Ingres
6. In addition a Library of all machine dependent code is provided under compatibility library or CL.
In order to ensure portability of Ingres code all Ingres facilities must use Compatibility library functions for machine dependent calls
for example I/O, memory or ipc calls etc.
7. Other non platform specific general routines which are used by all Ingres facilities are grouped under General Library or GL.



1. In Ingres Source these broad categories are grouped as
 1. Front End ,
 2. Back End,
 3. Common which has code used by both Front and back and
 4. CL/GL which contain platform specific code and general use code.



Let us understand the details of Ingres backend Facilities.

Back End Components



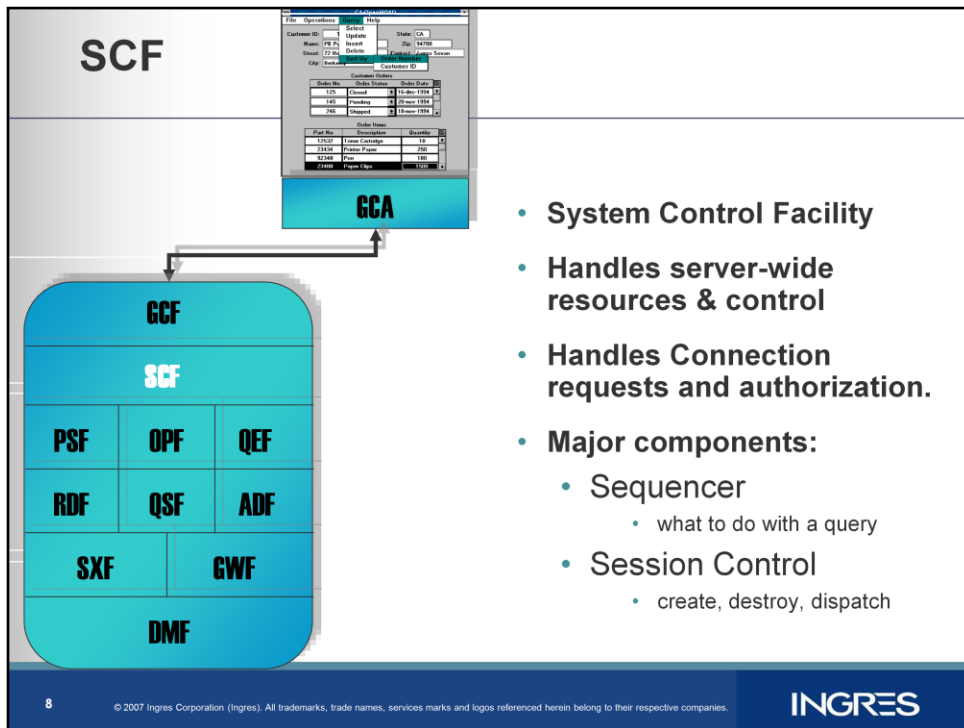
Back end is the most important part of the database.

It carries out storage, parsing, optimization and execution of the users query.

It also carries out Logging, Locking, archiving and recovery operations on the database.

This diagram shows the various facilities comprising back end.

Let us examine their interaction at more detail



SCF stands for systems control facility.

After a Query message is received by backend via GCF, SCF decides what to do with the query.

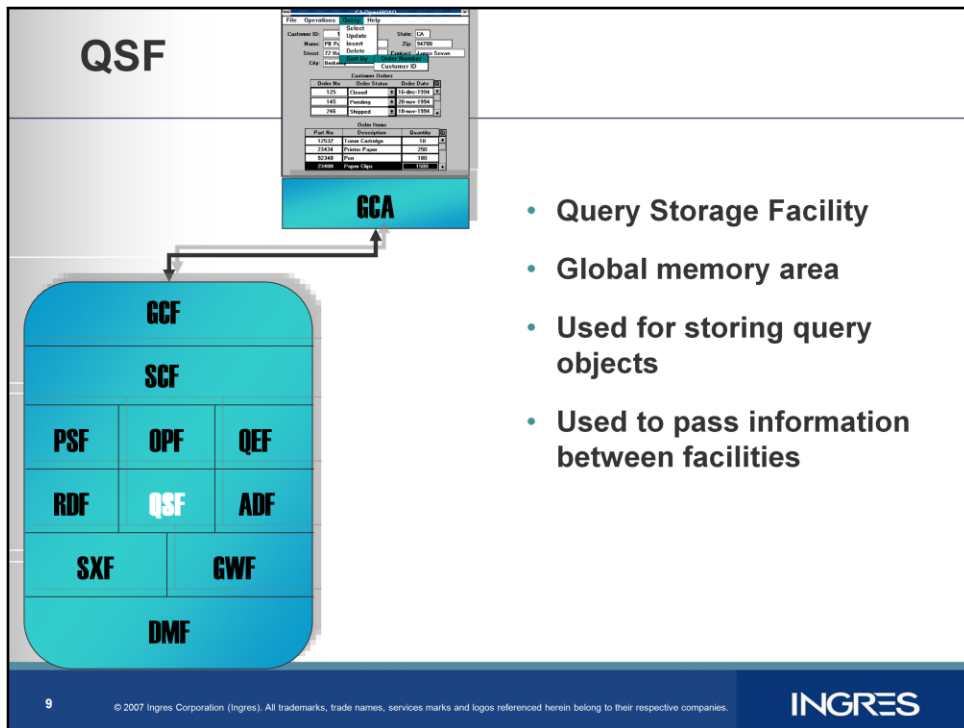
SCF handles server wide resources and controls

SCF also handles the connection requests and authentication process.

Major components of SCF are

a) Sequencer – What to do with a query.

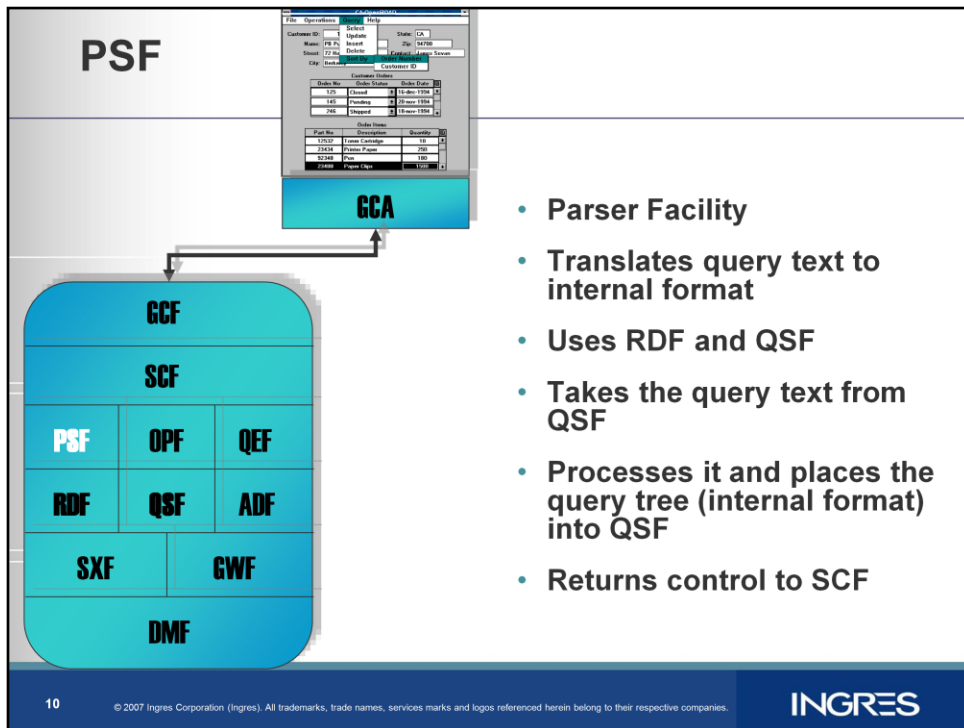
b) Session Control – How to Create, destroy and dispatch.



QSF stands for Query Storage Facility to store query objects

This serves as Global memory area

It is used for storing query objects and pass them between facilities.

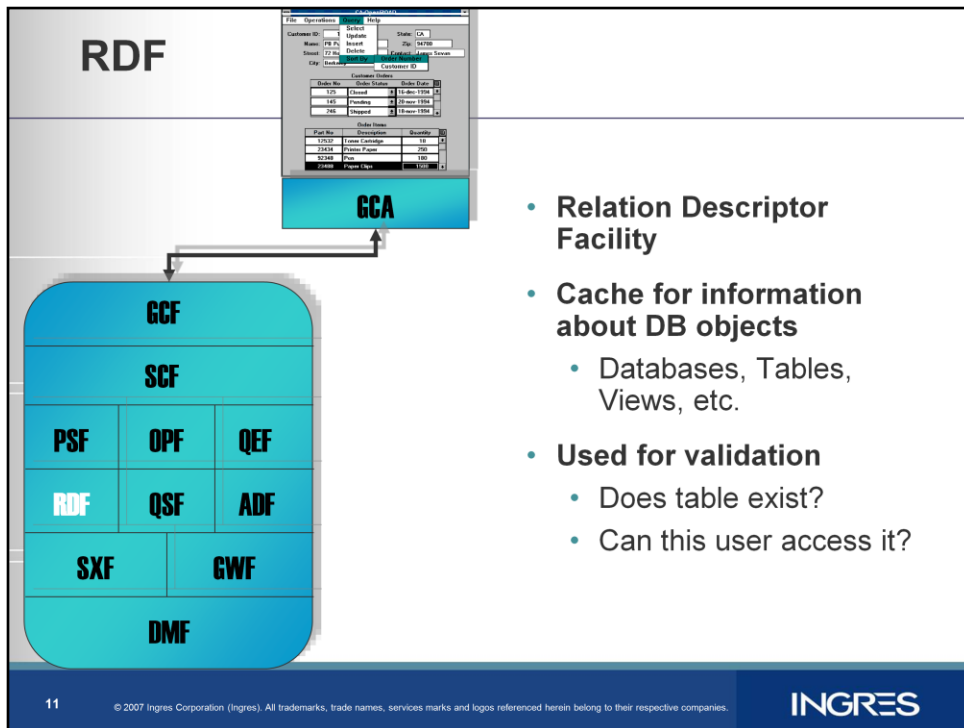


Parser Facility

Parses a query and translates query text into internal format (Query Tree).

It uses RDF for validation and QSF for obtaining query data and placing the prepared Query Tree.

PSF is called from SCF and after parsing the query the control is returned to SCF



Relation Descriptor Facility

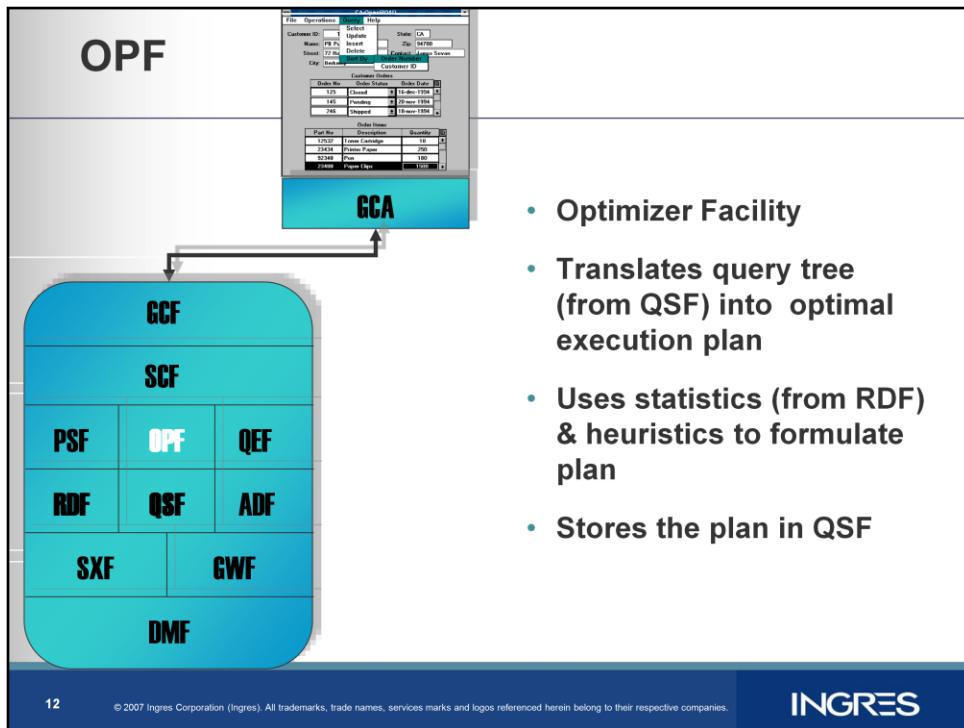
Cache for information about DB objects

Databases, Tables, Views, etc.

Used for validation

Does table exist?

Can this user access it?

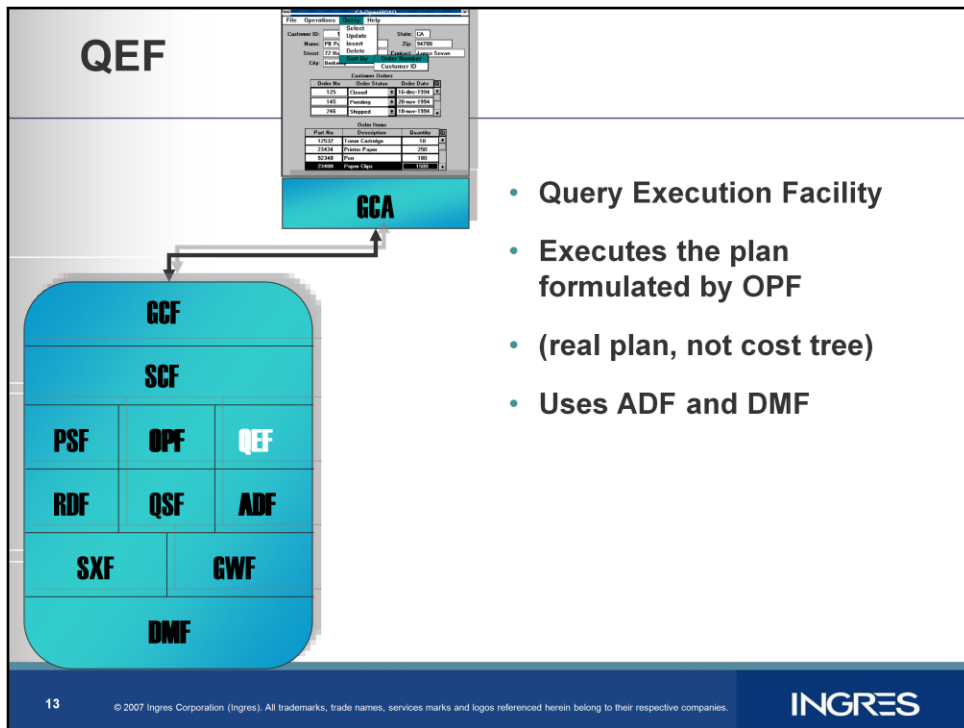


OPF is the query optimizer facility

It translates a Query tree from QSF and creates an execution plan for it.

It uses statistics from RDF and heuristics to formulate the query plan

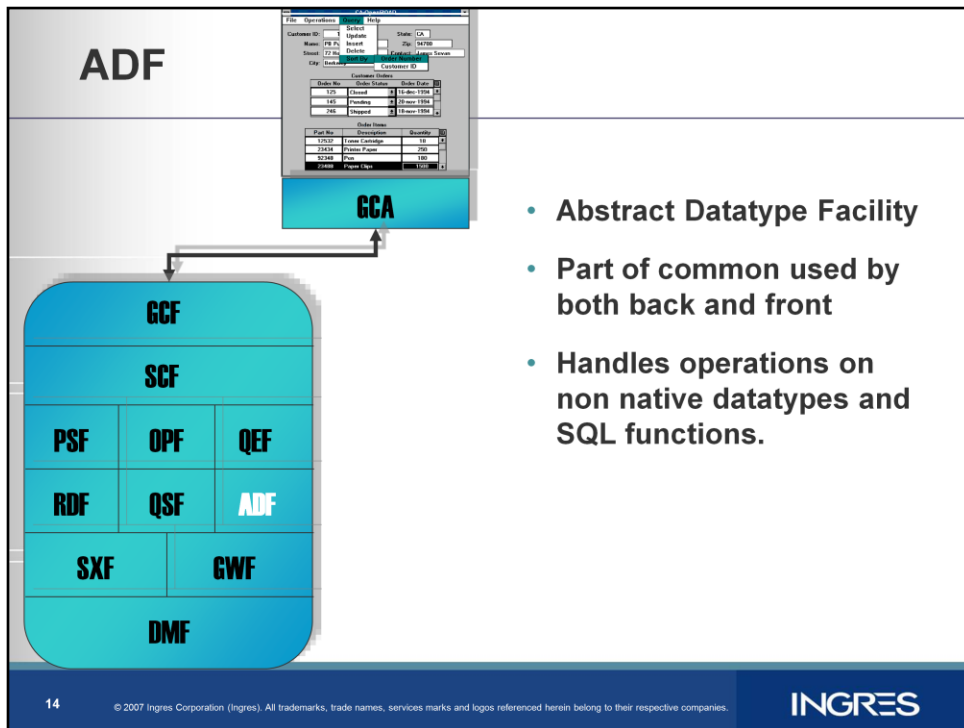
A formulated Query plan is stored in QSF



QEF is the Query Execution facility.

It executes the query plan prepared by OPF

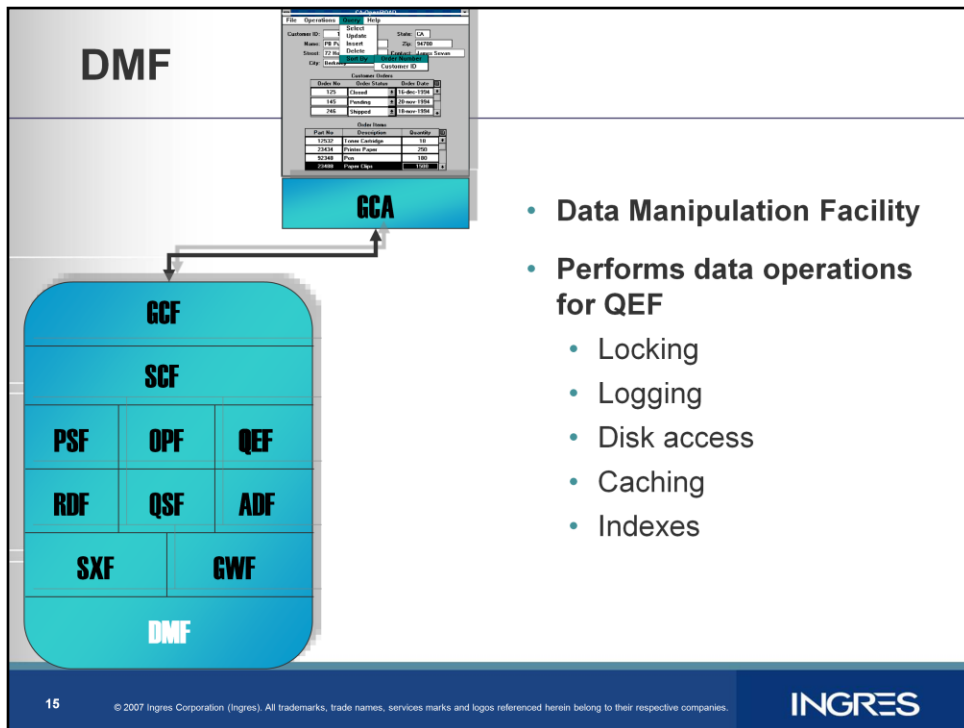
It uses ADF to handle datatype operations and DMF to carry out the specific query instructions.



ADF is abstract datatype facility and is part of Common

As it is used both by front and back end.

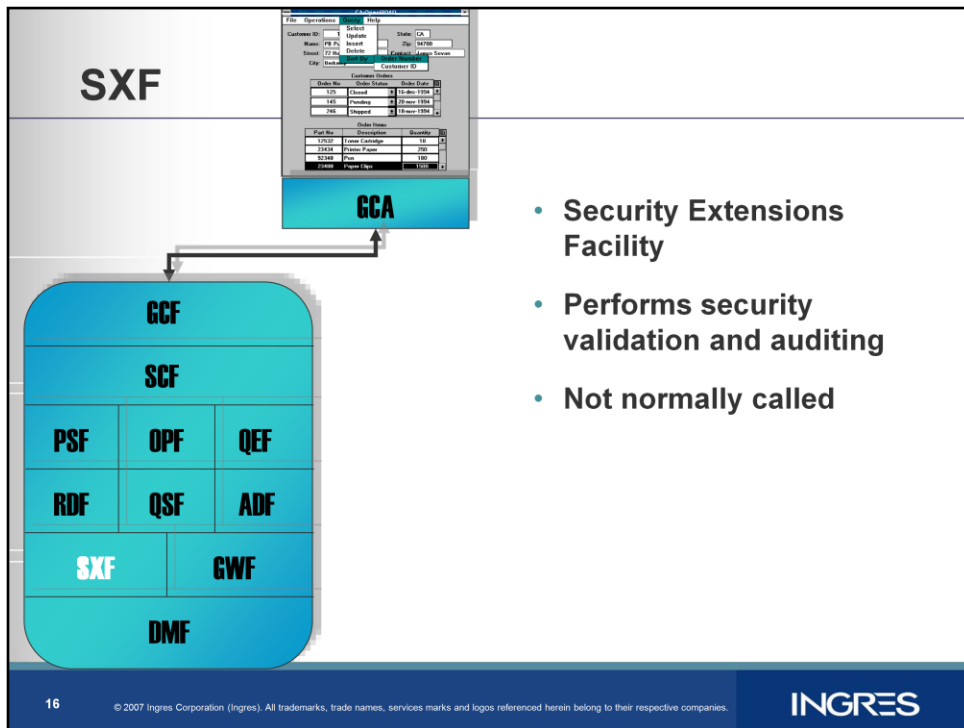
Handles non native operations on datatypes eg date, money operations.



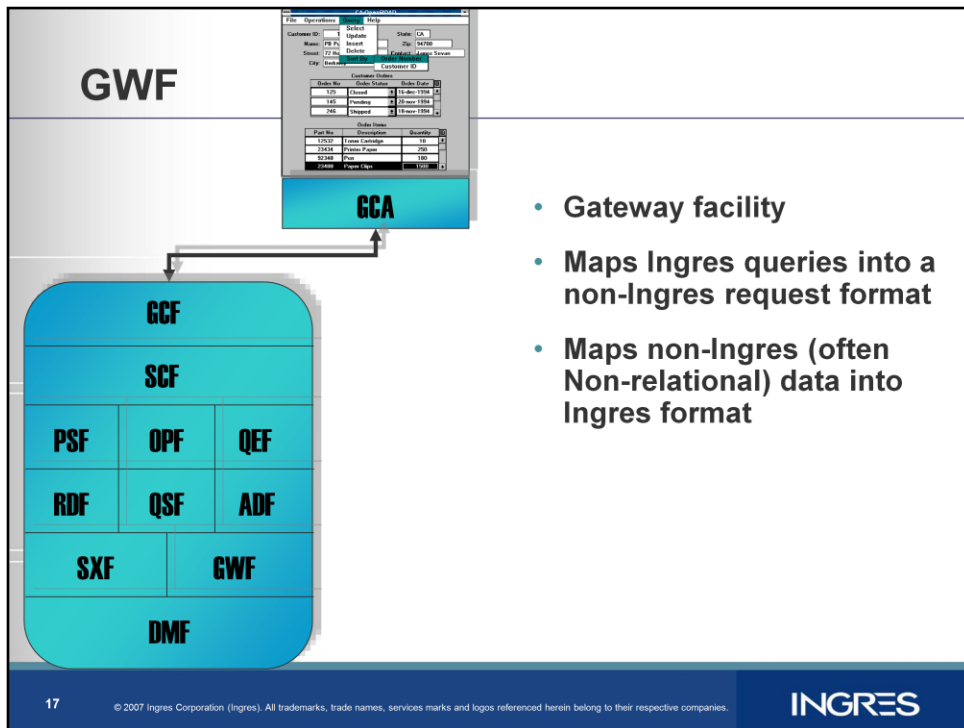
DMF is data manipulation facility.

It performs data operations.

It provides Locking, Logging, Disk access, Caching and Indexing processes.



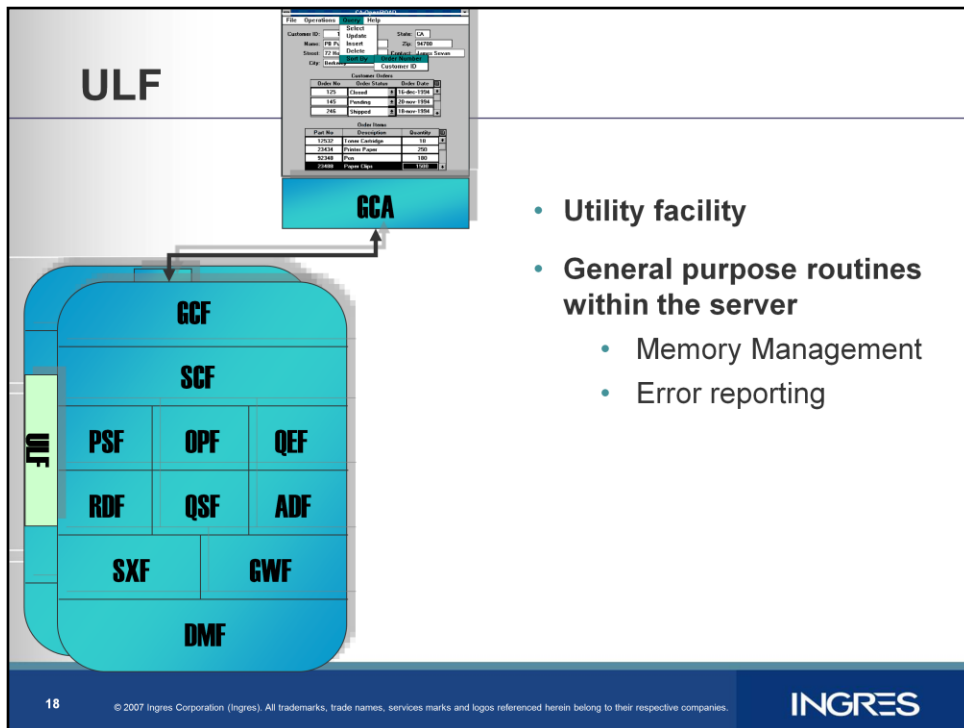
SXF is the security extension security.
 It has routines for security validation and auditing.
 It is only used for enhanced security configurations.



GWF stands for Ingres gateways.

It maps Ingres queries into a non Ingres request format.

It also maps non-Ingres data into Ingres format



ULF is the Utility facility used by all back end
 It contains general purpose routines for
 memory management
 error reporting

Other Back End Facilities

- Log file operations (LG)
- Lock manager (LK)
- Logging and Locking Shared Components (LGK)
- Transaction Processing Facility (TPF)
 - Transaction state tracking
 - Distributed coordinator recovery logging
 - Distributed transactions recovery
- Remote Query Facility (RQF)
 - Coordinates distributed query plan execution.

19

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Log file operations. (LG) Used by DBMS for database logging operations.

Lock manager. (LK) Used by the DBMS to provide concurrency control over database objects.

(LGK) Shared components of Logging and Locking facility are in LGK facility

TPF manages distributed transactions for STAR

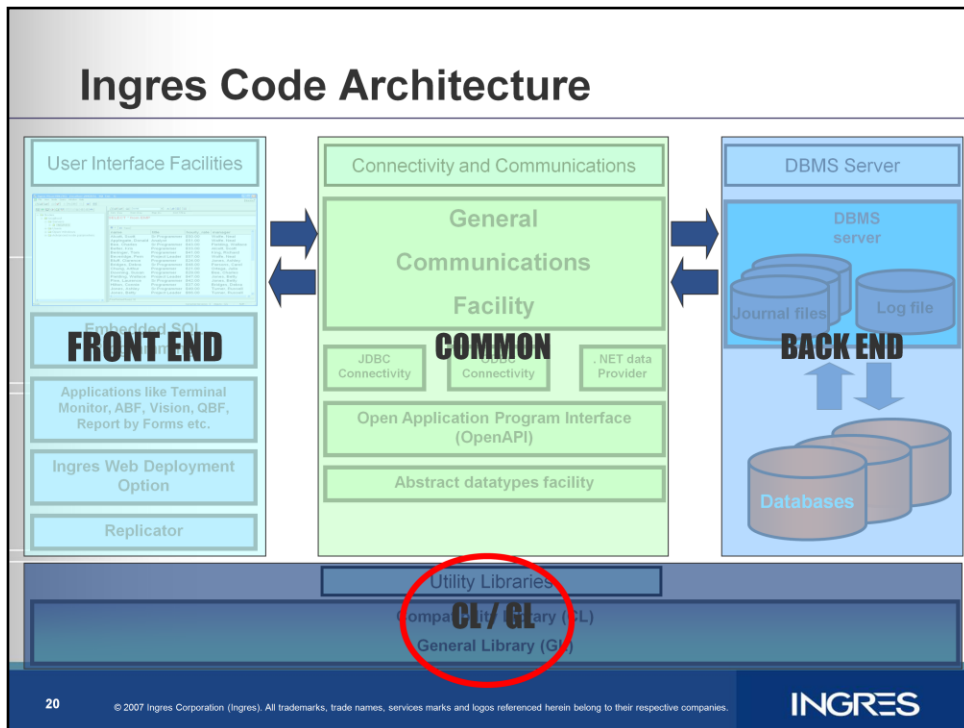
It tracks state of all transactions under the server

Ensures necessary loggings is done for distributed coordinator's recovery

Responsible for distributed transactions recovery.

RQF is remote query facility

Also used for STAR, Coordinates the execution of a distributed query plan.



Let us discuss the details of Compatibility Library CL and General Library GL

Ingres Compatibility Library (CL)

- **Interface to operating system.**
- **Small set of machine dependent code**
- **Common interface for**
 - Memory
 - I/O
 - IPC ...
- **May not call higher level Ingres code**

21

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

CL or compatibility library is Ingres interface to underlying operating system.
It provides library of functions which contains smallest set of machine dependent code

for Ingres base product to run on any system.

Provides the common interface for

Memory

I/O

IPC ...

CL is a layer within the Ingres architecture.

CL is not allowed to call higher level Ingres code.

All Ingres code must use CL functions to ensure portability across supported platforms.

CL Components

- Bit array manipulation (BT)
- Customer information authorization (CI)
- Checkpoint/Recover (CK)
- Character manipulation (CM)
- Change protection for file operations (CP)
- Control system (CS)
- Conversions (CV)
- Cluster extension (CX)

22

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

CL consists of Following components

Bit Array Manipulation (BM) Operations on arbitrary length bit strings, such as logical and, or, not; test bit, set bit and clear bit.

Customer information authorization (CI) Used for Authorization routines, contains a DES-like encryption system.

Checkpoint/Recover functions. (CK) Makes disk or tape checkpoints of database directories. Used only by DBMS executables.

Character Manipulation. (CM) International character support, used by all Ingres programs to manipulate and classify strings and characters that may be either normal single-byte C language strings or multiple character strings. Use CM operations rather than normal C pointer manipulation.

Change Protection for file operations.(CP) Creates and writes files for "copy out" operations with the correct user permissions. Used only by the DBMS Auditdb program

Control System. (CS) Threading control mechanism used by DBMS. CS supports only non preemptive threading but there is a POSIX threading model counter part (CSMT) on UNIX/Linux.

Conversions. (CV) A collection of data conversion functions used by all Ingres programs, such as character string to anything, or anything to character string.

Cluster Extension (CX) Contains functions needed to support Ingres clusters.

CL Components

- Database I/O (DI)
- Dynamic loading (DL)
- Data structure compiler (DS)
- Error message text handling (ER)
- Exception handling (EX)
- Floating point handling (FP)
- General communication (GC)
- Global symbol values (GV)

23

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Database I/O. (DI) Page oriented I/O used by DBMS to access database files.

Dynamic Loading. (DL) Provides access to user-written 3GL code. Creates dynamic-link object modules in machine native format and loads them into the current process.

Data Structure Compiler. (DS) Used for “compiled form” source files. User interface programs write files containing data structure definitions that will be compiled and subsequently linked into Ingres applications.

Error message text. (ER) Enables Ingres programs to get all text messages from external files. The error message look up operations (ERlookup or ERget) obtains the message text given a message number. The ERsend function writes messages to error log files.

Exception handling. (EX) Errors and events handling. Interacts with C signal and setjmp/longjmp facilities.

Floating Point handling. (FP) Floating point operations and functions.

General Communication. (GC) System dependent support for GCA facility, provides inter process communication between applications and servers.

Global symbol Values. (GV) Defines global variables of general use to all Ingres executables.

CL Components

- Identification (ID)
- Journal file operations (JF)
- File location manipulation (LO)
- Memory manipulation (ME)
- Math functions (MH)
- Mutual exclusion (MUCL)
- Name/value pair lookup (NM)
- Other language interface (OL)

24

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Identification. (ID) Used to Determine user names.

Journal File operations. (JF) Used by DBMS executables to access journals and dump files used in on-line checkpoint.

File location manipulation. (LO) Hides system dependent file names, mostly used by user interface programs.

Memory manipulation. (ME) Used by DBMS executables to access journals and dump files used in on-line checkpoint.

Math functions. (MH) This provides the collection of math operations used in both the DBMS and in user-interface programs.

MUCL – operating system specific mutual exclusion

Name/value pair lookup. (NM) Corresponds to UNIX/NT environment variables or the VMS logical symbols.

Other Language interface. (OCL) A way for C program to call functions that are written in other languages, such as FORTRAN. Used only by ABF.

CL Components

- Process control (PC)
- Permissions (PE)
- Queue manipulation (QU)
- Security auditing (SA)
- Stream file I/O (SI)
- Sort file manipulation (SR)
- String manipulation (ST)
- Test communications (TC)
- Terminal handling (TE)

25

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Process Control. (PC) General process control functions used in all Ingres programs, such as exit, start another process, sleep etc.

Queue manipulation. (QU) Operates for doubly linked list.

Stream files I/O (SI) Stdio.h like operations for user-interface programs not for DBMS code

Permissions. (PE) Set permissions on files created by SI, and is therefore only used in user-interface programs

Sort file manipulation. (SR) Used by DBMS to operate on Files temporarily used for sorting.

String manipulation. (ST) Functions used everywhere for manipulating NULL terminated C strings. The strings may contain multi-byte characters

Test Communications. (TC) Test point access to user-interface programs taking input from SI and TE.

Terminal Handling. (TE) User Interface Access to character terminal for input and output.

CL Components

- Time functions (TI)
- Trace functions (TR)
- Utility invocation (UT)

Time functions. (TI) A collection of functions for determining the current timeprocess accounting information.

Trace functions. (TR) Functions to write messages to the log file

Utility invocation. (UT) Operations for user interface, to compile and execute other programs and to print files.

General Library Components

- Generic routines used across all platforms
 - **Parameter management (PM)**
 - **Splay trees (SP)**
 - **Management objects (MO)**
 - **Mutual exclusion (MU)**

GL contains utilities which are platform independent yet are useful for all facilities.

GL also contains headers for CL function which are available to use for other facilities as external functions.

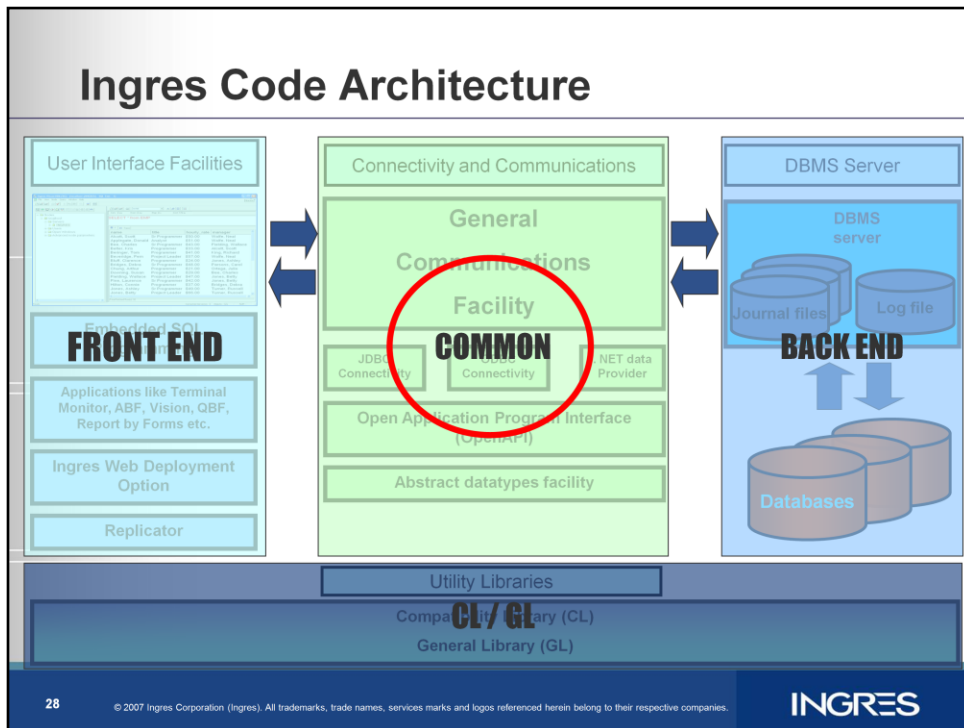
Some libraries provided by GL are:

Parameter Management (PM) - Provides uniform way of getting at static system parameters. Used to store server parameters and other installation configuration information.

Splay Trees (SP) - Provides access to very efficient in-memory tree structure suitable for symbol tables, lookup tables, priority queues etc

Management Objects (MO) - Provides a way of making program variables visible as named objects through management interfaces

Mutual Exclusion (MU) - Provides semaphore operations for GL and 3GL code that needs to be thread safe and sometimes runs in a threaded environment and sometimes doesn't.



Now let us discuss components included in Common.

Common Components

- **Abstract Datatype Facility (ADF)**
- **Common Utility Facility (CUF)**
- **General Communications Facility (GCF)**
- **Ingres .NET Data Provider**
- **Java Database Connectivity (JDBC)**
- **Open Database Connectivity (ODBC)**
- **Open Application Program Interface (OpenAPI)**

The components common to front end and backend are

General communications facility (GCF)

Abstract datatype facility (ADF)

Application program Interface (API)

Java Database connectivity driver (JDBC)

Open Database connectivity driver (ODBC)

Ingres DOT NET data provider.

Abstract Datatype Facility (ADF)

- **Non-native type operations**
 - Date
 - Money
 - Long varchar
- **SQL tuple functions**
 - *Select **sum** (cost) from po_table*
 - *Select **concat** (firstname, lastname) from empdata*
- **Coercion and collation.**
- **Basis for User Defined Types (UDT)**

30

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Abstract datatype facility handles operations on non-native SQL data types
eg Date, Money, Long Varchar

Any SQL functions eg Aggregate operations like SUM, COUNT etc are handled in ADF

ADF handles Coercion between various SQL datatypes eg varchar to money etc.

ADF also handles sorting and comparison based on collation being used.

ADF has provision for UDT's.

Any new datatype and operations should be added to ADF

Common Utility Facility (CUF)

- **Common utility for identifiers (CUI)**
- **Common utility for error translation (CUE)**
- **Common connection manager (CCM)**
- **Common utility for thread coordination (CUT)**

Common utility for Identifiers,

Contains routines for manipulating identifiers, Normalize and translate identifier, Unnormalize identifier, Find length of identifier etc.

Common utility for Error reporting (CUE)

has routines for translating SQLSTATEs to GENERIC errors

Common connection manager is used to create and drop connections to a data base for a user.

By reusing dropped connections for the same data base & userid, the connections are made much faster.

Common utilities for thread coordination

Is a collection of service modules designed to provide thread communications and coordination for cooperating threads within a single process.

General Communications Facility (GCF)

- **Processes and protocols for Ingres communications**
- **Transfers both *commands* and *data***
- **Application protocols supported**
 - initiation and termination of associations
 - query processing
 - interruption
 - transaction management with one or two phase commit
- **Multiple network protocol implementations**
 - TCP/IP
 - DECNET
 - SNA

32

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

General Communications Facility

Handles processes and protocols for communication between various components of Ingres

Allow transfer of commands and data between any client, any server, or gateway

Supports application protocols i.e. initiation and termination of associations, query processing, interruption, and transaction management with one or two phase commit.

Uses standard proprietary network protocol implementations e.g. TCP/IP, DECNET and SNA

GCF also has the Name server which serves as a register to various dbms processes or servers.

And Net server to enable connection to remote Ingres installations.

Ingres .NET Data Provider

- Allows applications to connect to Ingres database
- Moves data between DBMS and application
- Provides interfaces for the application to invoke
- Provides interfaces to ADO.NET
- Communicates with Ingres Data Access Server
- Stand-alone
- Namespace
- **Ca.Ingres.Client**

33

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

.NET framework provides a rich set of classes for its Common Runtime environment.

.NET provides classes for data access Interfaces to retrieve and update data
Ingres .Net data provider allows .Net applications to connect and transmit data to Ingres Database.

.Net Provider provides

Interfaces for the application to invoke

Interfaces to ADO.NET

Communication with Ingres Data Access Server

.Net Provided has Stand-alone Direct wire communication with Server

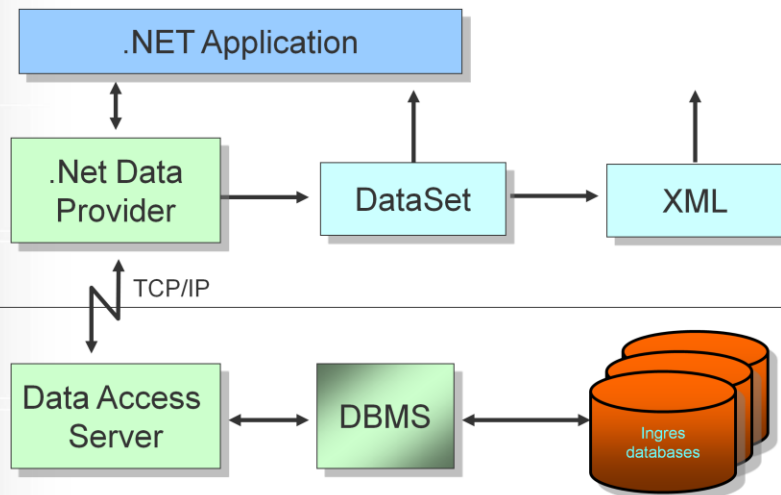
(does not require Ingres components on client machine, i.e Ingres/Net Client not needed)

Uses Namespace: Ca.Ingres.Client

It is written in 100% pure C#, Useable by any .NET managed language C#, VB.NET, J#, COBOL, Perl, etc.

It provides for Connection Pooling Requires .NET 1.1 or later Visual Studio .NET 2003 Integration

Ingres .NET Data Provider: Data Access



34

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Here a .Net Data provider connect to Data Access server through TCP/IP

Populates the Ingres data to the dataset created by .Net application, which can be converted to XML

Datasets are In-memory copy of database data, Passed to other tiers/components as XML.

JAVA database connectivity (JDBC)

- **Full JDBC 2.0 Compliance**
 - JDBC 2.0 Standard Extension
 - JDBC 2.1 API
 - J2EE Application Server Support e.g. JBOSS, Websphere
- **Type 3**
 - Pure Java JDBC Driver
- **Server Based**
- **3-Tier Architecture support**

35

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

JDBC is call level Java API for database Independent connectivity.

It enables a Java application to establish database connection, send SQL statements and process the results.

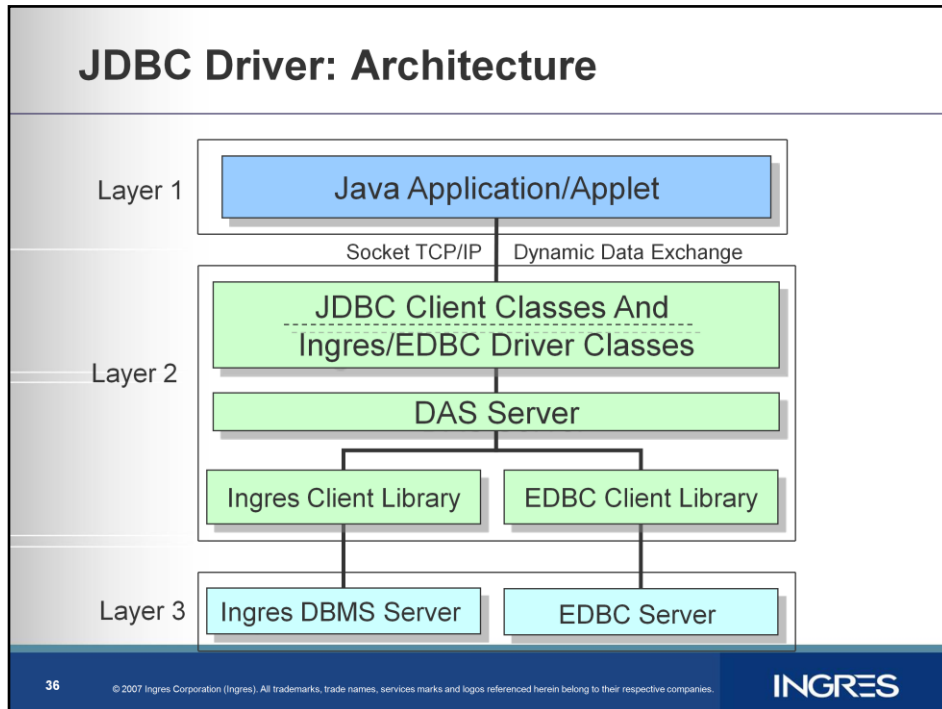
Supports JDBC 2.0 Standard Extension and JDBC 2.1 API

In Ingres we support We have Type 3 Native JDBC Driver (v 2.0)

Type 3 - pure java code works with dbms middleware (NET Protocol) to access database

Server Based (ijdbc) JDBC server translates to DBMS Protocol

Supports 3-tier model



A Java capable browser in Tier 1 connects, via TCP/IP, to the JDBC layer in Tier 2. It imports the JDBC Classes it needs to communicate with the JDBC server in Tier 2. It uses these classes to initiate a database connection and issues a number of queries.

Tier 2 accepts the remote request for a database connection and services the query(s). It does this by converting from the JDBC API Calls the client makes into an internal Ingres protocol. The client requests are serviced by the JDBC Client Classes. The JDBC Client Classes reformat and pass the requests on to the Ingres/EDBC specific classes. The Ingres/EDBC Classes convert the requests into an internal form which is then passed to the IIJDBC DB Middleware layer. The middleware layer communicates directly either with the standard client libraries for Ingres or for EDBC. These libraries pass the request on to the DBMS server in Tier 3.

In Tier 3 the data is retrieved from the server and passed back up to tier 2 where it is formatted correctly for transmission to the application in tier 1.

ODBC Driver

- ODBC 3.5 driver
- Uses Ingres API and Ingres Net.
- **Manages**
 - Connection
 - Catalog functions
 - Information, capabilities and options
 - SQL transformation
 - Data type conversions
 - Error reporting

37
© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.
INGRES

ODBC is standard application programming interface.

It Supports SQL syntax, Has a Defined call-level interface, standard way to connect and log on, standard datatypes and defined result sets and Standard error codes.

Ingres ODBC driver Includes 2.8 ODBC Driver features and ODBC 3.0 and 3.5 Features

Ingres ODBC Driver manages connections, catalog functions, information capabilities, SQL transformations, Datatype conversions and Error reporting.

It uses Ingres API and Ingres Net.

It is Thread-safe,

Has Connection pooling support

And Microsoft Transaction Server support

Connection

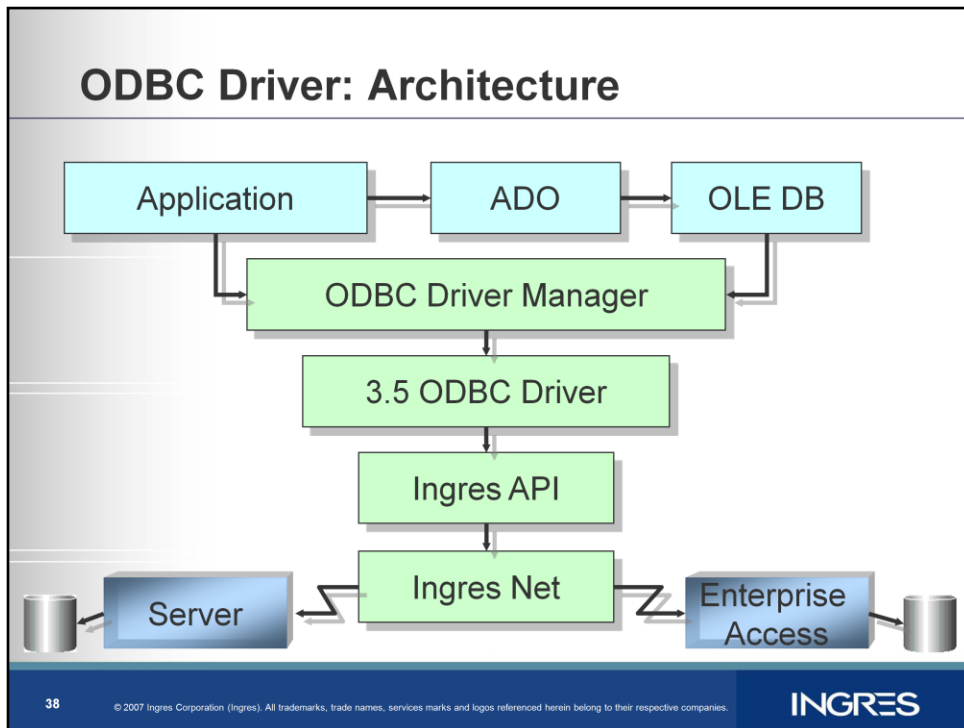
Catalog functions

Information, capabilities, and options

SQL transformation

Data type conversions

Error reporting



This is the general architecture of Ingres ODBC driver.

The Application connects to the Driver Manager,

Which connects to the ODBC Driver

Which connects to the Database server through Ingres Net via Ingres API.

Open Application Program Interface (OpenAPI)

- **C language interface for database access**
- **Asynchronous processing**
 - Uses callbacks.
- **Synchronous processing**
- **Multithreaded application support**
 - Asynchronous Calls.
 - Operating system thread support
 - Example
 - Applications servers
 - Windowing applications

39

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

API is C language call-level interface for access to Ingres II server

It is a set of C language functions to Enables SQL access to Ingres and Enterprise databases

Support for native operating system threads.

An alternative to Embedded SQL or Ingres ODBC additional capabilities for asynchronous processing

Asynchronous or synchronous calls are possible

Asynchronous calls use call back functions.

Supports Multithreaded applications with

Asynchronous calls

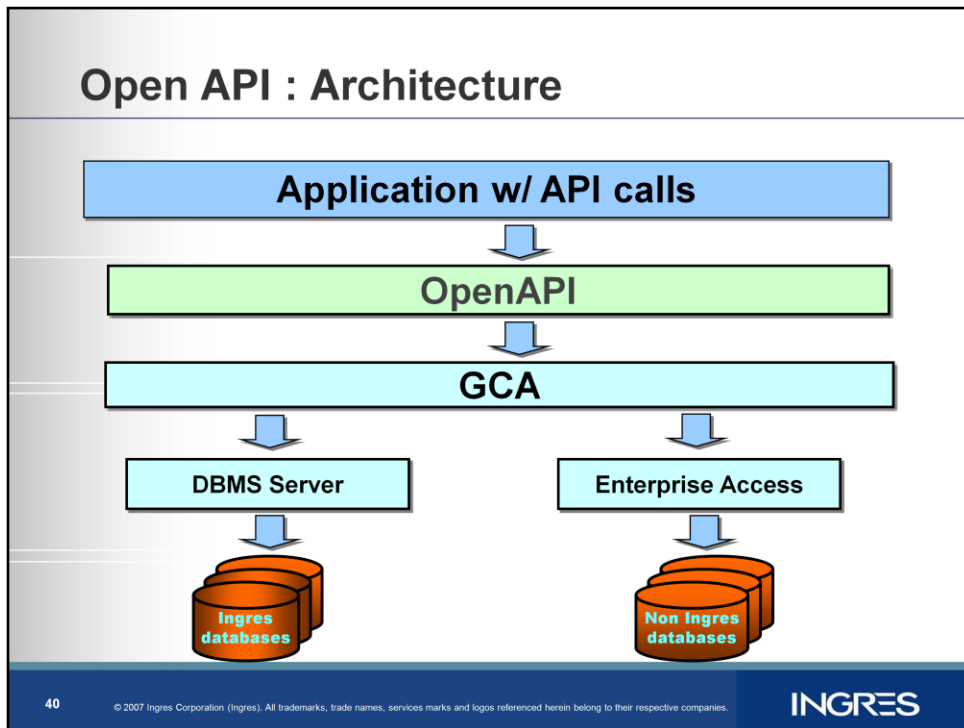
Operating system thread support

Example

Application servers

Windowing applications

API are also used by internal components odbc and jdbc driver.



A C application uses API calls to connect and access the database.

API uses Ingres net to access the dbms server.

Ingres Net automatically converts datatypes for heterogeneous networks

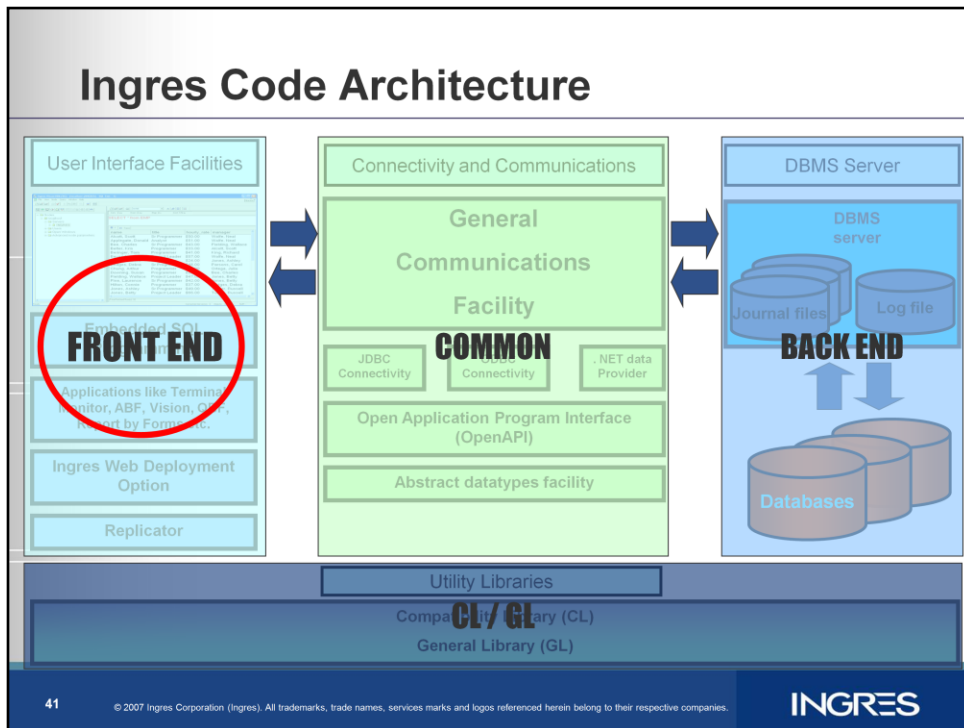
Most Ingres data types map to "C"

API provides conversion for some data types eg:

- Money

- Decimal

- Date



Finally moving on to the front end components.

Front End Components

- **Embedded SQL**
- **Character based tools**
 - Applications by Form (ABF)
 - Query by Form (QBF)
 - Report by form (RBF)
 - Terminal Monitor (TM)
 - Vision
 - ...
- **Web deployment option**
- **Replicator**

42

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Here are the components of Ingres front end.

Namely

Embedded SQL support

Character based tools like TM, QBF, RBF, VISION etc.

ICE

Replicator

Front End Components: ESQL

- **Embedded SQL and QUEL**
- **Language supported**
 - C/C++
 - Cobol
 - Ada
 - Fortran
 - BASIC
 - Pascal
- **Static and Dynamic SQL**
- **Uses LIBQ and LIBQGCA**

43

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Embedded SQL provides precompilers for programmatical execution of Queries.

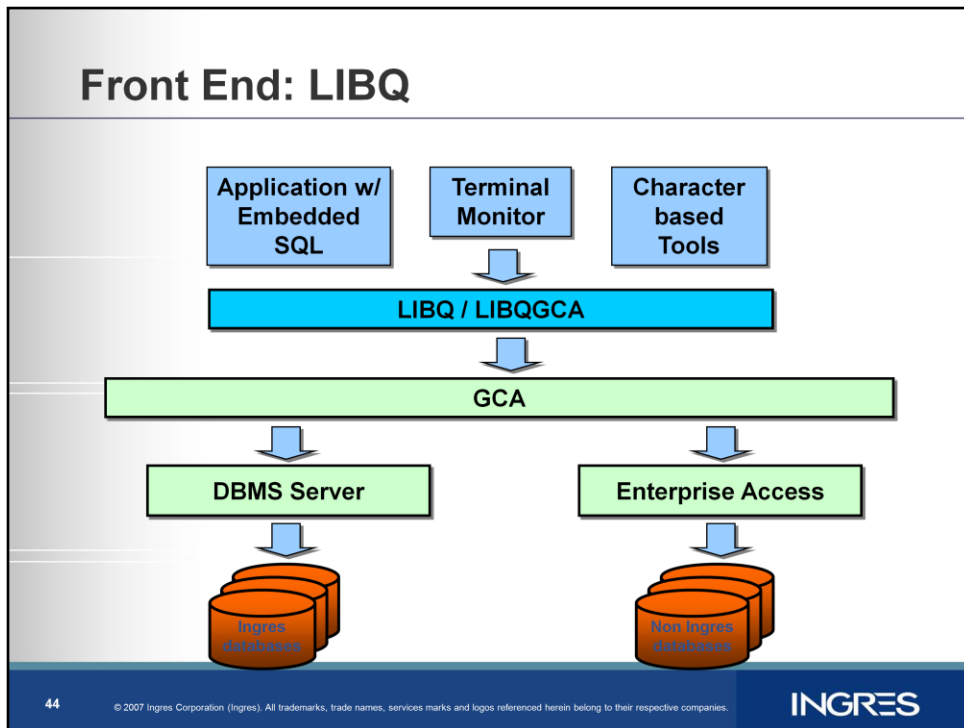
Precompilers compile the embedded SQL programs and convert them to native programming language eg C code.

Ingres has ESQL support for SQL as well as QUEL precompilers

Language precompilers are available for C, Cobol, Ada and Fortran.

Embedded SQL has support for Static and dynamic SQL

Uses LIBQ and LIBQGCA to interact with DBMS server.



LIBQ/LIBQGCA provide common library of routines which are used by most front end programs like

Esqcl, Terminal monitor etc to provide access to dbms server

It connects to the database server using GCA calls.

Front End Components: Character Based Tools

- **Application By Forms (ABF)**
- **Query By Forms (QBF)**
- **Report By Forms (RBF)**
- **Terminal Monitor (TM)**
- **Vision**
- **Others**

45

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

ABF – Application by Forms is a workshop for creating customized forms-based applications. It takes you through the development of a new application by presenting a series of forms that can be filled in and also menus that you can choose. The resulting application uses Ingres forms and menus to access the database and perform queries updates and reports. ABF allows you to define, test and run fully developed applications without using conventional programming languages. It also has support for writing 4GL scripts.

QBF – Forms based querying tool

Can append delete update and view data without any knowledge of SQL

Default display forms are provided by QBF using table definitions.

RBF -Interactive forms based report creation tool

Produce report immediately with preview

Customize reports using RBF editor.

No programming needed as Report writer code is generated by RBF

TM - Allow users to send query and receive results on the terminal

Can accept SQL or QUEL statements

Can accept input from file for batch processing.

Invoked using SQL database name

Vision – High productivity application code generator

Produces the screens and code

A Frame has Forms + code for users interaction with the form and database access

And procedure

Front End: Web Deployment Option

- Web Data access
- Internet Commerce Enabled(ICE) Server
 - Security, session management, storage management
- An example ICE query

```

<!-- #ICE
DATABASE=`demodb`
SQL=`SELECT *
      FROM plays`
ROWS=`5`
TYPE=`TABLE`
-->

```

title	performed	type
The Two Gentlemen of Verona	1598	comedy
The Taming of The Shrew		comedy
Henry VI part 1	1591	history
Henry VI part 3	1595	history

46

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, service marks, or registered trademarks belong to their respective companies.

INGRES

Ingres web deployment option is an extension to the DBMS that allows the inclusion of data direct from an Ingres data source into an HTML page.

ICE is analogous to embedded SQL in HTML giving access to business logic and a standard presentation with (HTML)

ICE Server provides security, session management, storage management and extensions to the macro language.

Auto commit transactions (default) relinquish control after use. Named transactions are maintained for the duration of the session context and the connection is not returned to the pool until either a commit or a rollback.

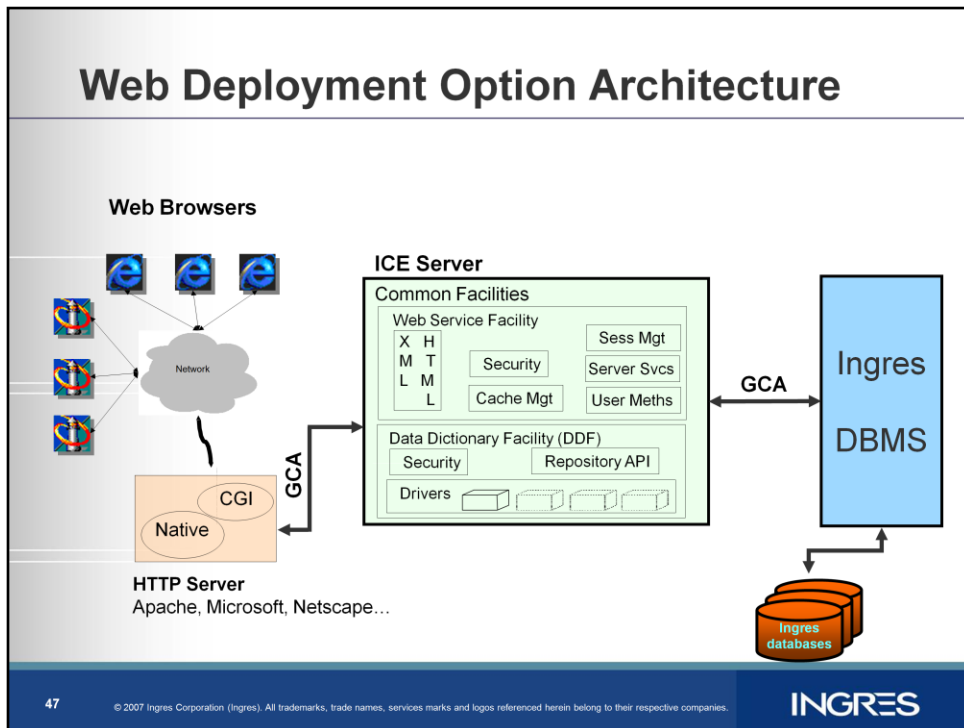
Here is a very simple ICE query, a single table select.

In the body of the query we set the database connection to be used, 'ice tutor'.

Set the SQL statement and if this is the first time, open the transaction 'Complete' and the Cursor 'Works'. Subsequent invocations will re-use the tx & cursor.

Set number of rows = 5 to be presented to the user

The result set will be automatically formatted as an HTML table or other available formats.



Web browsers connect to an HTTP server.

By using HTML pages, an end-user can access database information through the ICE server.

The ICE server is responsible for the communication between the HTTP server and the database server and guarantees access control, coherence and optimisation

When a web user directs their browser at an ICE extended URI, the HTTP server passes the request on to the ICE server via the ICE HTTP CGI Client.

This forms up a GCA request containing the ICE Query and sends it to the ICE server which processes it, accessing the DBMS and File System as required.

It formats the result either into legal HTML under direction of the HTML template file or XML and returns the result to the HTTP server and then to the user.

Front End Components: Replicator

- **Replication Change Recording**

- Part of the DBMS Server
- Detects DBMS changes

- **Replicator Server**

- Propagates changes
- Uses Open API

- **Replicator Manager**

- Replicator configuration
- Uses Embedded SQL

48

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Ingres replicator allows maintaining a copy of source data at one or more targets. Any change to a source database can be applied to one or more destinations. Ingres Replicator detects collisions, replication of horizontal or vertical subsets

Ingres Replication change recording is part of DBMS

“DMF change recorder” in back end is integrated code which detects when a table or column under replication scheme is modified.

When a user COMMITS a transaction, the DMF Change Recorder adds the transaction ID to a shared memory queue. If the user rolls back, the input queue, shadow and archive INSERTs will be undone as part of the user ROLLBACK. It puts the rows in a distribution queue and signals the Replicator Server to propagate the changes

Replicator Server Allows Change Propagation, It uses OpenAPI.

When Replicator Server receives a database event from the distribution thread(s), it reads from the distribution queue into an in-memory list.

For each row to be replicated, RepServer then executes a database procedure on the remote databases.

RepServer local accesses go through a DBMS server.

Replicator Manager

Allows Configuration, defining databases, tables columns and replication scheme to follow.

Uses Embedded SQL code.

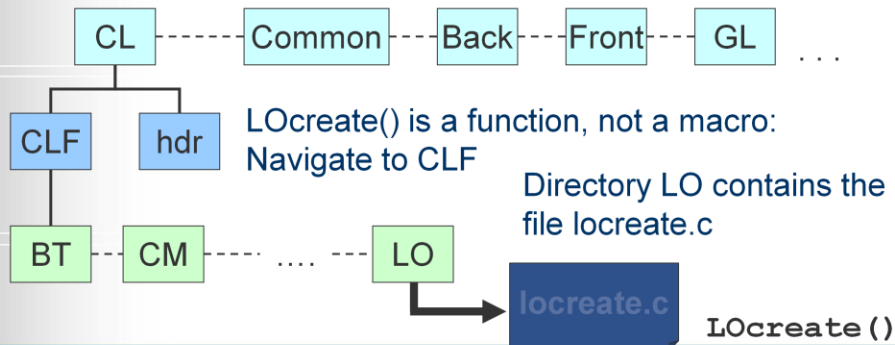
INGRES



Understanding the Ingres Source Structure

Navigating the Source Tree

- Where do I find the LOcreate() function?
- Two letters indicate a CL/GL subdirectory, "LO"



50

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Naming convention eases Ingres source code navigation.

For example searching for CL functions LOcreate() which creates files or directories as specified.

First two letters indicate CL directory where the source code is.

Look under CL/CLF

Go to LO

The function LOcreate() is found under LO directory.

Source Format: File Documentation

```
/*
** Copyright (c) 2004, Computer Associates Intl, Inc.
*/
<#includes>
/**
** Name: <filename> - <short comment>
**
** Description:
**     <comments>. This file defines:
**
**     <proc1> <description>
**     <proc2> <description>
**     <...> <description>
**
** History:
**     <manual history entries>
**
**/
```

Ingres file comment section names

51 © 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies. INGRES

Ingres uses self documenting code. This means that the file and executable interface documentation is generated from special comments in the source code. A number of 'markers' are used to identify documentation comments. The keywords within them are used as section titles in the resulting documentation. The markers and their meanings are illustrated below:

Name: ¹	Name of the file or function
Description: ¹	Detailed description
History: ¹	Ignored for documentation purposes

Notes

¹: These keywords are used by both File and Function comments. The others are used by Function comments only.

Source Format: Function Documentation

```

/*{
** Name: <function name>      [- <short comment> ]
**
** Description:
**     <description text>
**
** Inputs:
**     <param name>           <input value description>
**
** Outputs:
**     <param name>           <output value description>
** Returns:
**     <function return value>
** Exceptions:
**     <exception codes>
**
** Side Effects:

```

Ingres function begin comment marker (points to `/*{`)

Ingres function end comment marker (points to `**`)

52 © 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies. **INGRES**

Comment Markers:

Comment Marker Meaning

<code>/**</code>	File Comment
<code>/*{</code>	Function or Other Comment
<code>**</code> (E re-formatted)	Comment section to be reproduced verbatim (not re-formatted)

Source Format: Function Documentation

```
/*{  
** Name: <function name>    [- <short comment> ]  
**  
** Description:  
**    <description text>  
**  
** Inputs:  
**    <param name>          <input value description>  
**  
** Outputs:  
**    <param name>          <output value description>  
** Returns:  
**    <function return values>  
** Exceptions:  
**    <exception codes>  
**
```

Ingres function comment section names

```
** Side Effects:  
53 **
```

© 2007 Ingres Corporation (Ingres). All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

INGRES

Comment Keywords:

Comment Keyword	Meaning
-----------------	---------

Inputs:

Input parameters

Outputs:

Output parameters

Returns:

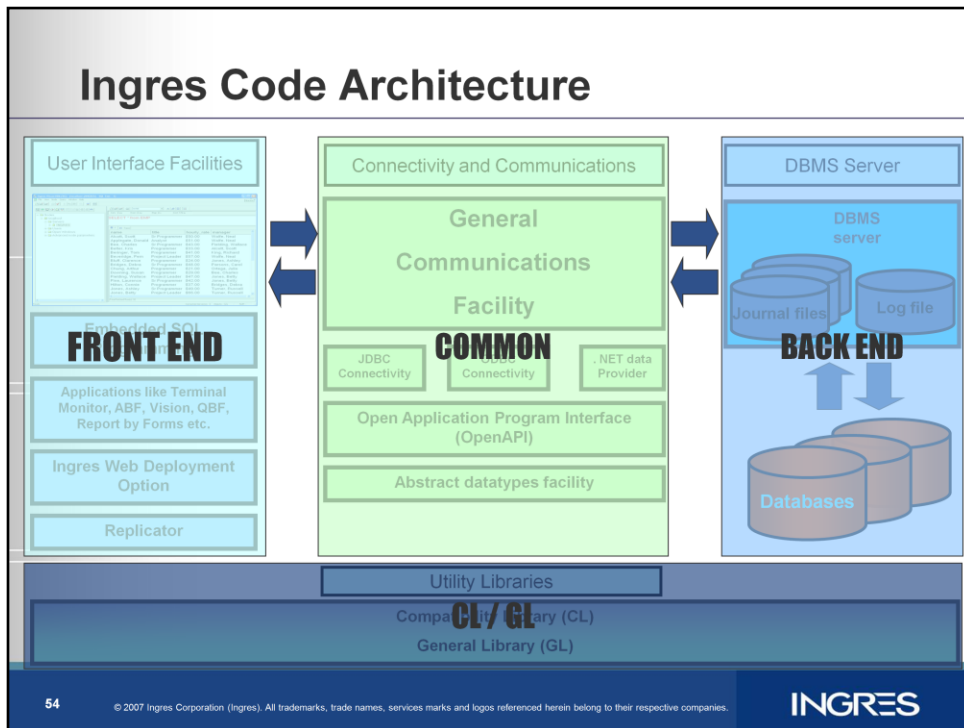
Return Values

Exceptions:

Exception conditions and handlers

See Also:

Reserved



We discussed Ingres Code Architecture and its various components
 We discussed components of Back end, DBMS and how a query is processed by various facilities in DBMS

We discussed the components of CL and GL

We discussed how CL isolated the platform specific code for all facilities for cross platform portability.

We discussed components of Common

The General communication facility that allows mechanism of communicating data between backend and front end

We discussed various components of Front end

The embedded SQL programming for database access

Terminal monitor and other character based applications for database access

Ingres web deployment options for web based access to Dbms

Questions & Answers

```
SELECT question  
FROM audience a, state s  
WHERE a.participant = s.participant  
AND s.state = 'Conscious'
```

I now query for questions. ?