

The emergence of powerful workstations, high bandwidth networks, and cooperative processing based upon open systems has fundamentally altered the environment for information management and corporate computing. Model 204 is one of the most powerful tools available for constructing interactive applications that reference large databases. Yet, Model 204 runs exclusively on the mainframe and embraces a centralized approach to application development.

The Model 204 customer base has amassed a huge inventory of applications and databases. Most customers are satisfied with their Model 204 application suites and they find it easy to add new applications. There is an increasing tendency, however to label these applications as "Legacy Systems" because they run on mainframes and have no apparent relevance to the workstation dominated programming environments of the future. As organizations evolve from SNA networks of 3270 clusters to networks of workstation LAN's, end users are no longer satisfied with yesterday's application interfaces and response times.

Janus is a family of products that provide integration of Model 204 databases and applications in a TCP/IP based client/server environment. With Janus, an organization can capitalize on the power and efficiency of User Language while migrating from a centralized, mainframe-based computing model into a distributed client/server architecture. In addition to general C client support, Janus offers integration of Model 204 in the SYBASE Open Client/Server Architecture.

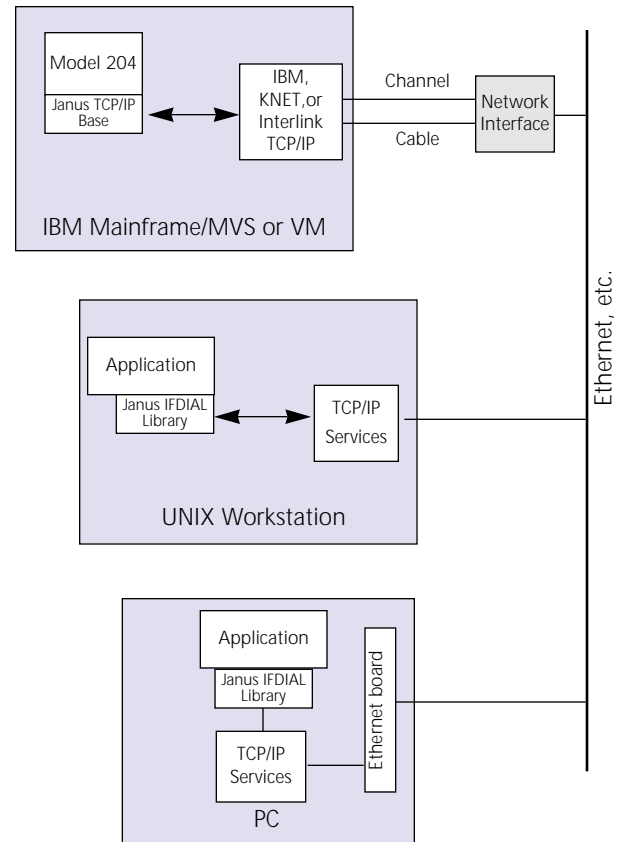
Janus consists of three products: **Janus TCP/IP Base** provides native support within Model 204 for TCP/IP networks, eliminating the need for gateways and SNA bridges. Janus TCP/IP Base also includes a collection of C routines in its IFDIAL Library to be used by workstation programmers to allow client applications to directly pass parameters to, and read and write data from and to Model 204. The **Janus Open Server** allows client applications using the SYBASE Open Client DB Library to invoke services from Model 204. The **Janus Open Client** allows User Language programs to request services from SYBASE SQL Servers and other specialized servers that are built upon the SYBASE Open Server library.

Janus TCP/IP Base

Janus TCP/IP Base provides all the tools necessary to access Model 204 databases and applications directly from workstations. It provides client/server access over TCP/IP networks similar to Model 204 BATCH2-IFDIAL support.

Janus TCP/IP Base consists of three components:

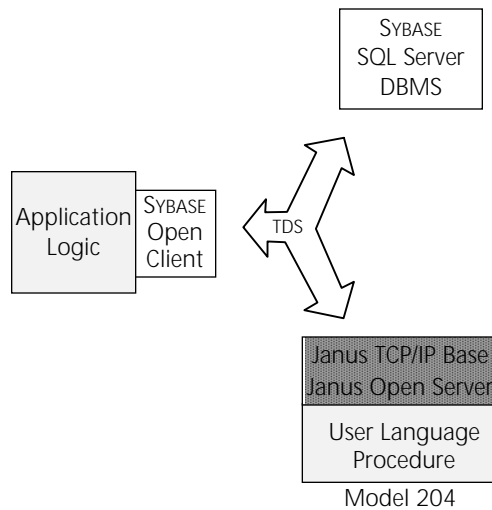
- Object decks that are linked with the Model 204 ON-LINE load module to implement high performance TCP/IP communications support.
- A Model 204 command set for defining and managing server threads.
- A C routine library that enables communications with the Model 204 ONLINE. Some of these routines can be used by programmers to write applications that communicate and exchange data with Model 204, while others can be used to communicate with a Model 204 ONLINE without any additional programming.



Janus Open Server

Janus Open Server allows Model 204 User Language procedures to be invoked by Remote Procedure Calls (RPC) or Transact SQL EXEC statements. There is no difference in the way client applications using SYBASE Open Client Library invoke a stored Model 204 User Language procedure and a stored Transact SQL procedure. The SYBASE client sends the name of a stored procedure and an arbitrary number of parameters, and the Janus Open Server executes that stored procedure and returns the requested data in Model 204 images that appear as "rows" to the client, or as RPC return parameters.

Model 204 as a high performance server in the premiere distributed computing environment.

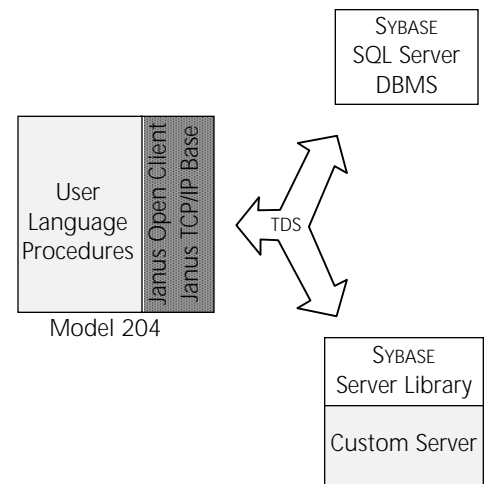


- Precompiled User Language procedures invoked via RPC or Transact SQL EXEC from Open Client
- Janus Open Server stored procedures are plug compatible with SQL Server stored procedures
- \$functions for retrieving RPC parameters and passing back result "rows"

Janus Open Client

Janus Open Client provides a set of \$functions that allow SYBASE Open Client applications to be written in Model 204 User Language. The User Language client applications send RPCs or language requests (e.g. Transact SQL) to a SYBASE SQL Server or other Open Server and retrieve the results. The Janus Open Client \$functions support User Language applications as clients to several different servers simultaneously. In fact, a User Language Open Server application may also act as an Open Client application.

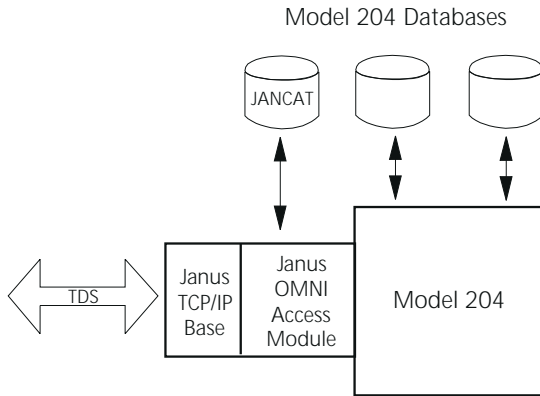
Model 204 User Language Procedures as clients in the SYBASE Open Client/Server Architecture.



- Easy to use interface based on \$functions and images
- Lets Model 204 legacy systems synchronize with other Open Client/Server applications and databases.

Janus OmniSQL Access Module

The SYBASE OMNI SQL Server provides SYBASE client applications with the benefits of distributed programming across a wide variety of heterogeneous data sources. Any client application that works with SQL Server can be used without change against an OMNI SQL Server. Client applications are presented with a unified relational view of the data contained in the various sources as if the data were managed by a single SQL Server.



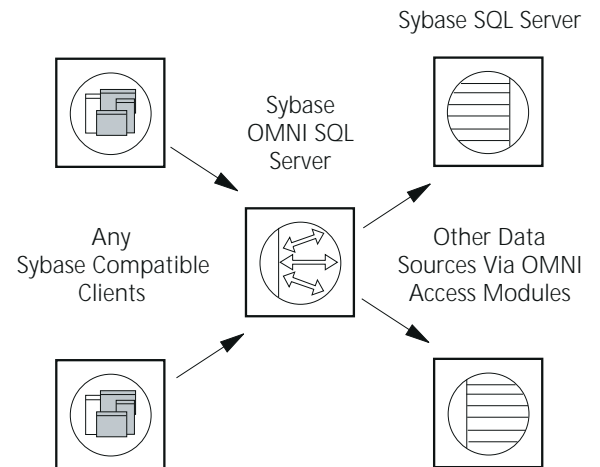
- *Global catalog management*
- *Global request optimization*
- *SQL dialect translation*
- *Heterogeneous database joins*

The Sybase OMNI SQL Server supports the Transact-SQL language, which is a superset of ANSI SQL designed to facilitate stored procedures and Remote Procedure Call (RPC). Clients may use stored procedures, static, or dynamic SQL. The OMNI SQL Server maintains a global catalog comprising metadata from each of the data sources. The OMNI SQL Server decomposes Transact-SQL requests into subrequests to be processed by the appropriate data source. If the data source is not a SQL Server, the subrequests are translated into a subset of ANSI SQL and sent as dynamic language strings to the OMNI SQL Access Module responsible for the data source.

The Janus OMNI SQL Access Module accepts requests from the OMNI SQL Server and parses the dynamic SQL strings. The Janus data catalog, JANCAT, is used to verify the SQL strings and direct their translation into User Language. The resulting User Language code is executed and the results are transmitted back to the OMNI SQL Server. If a source query joins data from multiple sources, for example a SYBASE SQL Server and Model 204, the join is performed at the OMNI SQL Server.

The JANCAT subsystem provides the DBA with a set of easy to use tools for maintaining relational views of Model 204 files. An automated generation mode scans a few hundred records of an indicated Model 204 file and makes a best guess of the record types, nested tables, and so on. Simple editing tools can then be used to refine the relational view.

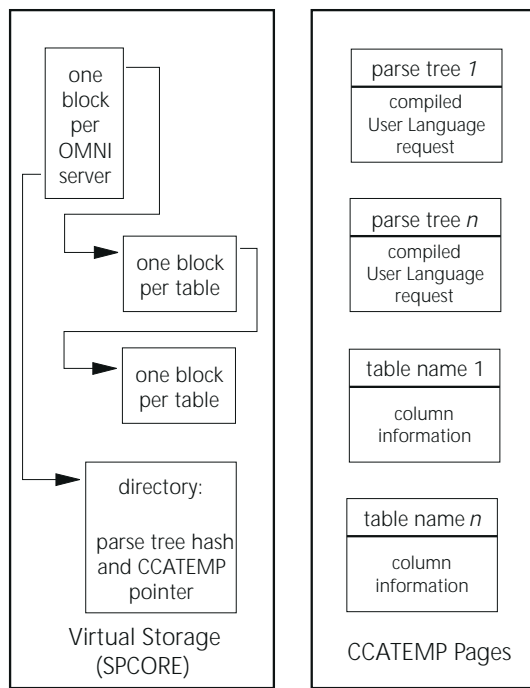
JANCAT has very few restrictions on the types of Model 204 files and groups that can be mapped. Multiple record types within a single file or group are supported as multiple tables. Individual tables may be defined as read-only. Nested tables need not possess a primary key field. A multiply occurring field can be defined as multiple columns instead of nested tables.



- *OMNI SQL Server sends language request with subset of ANSI SQL*
- *Janus OMNISQL Access Module replies with results data stream*

The Janus OMNI SQL Server has been highly optimized for processing repetitive dynamic requests, of the kind frequently produced by application generators and query tools. When a Janus server with type "OMNI" is started, a set of control blocks is allocated in above the line storage within the Model 204 region and primed with information from JANCAT. This information includes just the most basic information about each table in the relational view and a directory of compiled requests.

When a language request is received from an OMNI SQL Server, the information required to parse the request is completely resident in virtual storage. This means that no I/O is required and very little CPU is consumed to validate a request. Once a valid parse tree has been built the tree is hashed and looked up in the resident request directory. If the request has already been compiled and saved, its pre-compiled form is loaded from CCATEMP and its LRU queue position is updated. Due to the small and highly specialized nature of the compiled OMNI User Language requests, this load is extremely efficient.



Summary

In response to the explosion in desktop computing power, many organizations have stated a goal of no further increase in mainframe capacity. Their objective is to stabilize mainframe applications and to develop new applications on networked workstation environments. Janus uniquely satisfies this objective:

- Janus provides industrial strength client/server access to Model 204 applications, making it possible to offload the presentation services and significant additional portions of application logic. This can reduce mainframe CPU consumption by up to 50%, providing increased capacity without costly mainframe upgrades.
- Janus greatly simplifies the task of coexistence and provides for an orderly migration of application development. Workstation application programmers can use the Janus IFDIAL Library to access Model 204, or the SYBASE Open Client DB-Library to transparently access data managed by either Model 204 or SYBASE SQL Server. This dramatically reduces the need for cross training and allows for the most efficient deployment of both personnel and hardware resources.

If the request is not found, it is translated into User Language and compiled. The parse tree and compiled code are then saved in a CCATEMP page and the directory is updated. If the maximum number of requests have already been saved, the compilation and parse tree for the least recently referenced request is discarded.

The User Language requests created by the Janus OMNI SQL Access Module make use of Janus dollar functions to accept parameters and return results. Compilations for requests that include terms of the form:

<column name> <operator> value

can be shared even when the various requests provide different values. The request is compiled with an invocation of an initialization dollar function that sets the percent variable used to represent the value. The combination of these optimizations and the sharing of compilations results in a dramatic reduction in overhead. Simple SELECT, UPDATE, and INSERT statements can be processed with only 10% to 15% CPU penalty over pre-compiled User Language.

Janus runs under Model 204 version 2.1 and later, and is available under MVS and VM/CMS operating systems.