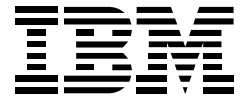


S/390 Division



OS/390 Guide to e-business Connectors: Second Edition

December 1999

This second edition of the e-business Connectors Guide provides more guidance on performance expectations. This white paper is available at <http://www.s390.ibm.com/ebusiness/connectors/>. Both technical explanations and customer examples are used to give performance advice. This edition supersedes the prior edition. We expect to update this white paper periodically. For a higher level business view, please refer to the *OS/390 Web-Enablement Overview*, GF22-5138 paper.

Table of Contents

Overview	2
Base Software Considerations:	3
"2-Tier" versus "3-Tier" Considerations	4
Web Enablement Performance Overview	5
Application Model Considerations for Performance	5
Language Choice Considerations for Performance	6
Server Scaling: WLM, Parallel Sysplex	7
OS/390 Release Considerations for Performance	7
IBM Global Services, e-business services	8
IBM Internal Performance Experiences	8
Customer Scenarios	9
CICS	9
IMS	10
DB2	12
HOD Customer reference example	12
Connector details and ordering	12
Second Generation (Java) Connectors	13
Additional Information on Connectors	15
Appendix A: Connector Matrix	16

Goal: Help customers choose how to access existing data and applications to meet their e-business objectives with the appropriate expectations.

Audience: This white paper is intended for technical personnel (architects, systems programmers, etc.).

Frequency: The intent is to update this paper three times per year

Overview:

The objective of this white paper is to give OS/390® customers advice on how to choose the e-business connectors for their existing subsystem (CICS®, DB2®, IMS™, and MQSeries®) applications and data. “Connectors” in this paper refers to the software programs that connect a Web application or browser to an existing application and data. The key considerations for the e-business application being architected include: performance requirements (e.g. transactions per second), security requirements (e.g. SSL and its associated performance), “2-Tier” versus “3-Tier” considerations, existing levels of OS/390 and the subsystems (including whether an upgrade can be considered as part of the total solution), and the programming skills available for this solution (which could include contracted services). Some customer scenarios are provided as examples.

This white paper focuses on the “2-Tier” solution where the Web server, Web application server and existing application and data reside on an OS/390 server. The WebSphere™ Common Connector Framework (CCF) connectors are designed to support development and deployment on middle tier servers and portability to OS/390 without change. The deployment may benefit from additional local optimizations. S/390’s qualities of service are the added value of the deployment on OS/390. IBM has worked closely with JavaSoft™ and the industry to define a “Common Connector Framework.” An official JavaSoft draft review specification drawn from this collaboration is expected shortly on their Web site.

The connectors have been categorized into three groups. A Web-based tool (*The OS/390 Planning Assistant for e-business*, www.s390.ibm.com/os390/wizards/ebiz) has been developed to help guide you through the decision process for picking the right connectors based on the criteria outlined in this white paper.

First Generation (API) connectors which have been available for some time, are characterized as being very quick to implement with existing programming skills. These include the API-based connections using CICS Web Support (CWS), IMS TCP/IP OTMA Connection

(IMS TOC), Net.Data®, and CWS with 3270 Bridge and Templates. API connectors tend to have less standard tooling, less portability and typically better performance than Server-side Java connectors. They are less focused on application re-engineering or technology upgrade and more on optimizing Web enablement to their existing application and data.

Second Generation (Java) connectors are characterized by the use of Java™ and tooling. For relational databases, JDBC and SQLJ are provided. CICS and IMS have developed respectively the CICS Transaction Gateway (CTG) and the IMS Connector for Java based on the IBM developed Common Connector Framework (CCF). This framework is supported by the Java application development tools of WebSphere, specifically VisualAge® for Java. Host on-Demand is another solution for 3270-based applications. These connectors are generally available now.

Second Generation connectors are more focused on standards, tooling, portability and beginning the process of application re-engineering. Performance/scalability is currently below that of corresponding First Generation counterparts (e.g. CTG vs. CWS, IMS TOC vs. IMS Connector for Java, JDBC vs. Net.Data).

Third Generation Enterprise Java connectors introduces Enterprise JavaBeans™ (EJBs) and the “Java 2” family of protocols (e.g. JMS, JDBC 2.0,) which will be introduced in 2000. The third generation connectors are an evolution of second generation with enhanced transactional and security context at the thread level, and data persistence built into the application programming model. Even more of the S/390 platform’s QoS (Qualities of Service) are provided to the application transparently. Examples of these key QoS advantages are Workload Management for Address Space management, RRS (Recoverable Resource Services) for distributed transactions and Parallel Sysplex® for data sharing.

The connectors emphasized in this report are the Java based Common Connector Framework (CCF) connectors, namely the CICS Transaction Gateway (CTG) and the IMS TCP/IP OTMA Connection (TOC) Connector for Java (IMS Connector for Java). The IMS Connector for Java uses IMS TOC. The CCF Connectors are part of IBM’s Application Framework for e-business that is being used to focus our e-business activities. We will also discuss the relational database access connectors, JDBC and SQLJ, which are the strategic way to access both IBM and non-IBM relational data. In addition, we will cover a number of other available connectors that are being used by some current customers. This is not intended to be a complete

list of connectors but a guide to provide input on those connectors for which IBM perceives the most interest and/or value.

The IBM WebSphere family of products is a very powerful set of facilities for developing Web-enabled applications that combine a broad set of industry standard programming interfaces (e.g. Java servlets and Enterprise Java Beans [EJBs]) with a set of application development and life cycle management tools to enhance the programmer's ability to rapidly compose business objects into e-business applications for deployment across multiple server platforms. With the delivery of the CICS Transaction Server and the Component Broker technology delivered as part of WebSphere Enterprise Edition on OS/390, EJBs will become the preferred connector technology. Combined with rapidly improving Java performance across the platforms, IBM believes solutions based on Java technology will quickly become the best total solution for enterprises requiring access to existing data and applications. As a result, many of our performance measurements and supporting services (e.g. UNIX System Services, TCP/IP, JDK) are being focused on making this environment best of breed.

The CCF connector support is delivered in three parts:

1. VisualAge for Java provides the base connector classes,
2. The WebSphere Application Server for OS/390 provides the optimized OS/390 classes, and
3. The connectors themselves (CTG, IMS Connector for Java, etc.).

Base Software Considerations:

For a variety of reasons, current S/390® customers may need to consider tactical alternatives to full-scale WebSphere development and deployment (e.g. because of time delays in updating base software).

You have several considerations to make when choosing how to Web-enable. The first consideration is the subsystem (e.g. CICS, DB2, IMS, MQSeries) you need to connect to and what subsystem level you are currently running. For example, some of the Generation 2 connectors are not supported on older versions of CICS. Another consideration is the level of business application function implemented in the subsystem and the need for new application function to create a Web-ready application.

Other business considerations (e.g. Y2K, application programmer skills) may not permit you to upgrade the

subsystem to the most current level in time to establish a presence on the Internet and providing connectivity to production applications.

There are a number of ways to address this, such as running newer levels of software in a separate logical partition (LPAR) along with the IBM HTTP Server, and if needed, the WebSphere Application Server. CICS Transaction Server is particularly good in this area and offers a number of methods of interconnecting systems. There is also a services offering NetCICS specifically aimed at Web enabling older CICS applications. Other potential solutions might include using MQSeries between LPARs.

You should consider implementing new business function in the WebSphere Java environment while re-using existing business function in the subsystem environment. The e-business connectors facilitate this by enabling Java applications to access applications and data in the subsystem environments. In general, the level of the subsystem is not a major performance consideration as it relates to the connection to the Web, but it may influence or limit the specific connectors that may be selected. Assessment of the performance / scalability of a specific connector given the application model (e.g. 3270 transactions) as well as a view on migration (e.g. EJBs) are key factors to be considered before selecting the best connector strategy to use.

The next thing to consider is the level of operating system in production now (and the next level planned, and when). In addition to functional dependencies that may require a certain level of OS/390, there are performance considerations of the underlying components (TCP/IP, Web server, Java, etc.) to factor in when selecting the best connector. One example would be the level of security and related performance required. OS/390 V2 R6 with the IBM HTTP Server for OS/390 (previously named the Lotus® Domino™ Go Webserver) has enabled the use of hardware encryption (over software-based encryption) to improve SSL performance. Another factor to consider is the use of the Workload Manager (WLM) to improve the scalability of some of the connector solutions. Some connectors are limited by storage, threads, or back-end connections to the number of connections that can be supported in a single address space. Using the WebSphere Application Server in scalable mode enables these applications to scale to significantly higher transaction rates. WAS 1.2 is strongly recommended for scalable WLM mode. Your performance and security attributes will be related to the level of the operating system on which you plan to deploy the connector.

"2-Tier" versus "3-Tier" Considerations:

Many customers start out in the Web environment by simple page serving on a middle tier platform (UNIX® or Windows NT®). As their requirements grow, such as connecting to back-end transactions or increased security concerns, the requirement for S/390's qualities of service becomes greater.

While most Web applications are inherently 3-tier applications, consisting of presentation, Web logic+connectors, and business logic, for the purposes of this document, 2-tier vs. 3-tier is a discussion about hardware server tiers, not application tiers.

Some of the connectors are very portable (e.g. CCF) across multiple WebSphere platforms. Others that use distributed connections to the back-end subsystems have different interfaces when they run natively on OS/390 versus running on a middle tier server. With WebSphere, you have a choice of 2-tier or 3-tier configurations for deployment, but the application development and operational management characteristics of these configurations may be significantly different. Some multi-tier connector configurations may require both TCP/IP and SNA to be defined. In addition, there are some security considerations to take into account when deploying on a middle tier server. In the 3-tier model, security information must be configured on the middle tier and in some cases, security credentials must be communicated between the middle tier and S/390. Without physical network security or encryption of the connection, information will be transmitted in the clear.

The 2-tier WebSphere model provides for development of Java applications using VisualAge tools, simplified operational management within a single environment, improved scalability, and the mapping of security credentials by the Web server (working with the OS/390 Security Server) to the execution thread of the connector. The actual user ID/password required by the subsystem may require a different flow, but this is contained within the OS/390 environment. SSL communication between the browser and the Web server on OS/390 permits all client communication to be secure. The network protocol between the client and the server is all TCP/IP in the 2-tier model.

Three-Tier solutions can be very effective solutions. They don't require a change to existing subsystem or OS/390 levels. They leverage a variety of Web servers and the applications they support. They provide scalability and/or more cost effective solutions for functions which do not require OS/390 qualities of service (QoS) such as Caching or CPU-intensive Java translation of data formats. The key point is

development of new application business logic, especially when transactions or mission critical data access are involved, should be deployed on the platform where the required QoS are found.

Skills Assessment

Once you understand what connector options exist for you, along with relative performance levels, you have to consider what programming skills exist in your shop and what tools exist to assist you to develop the required support. If you are at an OS/390 and subsystem level where WebSphere and Java perform well and you have Java skills, using of VisualAge for Java with the Common Connector Framework should be the preferred direction. These are the configurations for which IBM will produce performance reports with frequent updates to assist in projecting throughput and assessing capacity planning. The key decision point is whether the current and/or expected performance improvements over time (including new hardware) will keep pace with your performance requirements. If you are not at a required level (OS/390 or subsystem—see tables on page 16) or your performance requirements are already beyond the Java connector performance capabilities, then you need to assess your options and skill requirements (e.g. develop/use GWAPI (Domino Go Webserver API) connectors in COBOL or C). If possible, look for connectors with application development tools to facilitate development as well as connectors that will run on multiple versions of the subsystem.

Web Enablement Performance Overview

Many customers approach Web enablement of existing transactions and data in an iterative fashion. They first look for the simplest method that doesn't require much investment or change to the existing infrastructure, including the operating system, the subsystem or the application itself. Tooling is usually the preferred route where the skills of the person required are more widely available. Security is almost never a concern initially although it becomes an issue when the new application is prepared to go online. Performance and scalability are usually the last items considered and, unfortunately, most often only when they become a problem.

The main concern for most enterprise customers looking for business advantage from expanding application access to the Internet, or using Internet technology to re-engineer their business is the ability for those applications to scale as demand increases, as this is usually the motivation for investing in Web enablement. The ability to add hardware and realize proportional gains in throughput commensurate with that investment is a fundamental expectation and a traditional strength of OS/390 subsystems (e.g. CICS, IMS, DB2). This includes scaling over a Parallel Sysplex configuration when the volume of transactions and/or data exceed the ability of one system to handle the load. This is especially true in e-business, where capacity peaks are typically significantly greater than in traditional intranet environments.

The next performance related concern to raise is the efficiency of the new connections, as they require additional capacity over that required for existing intranet connections. This is required for the additional layers of middleware (security, data translation, protocol conversion, etc.) needed to complete an end-to-end transaction, where the client is usually a Web browser. There are different models in the industry of how these connections are provided and they all have different performance characteristics which need to be understood further. The net effect is that the additional cost (hardware and software) must be competitive with other alternatives. Two-tier solutions, especially when a large amount of data is involved, tend to be more efficient (and somewhat more secure) than distributed, multi-tier solutions. However, there are cases when the interactions and data interchange are minimized and the middle tier(s) are acting as simple dispatchers, where they could be more cost effective without sacrificing data integrity (e.g. no intermediate session state data which must be recovered on failure). The client protocols (e.g. HTTP, IIOP, RMI, 3270, etc.), the Web application server

model (CGI, ASP, JSP/servlet, EJB, etc.) and the language choice (C, C++, Java, PERL, etc.) all have an impact on overall efficiency.

Additional factors affect the choice of application model and platform selection, such as availability, application/data integration, and security/integrity.. They are not addressed specifically in this version of the white paper.

Application Model Considerations for Performance

CGIs, ASPs and JSPs are three ways of generating dynamic Web pages.

1. **CGI (Common Gateway Interface)** programs are run on a Web server in a separate process from the Web server. Each request for a CGI starts up a process, gets environment variables from the server, runs and generates a response page, sends the page to the client and then terminates the CGI process. Because each CGI request starts up a separate process and then kills it after completion, CGIs are very slow.

There are several issues concerning CGIs. The first is the process create/destroy. FastCGI tries to eliminate this by having persistent CGI processes, so that the running CGI process hangs around after the call to be used again.

The other issue is that most sites generating dynamic content want a persistent infrastructure such as session support and process management that is not available with CGI. The benefit of CGI is its simplicity, but that is also its problem. To support session, application isolation, monitoring of memory usage etc. all of your applications would have to be written to support a common set of APIs and then this infrastructure, accessible through those APIs, would have to be implemented by each installation. The servlet APIs and Java technology provide this infrastructure where the customer does not have to reinvent the wheel for every deployment.

The reason one does not hear much about FastCGIs is that while they address the first problem, they do not provide the infrastructure described above.

FastCGI offers a compromise between the flexibility of CGI and the efficiency of plug-ins. FastCGI was invented by Open Market and both Domino Go WebServer and the IBM HTTP Server have been modified to support FastCGI. Unlike traditional CGI, FastCGI programs must be written in C and linked with the C routines provided by Open Market at <http://www.fastcgi.com>.

Plug-ins. In addition to CGIs and servlets, many Web servers support plug-ins or DLLs which are invoked and run in the Web server process itself, usually on the same thread of execution. Netscape has a version called NSAPI, Microsoft supports ISAPI and IBM supports a flavor called either ICAPI or GWAPI (different names for same function).

These plug-ins tend to perform the best of all options but are proprietary to the Web server that supports them and have become less accepted by application vendors. Tooling is also usually limited

2. ASP (Active Server Pages) are processed by the Microsoft® Internet Information Server (IIS) Web Server on Windows and by third party (ChiliSoft) ASP processors on UNIX systems. There are ASP specific tags that are embedded in static HTML files. The ASP processor strips out these tags and generates dynamic content depending on the type of tag. These tags can be used for simple server side include information like date, time, or request information, or can trigger complex operations such as database access, and access to COM objects. Currently in IIS, ASP processing can be done in-process (within the Web servers process) or out of process (in its own process). Because the Web server knows about ASP pages natively it can do performance optimizations like partial page caching.

3. JSP (Java Server Pages) were initially an attempt to embed Java into static HTML pages, making the Java language like a scripting language for dynamic Web pages. This effort has matured to supporting both embedded Java and JSP tags (much like ASP) to support dynamic content. In WebSphere AppServer, and all other JSP supported Web/App Servers, when a request is made for a JSP page, the page is processed and converted into a dynamically constructed servlet. This servlet is compiled and loaded into the server much like any other servlet. This processing is called page compilation. The dynamically constructed servlet is called implicitly from then on, unless the JSP file has been changed on disk, in which case the JSP is re-page compiled.

There are 2 basic models for using JSP:

1. An ASP-like model where a request is made for the JSP page and the page compiled servlet is run with dynamic content filled in.
2. A servlet to JSP model, where the client calls a servlet which generates the dynamic content and then calls a JSP page to be sent in response to the client request. This JSP page gets the dynamic content from the original servlet and then sends the response to the client. This has the benefit of having the creator of dynamic content (the called servlet) be independent of the user of dynamic content (JSP page).

Usually JSP pages will be slower than ASP pages since Web server optimizations cannot be made, and much of the processing is done in Java instead of natively as in ASP. The advantage of JSPs is that they are Web server and platform independent.

Language Choice Considerations for Performance

Most existing CICS/IMS applications are written in COBOL. Most CGI/plugin applications/gateways are written in C. As a result, they experience the additional overhead of Language Environment® (LE) transitioning between the two languages. Since they are also Posix (Pthread) applications, they have the limitations on the number of threads that can run in a process based on some below the line storage limits. There is also character code page translation between ASCII and EBCDIC for the Web to existing transaction model.

The newer applications, generated by tooling, use Java language, which presents a number of challenges. The first is that Java is an interpreted language, which requires translation of the byte codes as opposed to being compiled prior to execution. There exists a Just In Time (JIT) compiler, which will compile the Java code after a number of accesses to the classes. Java's native code page is Unicode, which creates additional overhead to bridge the environments. Another issue is linkage, where object oriented code typically creates more levels of nesting (methods call methods which call methods.) Then there's the issue of garbage collection, which is invoked periodically to deal with storage within the process. S/390 has enhanced the standard Java Garbage Collection by not requiring the process to be suspended while it's running. There's also the overhead to initialize the JVM, which is experienced on each process initialization. Servlets running on WebSphere for OS/390 do not suffer much from JVM initialization as OS/390 runs them in-process. There are a number of

optimizations being created within Java and the supporting environment to reduce these overheads to make Java much closer to the current languages. However, for the same function, Java can be viewed currently as at least 2x the cost of the procedural languages, such as C or COBOL

Server Scaling: WLM, Parallel Sysplex

The existing transaction systems (such as CICS and IMS) have consistently scaled with the largest processors built (full SMP) and more recently, with data sharing, across the Parallel Sysplex with a predictable increase in throughput proportional with the increase in CPU power. The newer application environments are experiencing limitations on all platforms based on technologies that are not yet mature enough to reach the same levels of scaling as the systems they are connecting to. TCP/IP stacks, including OS/390 as of V2 R5, typically scale well on most platforms up to at least 10 processors (4 processors on NT). Web servers tend to scale well for static pages (although the file systems may experience some constraints when update ability is required within a cluster (where replication may be required). However, when they are accessing relational data bases (e.g. JDBC) or they are running other programs (CGIs, servlets/JSPs), there are some overheads (process create/destroy, affinities, threading, context switching, etc.) that may limit scaling.

S/390 has some technologies that can assist these newer types of applications to scale to higher rates without requiring major changes to the applications themselves. The first is WLM, which can replicate application environments to provide expansion of the number of threads/processes without extra definitions or application awareness. It (WLM) can be used to manage address spaces and provide increased scalability when there are limits on the number of threads or connections to backend systems. CICS has an implementation defined limit of 100 concurrent EXCI connections per address space. This is intended to limit opportunities for “denial of service” attacks.

Another technology in OS/390 that facilitates server scaling is our Parallel Sysplex and data sharing support, such that applications can run on any processor without concern for data locality. Replicating Web server processes across a system or across the sysplex enables more transactions across a single logical application image. Use of session state in servlets creates process and/or processor affinities which may affect scalability. There are several schemes to help address this (shared memory, use of shared data within a cluster, etc.) but

they add additional overhead. OS/390 stores session state in DB2, which is sharable across processes and processors. Network load distribution technologies (like the IBM SecureWay Network Dispatcher or the Cisco® Local Director) enable this horizontal scaling with a single network definition (IP address) to all clients.

Another technology that OS/390 deploys in this space (which exists on multiple UNIX platforms) is “async I/O.” As opposed to Select, which incurs more overhead as the number of sockets increases, async I/O does not experience large system effects and requires less overhead on a per connection basis. The latest OS/390 Web server (as well as other application environments like Domino, ADSM, EJBs,) utilizes this support. This is of particular value for functions like SSL and persistent connections, where the life of the transaction is longer.

OS/390 Release Considerations for Performance

TCP/IP is obviously a major factor in the scalability and CPU efficiency of Web-based applications. TCP/IP was totally re-engineered for high performance and scalability in OS/390 V2R5, which makes it the base recommendation for deploying high volume Web solutions. Gigabit Ethernet with QDIO in OS/390 V2R7 greatly improves throughput, reduces latency and pathlength and increases the number of connections through a single connection by multiple factors (2-5) over other existing options. The Java based connectors will see significant improvements in performance on the latest level of the JDK (1.1.8). These are available on OS/390 releases (V1R3 thru V2R8).

The Language Environment (LE) continues to make improvements in the amount of storage (including below the line) used for each thread, so C and Java based connectors will benefit from later releases of OS/390. The Web server provides significant performance improvements for static Web serving in V2R7 through use of the Fast Response Cache Accelerator (FRCA) cache. The Web Application Server 1.2 has addressed some issues working with the WebSphere Application Server in scalable mode, which runs on OS/390 V2R5 and above. The V2R8 Web Server has improved support for scalable mode, including persistent connections and SSL support.

For those applications that use SSL, the Lotus Domino Go Webserver (renamed to the IBM HTTP Server for OS/390 in OS/390 V2R6) has been enhanced to work with hardware crypto, which offers a dramatic CPU reduction, in OS/390 V2R6 release. This reduction is achieved by off loading the software encryption instructions from the general purpose processors to a separate crypto coprocessor. This can be up to a tenfold improvement in CPU utilization for SSL handshakes (public key operations).

IBM Global Services, e-business services

The IBM S/390 Solution for e-business (ebSO): The IBM S/390 Solution for e-business allows you to introduce e-business without having to upgrade your current production environment on S/390. It can provide the skilled individuals that can allow you to take advantage of new e-business technologies such as: Web serving and integration with the WebSphere Application Server Enterprise Edition for OS/390; collaboration with Lotus Domino for S/390, and e-commerce with Net.Commerce for OS/390. All of these are enhanced with underlying technologies specific to S/390, such as Java for OS/390, Firewall Technologies for OS/390 and the RACF® function of the OS/390 Security Server. This solution can rapidly provide a scalable and secure e-business environment that can position you for growth well into the 21st Century.

To learn more visit the S/390 World Wide Web site at <http://www.s390.ibm.com/> or call IBM Direct at 1 800 IBM-CALL in the U.S. and Canada.

IBM Internal Performance Experiences

IBM has been conducting extensive tests (with a mix of customer and internal benchmarks) recently to understand the performance characteristics of the following key Web enablement environments: access to CICS and IMS transactions via the CCF CICS Transaction Gateway and the CCF IMS Connector for Java which utilizes VisualAge for Java to generate the servlet or JSP to provide access of these existing applications to standard Web browsers, and JDBC (or SQL-J) access to DB2, which also utilizes tooling to generate data beans for access to existing data.

Although this analysis is ongoing with continuing performance improvements, there are some conclusions that can be drawn from the work so far. Data for specific capacity planning should be available in the first half of 2000 to assist customers in determining how

much additional capacity is required to support this type of access. In addition, the levels to which these applications can be scaled will be provided, although there are numerous factors which will affect individual customer experiences.

For the CTG environment, there are some significant enhancements in VA Java V3.0 which optimize the generated code to be more efficient when using the Enterprise Access Builder (EAB) to translate from the COBOL copy books to Java servlets/JSPs. There are also significant improvements in moving to the latest PTF level of JDK 1.1.8.

Another factor to consider with CTG is the applet model vs. the servlet model. Current benchmarks have been run internally with over 250 transactions per second being achieved for the servlet model in certain configurations. We have some enhancements planned (such as CICS connection pooling) that will improve this even more in the not too distant future.

Summary of the CICS related improvements

- CICS Transaction Gateway (applet model) has improved performance 70% with CTG 3.1
- CICS Transaction Gateway (servlet model) has improved performance 20% with CTG 3.1
- CCF Enterprise Access Builder has improved performance >85% with VA-Java 3.0
- CICS TS 1.3 will improve performance 80% with 1Q2000 PTF release.

The performance of Web-enabled CICS transactions using the CCF/CTG connectors has improved by approximately ten times since the function first became available. These improvements have mostly come from changes to the JDK, VAJava, and the CTG. There are a number of improvements in the development pipeline which will continue this trend. The current costs of CCF/CTG are fairly consistent and independent of the complexity of the transaction with the exception of the COMMAREA size.

COMMAREA Unicode/EBCDIC conversion cost will increase with larger COMMAREAS. The average CICS transaction can be enabled with an approximate 100% increase in processor costs for COMMAREA sizes less than 512 bytes. High volume trivial transactions will experience greater relative costs and more complex transactions will experience less than a 100% increase. For high volume trivial transactions with critical response time criteria it may be worth handcoding the Java connections bypassing EAB Record Builder and the CCF connectors. This involves more knowledge on the

part of the developer, but is not difficult for anyone experienced with Java.

For the JDBC environment, the newest version provides around a 85% improvement over the previous version. There is an enhancement planned to address some performance problems running multi-context which should provide even better performance. Internal performance tests indicate that 50 transactions per second can be reasonably supported. The Java Native Interface (JNI) is the largest component affecting performance on all platforms.

Today the 3-tier model scales better and consumes less CPU resource on the 390. That's because the DRDA[®] overhead is less than the Java overhead. If you decide to use stored procedures for the SQL-intensive work (which we always recommend for anything that is high volume), you get excellent CPU cost and scalability (assuming you write the stored procedure in Cobol or C). Focus is to keep the business logic on 390 for best reliability and scalability.

We'll be publishing the SQLJ and JDBC numbers for both our old and new drivers. The goal here is two-fold:

1. Let people know that we're improving the drivers significantly to aid capacity planning.
2. Quantify the performance benefit of SQLJ over JDBC.

Customers requiring high volume Web access to DB2 in production today should look at Net.Data to ensure satisfaction.

IMS TOC V2.1.3 performance data was collected in an OS/390 V2R6 and IMS V6.1 environment. The experiment was run with one, two and three IMS TOC address spaces. The host server processor (9672-RX5) on which IMS and IMS TOC resided reached its maximum CPU capacity (90+% busy) with three IMS TOC address spaces, resulting in 675 transactions per second.

For IMS using Java as a connector, some similar patterns have been observed and described in the CTG section, including the improvements within VisualAge for Java in 3.0. We have also seen significant differences between the JSP and base servlet model with JSPs more expensive in terms of pathlength. This difference is being investigated and is expected to be significantly reduced in the future. Work is ongoing to improve this, including looking at eliminating an additional address space crossing in the OS/390 version.

All performance results are based on tests that were done in a controlled environment. Results you might see will vary based on your actual system configuration and other factors.

Customer Scenarios

This section describes the customer configurations where they are achieving satisfactory performance with one of the e-business connector options. We hope this information will be helpful for you in selecting an alternative. Whenever possible, we have attempted to include OS/390, subsystem and other relevant product levels as well as rough performance/capacity information.

CICS

CTG (CICS Transaction Gateway) Customer reference example:

A customer in Europe is using CTG with the servlet model to drive CICS TS 1.2 with CTG 3.0.2 on OS/390 V2 R6.

Rabobank in Europe is using CTG with the applet model to drive CICS TS 1.2.A Java applet, containing the application and infrastructural components (e.g. IBM CICS JavaBeans) is served (pushed) from the OS/390 Domino Go Webserver to the local proxies. Local clients start the client applet within the browser and JVM. A direct connection is then started between the Java applet and the CTG / CICS Transaction Gateway on the OS/390 (no middle tier !). The gateway then transfers the Java request as an external call interface (EXCI) request to CICS TS1.2, and the appropriate COBOL transaction is started.

The applet was built using VisualAge for Java 2.1 enterprise edition. The entire applet operates stateless. The back-end application was built using IBM VisualAge COBOL 2.2 enterprise edition. The back end transactions were developed using a Component Based Development approach where the application layer was separated from the business layer, which was separated from the data layer. These last two are stateless as well. This makes it possible to reuse the built business layer transactions, as well as address the business transactions from any platform. Initial production tests show that the gateway is stable and performs well. One entire transaction from front-to-back-to-front takes about two seconds, with just a fraction of that spent on the S/390.

To reduce data flow over the lines, they implemented the standard Java zip functionality in the minimal compression configuration. The performance of the JVM on OS/390 is one of the limiting factors. We expect (and have seen) that the current setup should be able to handle around 10 transactions per second without the CTG taking up an excessive amount of mainframe resources and stifling other critical processes. In general it can be said that the performance of the JVM on S/390 is the critical factor. This will get better in future releases.

CWS (CICS Web Support) Customer reference example:

The third largest health insurance in France wanted to improve business efficiency, open up access to its 7000 independent channel partners, reduce operating costs and find new revenue opportunities. The solution needed to be simple:

- Exploit existing technology (CICS)
- 24x7 availability
- Complete security and robustness
- Short project time
- CICS WEB SUPPORT to CICS ESA 4.1 running on a G5 easily handles the customer requirements of
- 7000 concurrent users
- 35,000 transactions per hour
- with ample room for business growth.

Please see the following Web sites for more CICS customer examples including a sysplex example.
<http://www.software.ibm.com/casestudies/swcs.nsf/customername/E3B6A022F22F9CE20025678B000659B9>
(CWS - Health Ins Corp)

<http://www.software.ibm.com/solutions/internet/g325-1336-00.pdf> (CWS - Den Danske Bank)

IMS

IMS TCP/IP OTMA Connection (IMS TOC) Customer reference example:

Another French insurance company has undergone a major change in company structure, having merged with two other large companies. This, in turn, has required AGF to make radical changes in the way it processes transactions using IMS.

IMS TCP/IP OTMA Connection (IMS TOC) was recently implemented to replace a homemade APPC middleware package that became incapable of handling the company's growth. The new group comprises 32,600 employees throughout the world: 11,800 in international branches and 20,800 in France.

On the company's production system, the IMS online activity is around 1.8 million transactions per day. The profile of a transaction is around 35 DL/I calls and 5 DB2 calls. Today, the client/server architecture is based on IMS TCP/IP OTMA Connection (IMS TOC). The network is full TCP/IP. Today, 40% of online activity is processed through IMS TOC, and works as follows:

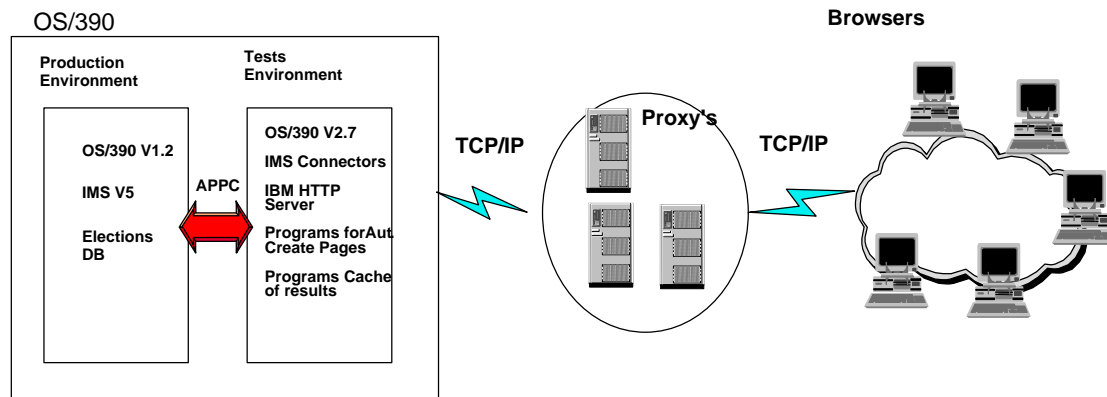
On the client workstation under Windows 95 or Windows NT, the client/server layer is now using the socket API. On the mainframe, IMS TOC has replaced the client/server gateway. IMS TOC passes the request to IMS through OTMA and returns the output message to the TCP/IP client. The IMS TOC exit has been enhanced to perform some security related tasks. For some revamping needs, the company uses IBM's SecureWay Host On-Demand solution in conjunction with ResQ!Net. ResQ!Net for Host On-Demand provides a thin Java applet that works as an add-on with IBM Host On-Demand (HOD) to graphically present host applications. An alternative approach is to develop the GUI using VisualAge for Java and to access the old 3270 transactions using Host Access Class Library (HACL); a good solution to reuse old transactions and existing host program code.

From a users perspective, around 250 brokers may access existing IMS 3270 transactions from a Web browser. The benefit of revamping is really to provide users with a modern view of IMS transactions developed in the 1980's without creating a revolution on the development side! The cost is then minimal.

IMS Connector for Java

Another domain of investigation today, is the use of servlets provided by VisualAge for Java and IMS TOC. The new IMS TOC Connector for Java provides a way to create Java applications or servlets that can access IMS transactions through IMS TOC. The Java application or servlet acts as a TCP/IP client to IMS TOC. This three-tier client/server implementation appears to be the most suitable implementation for Web browser access. It exploits the benefits of Java and provides a cost-effective replacement of Java applets downloaded on every Web browser in a two-tier implementation.

Mainframe IBM 9672-R26



This company has developed a prototype to see how their applications of the future could be developed and could run on a production basis. They use WebSphere and VJava 2.0 on an NT platform. A servlet has been developed with JSP and calls to a functional bean. The functional bean calls a technical bean, which builds/controls/performs the communication with IMS.

IMS connectivity

Customer reference example:

IBM and Portuguese Ministry of Justice Elections - Legislativas /1999

IBM together with the Portuguese Ministry of Justice (D.G.S.I.) implemented, with a major success, a solution to publish on the Internet the results of the European Elections. The pioneer solution, platformed on the S/390 (9672-R26), was based on IMS Connectors and on the Domino Go WebServer.

OS/390 V2R7 with IMS V6R1 on an S/390 R26 was used to support this application,

The Portuguese Ministry of Justice (D.G.S.I.) has an application (3270 type) that receives data from each region government all over the country, and stores the information on a central IMS Database.

As the result of the elections being updated in the central database, the IMS Connectors extracted the information, and generated the results in HTML format that were published on the Internet by the Domino Go Web Server.

The architecture of the solution is represented on the figure below:

There were several key issues in this kind of solution

- The use of DGSII the application;
- The use of two different OS/390 environments, possible by establishing an APPC connection for the IMS Connectors – One of the major requirements of the customer was not touching the production environment;
- Development of Programs in “C” to automate the Generation of HTML pages;
- Development of Programs in “C” to cache the results in order to reduce the direct access of the IMS Connectors to the production environment, what could cause serious damage to the performance of our solution;
- The use of proxy’s that avoided to much traffic to the OS/390, and guarantee a good time of response in high traffic periods;

In the first 24 hours of the site they recorded 650,000 hits to the Web Server installed on the S/390, and the performance of this solution was very good.

DB2:

Net.Data Customer reference example:

A large European bank had an application requirement to support 30,000 concurrent users. The database requirement was for a total of 30 tables containing 50 GB of data and index.

OS/390 was chosen as the application platform because of its core strengths of scalability, reliability and systems management. The database was built on OS/390 DB2 with daily feeds from several other platforms.

The bank developed the application in Net.Data which was used to invoke COBOL stored procedures on the OS/390 DB2 system. Realtime access to a non-IBM platform was also implemented using MQSeries from COBOL stored procedures. Net.Data allowed rapid development of the application with excellent prototyping capability for user evaluation. Initial development was in a PC environment on NT and the programs easily moved to and from OS/390 within the development cycle.

The application was used in their intranet so no SSL was required. They used persistent connections and 2216 routers for load balancing in their sysplex environment. They determined that 350 MIPS can support the 110 transactions per second required. They are running on an OS/390 V2R5 system with plans to upgrade to V2R7 next year.

HOD Customer reference example

Host On-Demand is the host access solution at a manufacturing company in the Northeastern United States.

Application:

Easy, Web browser-based host access for more than 11,000 employees worldwide.

Recently, this progressive company began seeking an IT solution for employees who access corporate data and applications on their corporate network. As employees used an ever-increasing array of "dumb" terminals and terminal-emulation programs, with a variety of operating platforms and version levels, the cost of maintaining and administering the network was high.

In early 1999 this company installed Host On-Demand, part of the IBM SecureWay Host Integration Solution,

on its corporate data network. Served from the corporate S/390 mainframe, Host On-Demand now provides employees around the world with access to all of the company's major application programs and full emulation support, through a standard Java-enabled Web browser. Now, rather than having to support a tangle of emulation programs, platforms, and version levels, the IT staff finds that administration and maintenance are much simpler. This is because Host On-Demand provides a single point of control -- code resides on the central server, and upgrades on users' workstations occur automatically.

Users no longer need to worry about having down-level terminal emulation software or incompatible operating-system platforms. The current version of the Host On-Demand Java applet is downloaded the first time a user accesses the software, cached locally for fast performance, and upon each subsequent use checked to ensure that the latest version is in use. If the version has changed, a version update occurs, unnoticed by the user. And because Host On-Demand is Java-based, its user interface has the same look and feel across various types of operating environments.

Because of the substantial savings this company has already realized in administration and maintenance, they estimate that the Host On-Demand product will have paid for itself within a year of the initial installation.

Connector details and ordering:

First Generation (API Connectors): Connectors where Java is not required:

1. **Connector: CICS WEB SUPPORT (CWS)**
This connector works with the WebSphere HTTP server or a CICS HTTP listener, which provides direct browser access to CICS. This enables an HTML datastream to be passed to CICS, where it may be processed by the CICS 3270 Bridge or by an HTML-aware application. The main reason to use CWS is if the customer's workload is almost entirely CICS, and the customer has mostly CICS skills. Tooling is provided by CICS (BLI, Business Logic Interface, which allows resulting HTML to be passed to the front-end application) and third party tools (editing HTML templates). The key performance consideration for CWS is TCP/IP stack performance. CWS requires CICS Transaction Server (CTS) 1.2 and OS/390 2.5 for 3270 Bridge. SSL is supported in CTS 1.3, or by virtue of the underlying IBM HTTP server. CICS applications implemented using the CICS COMMAREA model

are supported by the CWS from CICS/ESA 4.1 and above.

Ordering information: CWS is now shipped with the CICS Transaction Server and CICS/ESA 4.1. If you have an old copy of CICS/ESA 4.1, you can submit a refresh order (which costs nothing) which will result in a new set of tapes being sent with the CWS included. CICS 4.1 does not support 3270 bridge. CWS with 3270 bridge is delivered as a base component of the CICS Transaction Server 1.2 so any customer who buys CICS Transaction Server today or who has an existing copy of CICS TS 1.2 will have the CWS 3270 bridge.

2. **Connector: IMS TCP/IP OTMA Connection (IMS TOC)**

This connector is recommended for high performance access from TCP/IP clients (non-browser) directly into IMS. It doesn't require a Web server (on OS/390 or a middle tier). The key performance consideration for IMS TOC is TCP/IP stack performance. No tooling is available. Supported on IMS V5 and higher. No SSL support is available. IMS TOC includes an IMS Client for Java, which is a sample showing customers how to write Java applications/applets to access IMS without CCF. This connector is being enhanced, repackaged, and renamed to IMS Connect as a separately orderable feature in IMS V7.

Ordering information: Documentation, ordering and performance information for IMS TOC is available from the following Web site:

<http://www.ibm.com/IMS/>

3. **Connector: Net.Data®**

Net.Data V2 has significant performance improvements, good integration with OS/390 Lotus Domino Go Webserver (IBM HTTP Server), and optimized DB2 connections (uses Recoverable Resource Services [RRS]). The plug-in model is recommended if high performance is required. Net.Data V1 is required if on DB2 V4. SSL is supported by virtue of the underlying IBM HTTP Server. Customized tool support is available with V2.

Execution Environments:

- A. GWAPI (Lotus Domino Go Webserver Application Programming Interface) is recommended for high performance for a "trusted" application.
- B. CGI (Common Gateway Interface) is recommended for maximum isolation and reliability when performance is not the most important factor and ease of programming is a

high priority. Servlets are recommended for synergy with other WebSphere/Java deployments and delivers moderate performance.

Ordering information: Net.Data is a no-charge feature of DB2 and can be ordered as an upgrade by contacting your local IBM Marketing Representative.

1. **Connector: Templates**

It is an entry-level solution with good performance for Lotus Domino Go WebServer-based connections to applications running under CICS (4.1) and IMS (V5). It is supported on OS/390 V1 R3 and above. SSL is supported by virtue of the underlying Lotus Domino Go Webserver with OS/390 V2 R5 and above. No tooling support exists or is planned. The ebSO services mentioned earlier are recommended.

Ordering information: The Templates are available from the following Web site:

<http://www.s390.ibm.com/nc/download.html>

Second Generation (Java) Connectors:

1. **Connector: CICS Transaction Gateway (CTG)**

This connector works with WebSphere and supports Java servlets and applications. It provides CCF classes for interfacing to CICS applications. Creation of connector applications is supported by VisualAge for Java (VA-Java) 2.0 and above. The key performance consideration for CTG is Java/Servlet performance. Up to 100 concurrent requests to CICS may be made from a single WebSphere address space, enabling many hundreds of browser connections to be supported. In order to scale beyond this, the WebSphere Application Server in scalable mode is recommended. CTG requires CICS Transaction Server (CTS) V1 R3 and OS/390 V2 R5. OS/390 V2 R7 and JDK 1.1.8 are recommended for their performance improvements. It supports SSL by virtue of the underlying IBM HTTP Server. CTG is the logical follow-on for CICS Java Gateway.

Ordering information: CTG is available from the following Web site:

<http://www.software.ibm.com/ts/cics/downloads/>

2. **Connector: IMS Connector for Java**

This connector works with WebSphere and supports Java servlets and applications. It provides CCF classes for interfacing to IMS applications. Creation of connector applications is supported by VisualAge

for Java 2.0 and above. The key performance consideration for the IMS Connector for Java is Java/Servlet performance. It requires IMS V5 and IMS TOC. It supports SSL by virtue of the underlying IBM HTTP Server. The WebSphere Application Server in scalable mode is recommended for scalability.

Ordering information: The new IMS Connector for Java is currently available with VisualAge for Java, Enterprise Edition, Version 3.0 for Windows NT. It is available for download from the VisualAge Developer Domain to registered owners of VisualAge for Java, Enterprise Edition, Version 3.0.

3. **Connector: SQLJ**

SQLJ offers an easy to use static SQL syntax (stored procedures) that requires the application developer to know the table and column names when the application is written. The primary performance consideration is Java/Servlet performance. SQLJ can provide better performance and security than JDBC, based on the program author's SQL Privileges. SQLJ is pre-processed by the DB2 SQLJ Translator into JDBC. Coding SQLJ is much simpler, more efficient and performs better at runtime. This means more programmer productivity and better system utilisation. SQLJ is an open standard created by IBM, Oracle and others. It is supported by VA-Java based tools. It requires DB2 V5 and OS/390 V2 R5. SSL is supported by virtue of the underlying IBM HTTP Server.

Ordering information: SQLJ is available as a download for DB2 V5 and V6 from: <http://www.software.ibm.com/data/db2/os390/sqlj.html>

4. **Connector: JDBC**

JDBC uses dynamic SQL so it can deal with situations where you don't know the table and column names at the time the application is written. The primary performance consideration is Java/Servlet performance. It is supported by VA-Java based tools. It requires DB2 V5 and OS/390 V2 R5. SSL is supported by virtue of the underlying IBM HTTP Server.

Ordering information: JDBC is shipped with DB2 V5 and V6.

5. **Connector: Host On Demand**

The Host On-Demand Connector is a development tool that provides the basic support for Host connectivity found in the full Host On-Demand product. It provides support for 3270, 5250 and VT host systems. The connector does not provide "green screens", but does enable and simplify

connectivity to these hosts. The Connector is a non-visual bean that provides methods and properties for setting up and establishing communications with a host system. The connector provides the ability to send host commands and retrieve screens of host data. It determines the behavior and characteristics of the session with the host through its properties, which include such things as the session type (3270, 5250, and VT), host, port, session ID, PS size (for example, 24 rows by 80 columns), and the host code-page.

Ordering information: The Host On-Demand Connector is distributed as a component of VisualAge for Java Enterprise Edition or can be downloaded from the VisualAge Developer Domain at <http://www.software.ibm.com/vadd>. The full Host On-Demand product which provides a full function Java emulator and SecureWay Screen Customizer can be ordered separately.

6. **Connector: Host Publisher**

Host Publisher is one of the best solutions for quickly extending existing applications to the web without modifying the applications. It uses the Java Server-side processing model to provide access to back end host applications and data, and delivers HTML Web pages to Web browser users. Host Publisher works with WebSphere V2 to provide the runtime environment for Java Server pages, and creates reusable beans for Java, called Integration Objects, that can be incorporated into other Host Publisher applications or standard Java applications utilizing WebSphere Tooling. These reusable objects, when invoked, will access the existing host applications, invoke the desired transaction, and extract the desired data. You have the option of adding new business logic using Java, and you can merge data from multiple host applications into a single HTML output page creating completely new views to the user. Host Publisher provides access to 3270, 5250, VT, Java, and JDBC compliant databases such as IBM DB2, and Oracle. It includes SSL support to ensure secure connections, and provides connection pools to enhance performance.

Ordering information: For 3-tier environments, Host Publisher for AIX, Windows NT, and Solaris is available. Host Publisher for OS/390 is available for ordering and will be generally available January 21, 2000.

Additional Information on Connectors:

Links to more information on e-business connectors for OS/390 is available on our connectors Web site:
<http://www.s390.ibm.com/ebusiness/connectors/>

- **Redbooks:**
 - Revealed! CICS Transaction Gateway and more CICS Clients Unmasked, SG24-5277
 - Revealed! Architecting Web Access to CICS, SG24-5466
 - Web-to-Host Integration Solutions, SG24-5237-00 1998-10-01 (an update is in progress)
 - IMS e-business Connect Using the IMS Connectors, SG24-5427-00 1999-07-02
 - The Trader Story, A Performance Study of Web Access to CICS, SG24-5748-00 (coming soon)

- **Other sources for help:**
 - The OS/390 e-business Integration Test Report:
www.s390.ibm.com/ebusiness/techinfo/ebitr.html
 - Performance and tuning guidance:
<http://www.s390.ibm.com/ebusiness/> Select the performance category on the navigation bar on the left. This category will be available at the end of December, 1999.
 - CICS Web Selection Guide:
<http://www.software.ibm.com/ts/whitepaper/cics/librarycicsweb/>
 - The JavaSoft Web Page for the latest specification on the Common Connector Framework: <http://www.javasoft.com/>
 - The OS/390 Planning Assistant for e-business:
<http://www.s390.ibm.com/os390/wizards/ebiz>
 - The OS/390 Web-Enablement Overview, GF22-5138

Appendix A: Connector Matrix

DB2 Connector Options:

	Environment	Recommended Connector
a	<ul style="list-style-type: none">• OS/390 V2 R5 or higher• DB2 V5 or higher• Java	Java connector: <ul style="list-style-type: none">• SQLJ• JDBC• Host Publisher
b	<ul style="list-style-type: none">• OS/390 V2 R5 or higher• DB2 V5 (highly recommended) or higher• Minimal skill required for fast deployment (No Java or Java skills)	GWAPI: <ul style="list-style-type: none">• Net.Data V2
c	<ul style="list-style-type: none">• OS/390 V1 R3 or V2 R4• DB2 V4• Minimal skill required for fast deployment (No Java or Java skills)	GWAPI: <ul style="list-style-type: none">• Net.Data V1

CICS Connector Options:

	Environment	Recommended Connector
a	<ul style="list-style-type: none"> • OS/390 2.5 or higher • CICS TS 1.1 or higher • Java • No need to access 3270 transaction 	<ul style="list-style-type: none"> • CICS Transaction Gateway (CTG)
b	<ul style="list-style-type: none"> • Any OS/390 • CICS/ESA 4.1 or higher • Able to access business logic directly (no 3270) 	<ul style="list-style-type: none"> • CICS WEB SUPPORT (CWS)
c	<ul style="list-style-type: none"> • OS/390 2.5 or higher • CICS TS 1.2 or higher • Minimal skill required for fast deployment (No Java skills available) • Need to access 3270 transaction-----> • • No 3270 transaction -----> 	<ul style="list-style-type: none"> • CICS WEB SUPPORT • • • • with 3270 Bridge, Host Publisher • Encode/Decode
d	<ul style="list-style-type: none"> • OS/390 2.5 or higher • CICS/ESA 4.1 or higher • Java • Need to access only 3270 transactions 	<ul style="list-style-type: none"> • Host On Demand • Host Publisher (Java not required on the client)
e	<ul style="list-style-type: none"> • OS/390 1.3 or higher • CICS/ESA 4.1 or higher • Minimal skill required for fast deployment (No Java or Java skills) • Able to access dynamic program links directly (no 3270) • Web enabling CICS transactions and non-CICS data/transactions (e.g. IMS) 	<ul style="list-style-type: none"> • Templates (CICS/IMS)
	<ul style="list-style-type: none"> • Vendor solutions <ul style="list-style-type: none"> ▪ OS/390 level–see vendor for details ▪ CICS level–see vendor for details <p>Examples: CrossPlex, GT Software, Lincoln Software, Jacada (Note 1)</p>	

Note 1: For a more complete list of vendors who support the S/390 platform, please visit our Partners in Development Web site at: <http://www.s390.ibm.com/s390da/program/memlist.html>

IMS Connector Options:

	Environment	Recommended Connector
a	<ul style="list-style-type: none"> • OS/390 • IMS V5 • Web Browser client • Java 	<ul style="list-style-type: none"> • IMS Connector for Java
b	<ul style="list-style-type: none"> • OS/390 • IMS V5 • Any TCP/IP client, including any Java or Web browser client • Minimal skill required for fast deployment (No Java or Java skills) 	<ul style="list-style-type: none"> • IMS TOC
c	<ul style="list-style-type: none"> • OS/390 2.5 or higher • Web Browser client • Java • Need to access only 3270 transactions 	<ul style="list-style-type: none"> • Host On Demand • Host Publisher (Java not required on the client)
d	<ul style="list-style-type: none"> • OS/390 2.5 or higher • Web Browser client • Minimal skill required for fast deployment (No Java or Java skills) 	<ul style="list-style-type: none"> • Templates (CICS/IMS) • Host Publisher
	<ul style="list-style-type: none"> • Vendor solutions <ul style="list-style-type: none"> ▪ OS/390 level—see vendor for details ▪ IMS level—see vendor for details <p>Examples: TransLink and Neon Systems (Note 1)</p>	

Note 1: For a more complete list of vendors who support the S/390 platform, please visit our Partners in Development Web site at: <http://www.s390.ibm.com/s390da/program/memlist.html>



(C)International Business Machines Corporation 1999

IBM Corporation
Marketing Communications, Servers
Route 100
Somers, N.Y. 10589

Printed in the United States of America, 12/99
All Rights Reserved

This publication was produced in the United States.
IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the products or services available in your area.

You can find additional information via IBM's World Wide Web server at <http://www.ibm.com>.

IBM Hardware products are manufactured from new parts or new and serviceable used parts. Regardless, our warranty terms apply.

Actual performance and environmental costs will vary depending on individual customer configurations and conditions.

® CICS, DB2, DRDA, Language Environment, MQSeries, Net.Data, OS/390, Parallel Sysplex, RACF, SecureWay, S/390, and VisualAge are registered trademarks of the International Business Machines Corporation.

™ IMS and WebSphere are trademarks of the International Business Machines Corporation

® Lotus is a registered trademark of the Lotus Development Corporation.

® Windows NT is a registered trademark of the Microsoft Corporation.

® UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

™ Java is a trademark of Sun Microsystems, Inc.

™ Lotus Go Webserver is a trademark owned by Lotus Development Corporation

All other registered trademarks and trademarks are the properties of their respective companies.

All third party products mentioned in this publication are products of their respective providers. IBM has not verified the performance or functions of these products and provides no warranties, express or implied, including but not limited to the implied warranties of merchantability or fitness for a particular purpose, regarding those products. Any warranties and all performance and functional information concerning these products is available from their respective providers.