VisualAge: A Powerful New Vision of Programming™

Are your computer systems working for you . . . or do you do all the work for them? Today's businesses and programmers need a powerful industrial-strength solution. IBM's Object-Oriented Application Development provides that solution. At the heart of the solution is the VisualAge™ family of products, whose dynamic new technologies ease the transition to the next generation of client/server computing systems.

Today's heterogeneous, scalable computing environment is a fast-changing world of many choices. The VisualAge family gives you the tools and flexibility you need to make those choices:

- You can increase productivity many times over with the power of visual programming.
- You can develop applications in your language of choice that is portable across client platforms, communication protocols, and databases.
- With "construction from parts" technology, you can build applications from fully reusable components.
- Open standards, open systems, and application portability mean you can work with your vendors of choice.

Flexibility, productivity, new technologies, and open systems: together, they add up to a powerful new vision of programming. VisualAge offers you an end-to-end, reliable, fully integrated solution.

Today's business environment

Industries that rely heavily on their computing power face serious challenges as they move toward the future. Users of corporate information systems, large or small, expect intuitive, easy-to-use, yet robust business applications. Constructing these mission-critical applications is more challenging than ever. The aim of IBM is to meet these challenges as it moves today's information systems into the future.

Today's computing environment includes a vast array of hardware and software, both old and new, that must work together for companies to realize the full potential of their investments. **Interoperability** is essential. Companies need tools that work across the enterprise to deliver on the promises of client/server computing and networked systems. That way, they can leverage existing and emerging technologies for the greatest advantage in the marketplace.

More than ever, staying ahead of the competition demands **constant innovation**. To develop state-of-the-art software, companies like yours need tools that enable quick response, increased productivity, increased quality, and leverage of existing assets.

Responsiveness: The mission-critical software that enables your company to stay ahead of the competition must be flexible, interoperable, and responsive to your needs. It must change quickly and easily to support changing market conditions. Developing this software requires tools that can respond to the rapid pace of change in the marketplace, the shifting sands of regulation, and competition from every direction.

Productivity: Today's application development tools need to increase productivity by reducing time to market for new products and applications, and reduce time spent writing new code. They need to encourage reuse of proven application components that closely model your business practices.

But being productive doesn't simply mean moving faster. It means moving smarter, with tools that enable a close relationship between your applications and their underlying business models. It means true iterative development, with early and frequent end-user involvement.

Quality improvement: Reusable components can improve application quality, as companies come to rely on a growing base of existing, time-tested components to build their new applications. This kind of quality improvement also reduces risk; new application development projects can depend on proven components.

Asset leverage: Your company maintains enormous and growing value in your existing information assets, which include data, legacy code, computer skills, and business process models. New applications must protect these assets by maintaining easy access to them, whatever system they are on. Even more, to stay competitive, new applications must leverage these assets and build upon them.

Today's programming needs

Companies require a great deal from their programmers. Programmers must be increasingly productive to create the complex software needed by today's businesses, and must have the cross-platform skills required by today's heterogeneous systems. To create a world-class programming environment, companies must address the following issues:

Productivity: Today's client/server applications are more complex and more difficult to develop. Therefore, development tools must be easy to use. They must work within an integrated development environment, and work seamlessly throughout the software development cycle: modeling, design, coding, deployment, maintenance, and support. Such tools are even more productive when they are responsive to end-user input and enable iterative design changes.

Reuse: Many reuse factors can boost programming success: portability of code from one runtime platform to another, reuse of code components from application to application and within the same application, reuse of programming skills across platforms and from project to project. Programmers need tools that enable reuse wherever possible.

Teamwork: Effective ways of working together are essential to building today's complex applications. Team members need to share code easily and work with one another's components. They require version management and change control on an increasingly complex level.

Scalability: Your company's applications need to keep up with the growth and changes in the company itself. Your applications need to scale as the enterprise increases in size and complexity. And if the programming tools themselves can scale, it is easier to build applications that scale effectively.

New skills: The experience that programmers gain while working with new tools must be relevant and transferable. Programming skills must have a future, so that the programmers can feel satisfaction and security in their work, and be more open to innovation and risk-taking.

Job satisfaction: Programmers who use the tools they want, and develop the skills they want, are more likely to develop a sense of relevance to the company and an alignment with company goals. They will be responsive to customer needs and corporate needs, and will deliver the right software at the right time.

What makes VisualAge so powerful?

You can achieve great results with the VisualAge family of products because they harness the power of objects with visual programming. VisualAge enables true client/server solutions on an enterprise-wide scale, from the host to the desktop. But VisualAge is far more than a collection of powerful new technologies; VisualAge is an end-to-end solution.

Success with new technologies requires planning and training. It requires consulting, education, and support. You need more than technology; you need technology plus extended support and services. The VisualAge solution brings it all to you, with offerings specially tailored to ensure your success with these powerful new technologies.

The power of objects

True reusability: Objects enable you to reuse application components, so that a substantial amount of work has been done before you even start. VisualAge brings you extensive libraries of classes that act like independent building blocks you can use to assemble an application.

Reusability offers several benefits:

- **Faster development:** You can create applications in less time and respond more quickly to rapidly-changing requirements.
- **Higher-quality deliverables:** All applications developed with reusable components take advantage of the mature function available in those components.
- **Consistency:** Reusable parts can represent your common business objects and ensure a consistent look and feel to your applications.
- **Reduction of maintenance costs:** Defects are easy to pinpoint, and because they are self-contained in reusable objects, they only need to be fixed once.

Productivity: Objects increase productivity in a number of ways. The encapsulation of function and data within objects protects both your data and your code, and simplifies debugging and reuse. Inheritance enables parts to share common function that doesn't have to be constantly rewritten. This cuts down on development time, because developers only need to write new function, and can reuse function that already exists.

Pacing yourself: Because Object Technology is modular and flexible, you can implement it at your own pace. You can start small, with non-critical applications, and scale up to meet the needs of even the largest enterprise. Object Technology gives you precisely the flexibility and responsiveness you need.

The power of visual programming

Visual programming enables you to build your applications by manipulating icons and interface controls, and making connections between them. Each visual part and connection corresponds to a pre-coded, reusable object. In this way, visual programming enables you to focus on your business-specific coding when you create an application.

Visual programming provides a high level of expression that closely models your actual business processes. To use it, you don't need to learn complex programming syntax. And it's a natural model for GUI and report building, because GUIs and reports are visual to begin with.

Construction from parts: Just as a circuit board is manufactured from individual, prefabricated parts, you can combine individual software parts to create application components (which are actually composite parts). Just as a circuit board can then be combined with other complex components to create a computer, you can combine your composite parts into a full-function application.

The same composite parts can be recombined to form different applications. You achieve a consistent look, ease of use, and high application quality. Using visual programming techniques, you can wrapper complex application components and work with them as if they were a single part.

The prefabricated nature of parts also makes it easier to take your application apart and reassemble it as you go through the development process.

How you can harness the power

To ensure your success with visual programming across your enterprise, VisualAge gives you everything you need:

- Language of choice
- Platform of choice
- Interoperability
- Component management

- Extended support
- Partnerships

The key to VisualAge's power is the integration of extended support and customer partnerships with visual programming, languages, platforms, and tools. This is how VisualAge provides you with an end-to-end, proven solution to your application development needs.

Language of choice

VisualAge brings object technology and visual programming to the most popular industrial-strength OO languages, C++ and Smalltalk. For each language, VisualAge offers a complete end-to-end solution. You can construct parts and applications in the language most appropriate to your project, with all the benefits of Object Technology and a common Visual Programming paradigm.

Platform of choice

Today's computing environment requires that your data, application logic, and GUI are available wherever and whenever they are needed, and in the operating system you prefer. Each VisualAge language offers you a range of tools for accessing data wherever it resides in your enterprise and for communicating across your systems. You can use VisualAge products to develop your applications in OS/2, Windows, or AIX, with more platforms coming.

Within the language of your choice, most components are platform independent, so that parts and even entire applications developed on one platform can be ported to another. And development skills are platform independent, too: your developers can create Windows applications one day and OS/2 applications the next.

Interoperability

With VisualAge products, you have many paths to interoperability, including the easy construction of client-based front ends for your existing server applications. Other paths include dynamic data exchange, a broad range of communications parts, data access parts, and parts that wrapper existing C and COBOL applications. SOM support enables parts to communicate with one another across different languages. And DSOM extends SOM across a distributed environment.

Distributed objects enable a single application to run logic or store data transparently on the desktop, on intermediate servers, or on enterprise servers. You can connect to CICS or IMS systems with VisualAge. And soon, with Smalltalk for MVS, you can use the same business objects across your entire system, from the host to the workstation.

Whatever platform you choose to run on, the VisualAge products fulfill the promise of Open Systems. And IBM is dedicated to providing still more interoperability as part of its committment to a reliable client/server solution.

Component management

Whether your application development team is large or small, VisualAge component management makes it easy for them to share code components and to track changes as they develop new applications. In addition, the extensive VisualAge class libraries make it easy to reuse existing classes and parts.

Extended support

VisualAge gives you access to a wide range of traditional product support sources, such as help lines, online bulletin boards, and lists of frequently asked questions (FAQs). But our VisualAge support goes far beyond this. It's not just support; it's a solution.

You have access to extensive training and educational offerings in products and related technologies. Offerings include high-level overviews of Object Technology and its methodologies, as well as offerings in various approaches to application development with VisualAge. Courses are available world-wide.

Partnerships

Customer partnerships with IBM give you a full spectrum of access to the best of the VisualAge solution. IBM Consulting and Professional Services provide world-wide assistance in such areas as training/mentoring, designing and building reusable componentry, setting up multi-user development environments, and prototyping.

VASTeam members are independent vendors who have become VisualAge specialists to provide you the best in wide-ranging services. In addition, through the Object Connection, IBM brings you into partnership with independent vendors who build "plug-and-play" VisualAge parts for almost every business purpose.

Proven Results

VisualAge is already proving itself as a reliable end-to-end solution across a spectrum of industry sectors. In the past two years, many companies like yours have adopted IBM's pioneering VisualAge products as their strategic application development solution.

Whether their language of choice is Smalltalk or C++, these companies are achieving greater responsiveness to the needs of the marketplace, greater flexibility in developing applications, a quicker time to market, and an overall greater productivity.

Smalltalk case study—an automobile manufacturer: One of the world's largest automotive companies wanted to improve the inventory and finance management systems for their regional offices and dealers. This would improve time to market for their auto parts and speed the customer repair cycle.

Their overall strategy called for a robust application system that would manage their large centralized distribution facility. They wanted the business logic available at client installations where the auto parts were needed, and they also wanted continuous update of key corporate data on their centralized databases. This required a staged migration to a new client/server environment.

Results: Using VisualAge, the company developed all the required function within their aggressive schedule, yielding a distinct competitive advantage. According to their Vice-President of Information Systems, "VisualAge . . . gave us a methodology that I think is pretty revolutionary from the standpoint of classic business application development."

Smalltalk case study—an insurance company: A medium-sized insurance company wanted to streamline its delivery of new products. To successfully target products in the rapidly-changing insurance market, the company required a flexible and sophisticated use of database statistics.

To make this possible, they needed an effective client/server environment that could port business rules from mainframe into client applications, and provide online access from GUI clients to existing host databases. In this way they could tailor their new insurance offerings for precise market segments.

Results: As a result of using VisualAge, the company is now deploying applications at lower expense and in a shorter timeframe. They can deliver new varieties of products because the cost barriers are disappearing. According to the Vice President of their Personal Lines System, "I think the industry is going . . . to see that object-oriented technology is not just the wave of the future; it's here already."

C++ case study—a transportation company: A major transportation company wanted to speed the coupling of freight cars into trains and reduce misrouting. They needed to upgrade the way they tracked the locations and inventories of the cars. Their freight yard operators, located in control booths, needed all the flexibility and user-friendliness that a computer system can offer.

The operators required small, single-task applications that were entirely mouse-driven, to be used with one hand while the other hand was free for additional tasks. And due to the large number of these small applications required, the company needed a fast prototyping-to-delivery cycle.

Results: With VisualAge, the applications came in on time and they met the user requirements to rave reviews.

C++ case study—a government contractor: A government contractor wanted to move its customers from command-line driven applications to a more up-to-date GUI system. This contractor specialized in providing software for organizations ranging from municipalities to the Federal government. Because their customers ranged from the largest to the smallest, the software had to be scalable. And it required a simple design structure to accomodate maintenance and programmer turnover.

Many of the contractor's customers required complex contract-management software. The software had to clearly model customer business rules and governmental regulations. It also required strong accountability on changes and sign-offs.

Results: Using VisualAge, the contractor was able to develop a well-designed, elegant application system that fulfilled all requirements and was fully scalable. And they developed the system on time and on budget.

VisualAge: how it all fits together

The overall story is really quite simple: whatever your language and platform of choice, IBM brings you an integrated development environment based around powerful visual development tools.

The story begins with the Smalltalk edition of IBM's VisualAge. Smalltalk is a logical place to start because it is a pure object-oriented language. Once the Smalltalk edition was established as an industrial-strength application builder with rich functionality, IBM started porting the best of the new techniques and technologies to the world of C++.

The roots of C++ in C make it a popular hybrid object-oriented language that is familiar to existing C programmers. In the same way as Smalltalk optimizes for rapid application development, C++ optimizes for performance.

Once the two editions of VisualAge were established award-winning products, IBM turned its attention to extending the base products with even more powerful function and strengthening their position as power tools for enterprise development.

Both editions of VisualAge combine visual programming and an integrated development environment with the latest in client/server advances. With available parts that extend the base environment, and with a wide range of user services, VisualAge adds up to an application development solution for the '90s and beyond.

Integrated development environment

At the heart of each VisualAge product is an integrated development environment, which consists of the object-oriented language of your choice, combined with extensive part and class libraries, browsers, debuggers, and a state-of-the-art visual application builder.

Visual application builder

The visual application builder is far more than just an interface builder and prototyping tool. It's a power tool for creating industrial-strength applications, many of which do not require you to write any code at all. The VisualAge Composition Editor gives you a visual work surface on which to create your application. You simply select parts from an extensive library and make the appropriate point-and-click connections. Need a new part for a unique purpose? You can create one using the other tools of the integrated development environment.

Parts include everything you need to build GUIs, together with a broad range of application logic. It's easy to create additional application logic using the Smalltalk Script Editor or the C++ code editor. And the composite components you build can be wrappered as individual parts and used "as is" from application to application.

You can add new parts to the VisualAge palette, whether from IBM or from independent vendors, to greatly expand the business function in your applications. And new extensions to VisualAge are available all the time, both from our growing family of “Object Connection” partners and IBM’s own continuous enhancements to the base environments.

The parts palette: VisualAge parts are intuitive. They look like real-world objects. And, in fact, they closely model your existing business objects and practices. The VisualAge parts palette includes categories such as buttons, lists, windows, external functions, and database functions. Inside these categories are parts such as entry fields, sliders, ordered collections, program starters, database queries, and stored procedures.

Parts on the palette can be used over and over. So can the composite parts you build from your basic parts. It’s easy to modify parts and have your new parts display on the palette. You can even create new categories to suit your needs.

Database access: The VisualAge products provide native support for IBM DB2/2 databases, including a complete suite of functions like server-enforced referential integrity, stored procedures, and industry-standard SQL. In addition, VisualAge can be extended to access a range of IBM host databases.

The Smalltalk editions include far-ranging native support for ORACLE databases, bringing you the data access you need on the platform of your choice. Even if you must access multiple databases in your current environment, VisualAge can make the connections with its Open Database Connectivity (ODBC) interface for Smalltalk. ODBC enables you to create applications that access all your data sources, from spreadsheets to sophisticated relational databases like INFORMIX and SYBASE.

Communications and transactions: The VisualAge communications and transactions parts mask the complexity of communicating among system components and even to remote systems. You can access program logic and data by using a wide variety of popular communications protocols and APIs. For Smalltalk, these include APPC, CPI-C, CICS, ECI, IMS, MQSeries, NetBIOS, RPC, TCP/IP, and EHLLAPI (3270 terminal emulation).

Multimedia: Add multimedia to existing applications, or create state-of-the-art new applications. VisualAge Multimedia gives you access to industry-standard interfaces for playback of audio and video files. The multimedia parts include a rich set of visual controls to custom-design your own multimedia applications.

VisualAge in the C++ environment

In addition to the Visual Builder, the C++ development environment includes a rich set of tools such as a Data Access Builder to quickly construct useful data access parts, and a Visual Browser that enables you to edit your source code, as well as look at graphical relationships between program components, online documentation for classes, and types of program objects. Other tools include a state-of-the-art compiler and linker, a language-sensitive editor, a debugger, and performance analyzer.

The environment also includes the IBM Open Class Library, a comprehesive set of building blocks. These classes serve as the basis for all the VisualAge C++ parts, and provide a rich range of function in reusable, reliable, extensible objects.

To get you started quickly with VisualAge C++, the environment includes “Project Smarts,” a set of application frameworks you can use to build your own. These preconfigured applications lead you through the process of learning the development environment. In addition, the VisualAge Workframe enables you to fully customize your work environment, integrating tools of your choice with your VisualAge tools, and using the “Build Smarts” dialog to switch quickly between the most common build options.

VisualAge in the Smalltalk environment

In the Smalltalk edition, the Visual application builder is seamlessly integrated into the development environment. The applications you construct are actually Smalltalk objects, and they compile and link dynamically, so that no additional compile/link tools are needed.

The environment also includes a Script Editor that guides you through the construction of common, useful Smalltalk scripts, and a Public Interface Editor that helps you define the interfaces your VisualAge parts use to communicate with one another.

Browsers give you full access to the underlying Smalltalk code. The Organizer window helps you track your development work and get started using the tools, and a full-function debugger helps you analyze and debug your applications.

A rich variety of parts are available from IBM to extend the base Smalltalk environment. These are mature, tested parts that are easy to integrate into your applications. They include the many database, communications, and transaction parts mentioned earlier. In addition, **External function parts** enable you to wrap your existing C or COBOL code for use in VisualAge applications. **VBX support** does the same for Visual Basic components and applications.

Need reports generated from any of your Smalltalk applications? The VisualAge **Reports feature** provides a dozen report parts to help you create, format, preview, and print attractive and meaningful reports.

The **VisualAge AS/400 Connection** lets you work with AS/400 system resources as visual parts, including database files, data queues, and data areas. Using visual construction techniques, you can connect these parts into powerful client/server applications. You can make use of distributed and host server programming.

With the **VisualAge IMS Connection**, you can move still further into the world of client/server development by communicating between object-oriented client applications on your workstations and traditional System/390 programs on the host. The IMS Connection wraps host transactions into Smalltalk business objects for easy messaging in your workstation or LAN environment.

VisualAge and client/server enablement

VisualAge paves the way to an effortless and powerful client/server computing environment. Even with industrial-strength heterogeneous systems, VisualAge can help you make the connections you need. From the host to the workstation, all around the LAN and at any location where you need your data to run, VisualAge offers you an end-to-end, proven, integrated solution.

Existing Client/Server Assets: The applications you build with VisualAge provide easy access to your existing data and business logic wherever it resides in your system. External function parts make it easy to call existing C or COBOL code from any VisualAge application. With the database parts and communications/transactions parts, the AS/400 Connection and the IMS Connection, you can leverage your existing system resources. VisualAge gives your developers easy visual access to a wide range of communications and database protocols.

Cross-Platform Compatability: Most VisualAge classes are platform-independent; they exhibit a wide range of portability. It is easy to develop on one platform and run in another, or to exchange parts between platforms. You can also repackage or recompile your applications for different runtime platforms.

Distributed Computing: Distributed Computing is the wave of the future, and the VisualAge products are already riding that wave. Distributed objects help developers deal with the complexity of heterogeneous two- and three-tier client/server architectures. By enabling free distribution of function and data based on developer design considerations, distributed objects enable a high degree of scalability and robustness.

VisualAge Distributed enables you to easily develop distributed Smalltalk applications with true local/remote transparency across your system environment. You can partition your applications in the ways that best fit your business needs and you can protect them with industrial-strength security.

VisualAge **SOMSupport** enables C++ objects and Smalltalk objects to communicate with one another, so that applications are no longer limited by language or location. And by adding DSOM, applications can be composed of objects that can reside on different servers, or in entirely different systems.

The Object Connection

The Object Connection program gives you a shortcut to developing the complex and specialized parts that match your business needs. Through the Object Connection, many large and small vendors make specialized parts for VisualAge. These modular, self-contained parts fit seamlessly into the integrated development environment. Object Connection parts include business graphics, chart parts, frameworks for desktop publishing, multinational translation parts, and many more.

IBM encourages vendors to join the Object Connection program and provides the support they need to enable their parts for the VisualAge environment.

The industry-standard solution

IBM's VisualAge family of products conform to the latest industry standards, including the C++ ANSI standard, the proposed ANSI Smalltalk standard, and OMG's Common Object Request Broker Architecture (CORBA). This means that components, parts, and development tools adhere to the highest standards of quality and reliability in the industry. You can have confidence in the lasting value of your applications and your development tools.

VisualAge and you: making the connection

With the VisualAge products, IBM brings you powerful new tools for all your enterprise application development needs. VisualAge connects you to your data and existing applications wherever they reside in your enterprise, from the desktop to the host. And VisualAge connects you to powerful support, services, and partnerships. VisualAge: a true end-to-end solution.

Want to know more?

For further information on IBM's VisualAge products, call the following numbers:

In North America
 1-800-IBM-2279[1]
Outside the U.S. and Canada
 1-919-254-4760
In Europe, the Middle East, and Africa
 33-92-11-50-11

You can also append CompuServe under VisualAge (GO VISUAL).

For VisualAge information on the World Wide Web, go to the IBM Software Home Page:

http://www.software.ibm.com/

and do a search on the keyword **goVisualAge**.

1 After January 1, 1996, call 1-800-IBM-3333 ext. GS059.

IBM®

This publication may refer to products that are not currently available, or may not be available in your country. IBM product plans are subject to change.

This publication could contain technical inaccuracies or typographical errors.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries: IBM, AIX, "A Powerful New Vision of Programming," AS/400, CICS, DB2/2, IMS, MQSeries, OS/2, SOMobjects, "The Object Connection," VisualAge.

The following terms are trademarks of other companies:

C++ (AT&T Corporation);
INFORMIX (Informix Software);
ORACLE (Oracle Corporation);
SYBASE (Sybase, Inc.);
Visual Basic (Microsoft Corporation).

Windows is a trademark of Microsoft Corporation.

IBM's VisualAge products and services are not associated with or sponsored by VisualEdge Software, Ltd.

G325-0801-02 VisualAge V3.0 A powerful new vision of programming™ Software Solutions RTP, NC 27709

Contents

VisualAge: A Powerful New Vision of Programming™ . . . 1
Today's business environment . . . 1
Today's programming needs . . . 2
What makes VisualAge so powerful? . . . 2
  The power of objects . . . 3
  The power of visual programming . . . 3
How you can harness the power . . . 3
  Language of choice . . . 4
  Platform of choice . . . 4
  Interoperability . . . 4
  Component management . . . 4
  Extended support . . . 4
  Partnerships . . . 5
Proven Results . . . 5
VisualAge: how it all fits together . . . 6
  Integrated development environment . . . 6
  Visual application builder . . . 6
  VisualAge in the C++ environment . . . 7
  VisualAge in the Smalltalk environment . . . 8
  VisualAge and client/server enablement . . . 8
  The Object Connection . . . 9
The industry-standard solution . . . 9
VisualAge and you: making the connection . . . 9
Want to know more? . . . 9
Trademarks . . . 10

VisualAge: A Powerful New Vision of Programming™

Are your computer systems working for you . . . or do you do all the work for them? Today's businesses and programmers need a powerful industrial-strength solution. IBM's Object-Oriented Application Development provides that solution. At the heart of the solution is the VisualAge™ family of products, whose dynamic new technologies ease the transition to the next generation of client/server computing systems.

Today's heterogeneous, scalable computing environment is a fast-changing world of many choices. The VisualAge family gives you the tools and flexibility you need to make those choices:

- You can increase productivity many times over with the power of visual programming.
- You can develop applications in your language of choice that is portable across client platforms, communication protocols, and databases.
- With "construction from parts" technology, you can build applications from fully reusable components.
- Open standards, open systems, and application portability mean you can work with your vendors of choice.

Flexibility, productivity, new technologies, and open systems: together, they add up to a powerful new vision of programming. VisualAge offers you an end-to-end, reliable, fully integrated solution.

Today's business environment

Industries that rely heavily on their computing power face serious challenges as they move toward the future. Users of corporate information systems, large or small, expect intuitive, easy-to-use, yet robust business applications. Constructing these mission-critical applications is more challenging than ever. The aim of IBM is to meet these challenges as it moves today's information systems into the future.

Today's computing environment includes a vast array of hardware and software, both old and new, that must work together for companies to realize the full potential of their investments. **Interoperability** is essential. Companies need tools that work across the enterprise to deliver on the promises of client/server computing and networked systems. That way, they can leverage existing and emerging technologies for the greatest advantage in the marketplace.

More than ever, staying ahead of the competition demands **constant innovation**. To develop state-of-the-art software, companies like yours need tools that enable quick response, increased productivity, increased quality, and leverage of existing assets.

Responsiveness: The mission-critical software that enables your company to stay ahead of the competition must be flexible, interoperable, and responsive to your needs. It must change quickly and easily to support changing market conditions. Developing this software requires tools that can respond to the rapid pace of change in the marketplace, the shifting sands of regulation, and competition from every direction.

Productivity: Today's application development tools need to increase productivity by reducing time to market for new products and applications, and reduce time spent writing new code. They need to encourage reuse of proven application components that closely model your business practices.

But being productive doesn't simply mean moving faster. It means moving smarter, with tools that enable a close relationship between your applications and their underlying business models. It means true iterative development, with early and frequent end-user involvement.

Quality improvement: Reusable components can improve application quality, as companies come to rely on a growing base of existing, time-tested components to build their new applications. This kind of quality improvement also reduces risk; new application development projects can depend on proven components.

Asset leverage: Your company maintains enormous and growing value in your existing information assets, which include data, legacy code, computer skills, and business process models. New applications must protect these assets by maintaining easy access to them, whatever system they are on. Even more, to stay competitive, new applications must leverage these assets and build upon them.

Today's programming needs

Companies require a great deal from their programmers. Programmers must be increasingly productive to create the complex software needed by today's businesses, and must have the cross-platform skills required by today's heterogeneous systems. To create a world-class programming environment, companies must address the following issues:

Productivity: Today's client/server applications are more complex and more difficult to develop. Therefore, development tools must be easy to use. They must work within an integrated development environment, and work seamlessly throughout the software development cycle: modeling, design, coding, deployment, maintenance, and support. Such tools are even more productive when they are responsive to end-user input and enable iterative design changes.

Reuse: Many reuse factors can boost programming success: portability of code from one runtime platform to another, reuse of code components from application to application and within the same application, reuse of programming skills across platforms and from project to project. Programmers need tools that enable reuse wherever possible.

Teamwork: Effective ways of working together are essential to building today's complex applications. Team members need to share code easily and work with one another's components. They require version management and change control on an increasingly complex level.

Scalability: Your company's applications need to keep up with the growth and changes in the company itself. Your applications need to scale as the enterprise increases in size and complexity. And if the programming tools themselves can scale, it is easier to build applications that scale effectively.

New skills: The experience that programmers gain while working with new tools must be relevant and transferable. Programming skills must have a future, so that the programmers can feel satisfaction and security in their work, and be more open to innovation and risk-taking.

Job satisfaction: Programmers who use the tools they want, and develop the skills they want, are more likely to develop a sense of relevance to the company and an alignment with company goals. They will be responsive to customer needs and corporate needs, and will deliver the right software at the right time.

What makes VisualAge so powerful?

You can achieve great results with the VisualAge family of products because they harness the power of objects with visual programming. VisualAge enables true client/server solutions on an enterprise-wide scale, from the host to the desktop. But VisualAge is far more than a collection of powerful new technologies; VisualAge is an end-to-end solution.

Success with new technologies requires planning and training. It requires consulting, education, and support. You need more than technology; you need technology plus extended support and services. The VisualAge solution brings it all to you, with offerings specially tailored to ensure your success with these powerful new technologies.

The power of objects

True reusability: Objects enable you to reuse application components, so that a substantial amount of work has been done before you even start. VisualAge brings you extensive libraries of classes that act like independent building blocks you can use to assemble an application.

Reusability offers several benefits:

- **Faster development:** You can create applications in less time and respond more quickly to rapidly-changing requirements.
- **Higher-quality deliverables:** All applications developed with reusable components take advantage of the mature function available in those components.
- **Consistency:** Reusable parts can represent your common business objects and ensure a consistent look and feel to your applications.
- **Reduction of maintenance costs:** Defects are easy to pinpoint, and because they are self-contained in reusable objects, they only need to be fixed once.

Productivity: Objects increase productivity in a number of ways. The encapsulation of function and data within objects protects both your data and your code, and simplifies debugging and reuse. Inheritance enables parts to share common function that doesn't have to be constantly rewritten. This cuts down on development time, because developers only need to write new function, and can reuse function that already exists.

Pacing yourself: Because Object Technology is modular and flexible, you can implement it at your own pace. You can start small, with non-critical applications, and scale up to meet the needs of even the largest enterprise. Object Technology gives you precisely the flexibility and responsiveness you need.

The power of visual programming

Visual programming enables you to build your applications by manipulating icons and interface controls, and making connections between them. Each visual part and connection corresponds to a pre-coded, reusable object. In this way, visual programming enables you to focus on your business-specific coding when you create an application.

Visual programming provides a high level of expression that closely models your actual business processes. To use it, you don't need to learn complex programming syntax. And it's a natural model for GUI and report building, because GUIs and reports are visual to begin with.

Construction from parts: Just as a circuit board is manufactured from individual, prefabricated parts, you can combine individual software parts to create application components (which are actually composite parts). Just as a circuit board can then be combined with other complex components to create a computer, you can combine your composite parts into a full-function application.

The same composite parts can be recombined to form different applications. You achieve a consistent look, ease of use, and high application quality. Using visual programming techniques, you can wrapper complex application components and work with them as if they were a single part.

The prefabricated nature of parts also makes it easier to take your application apart and reassemble it as you go through the development process.

How you can harness the power

To ensure your success with visual programming across your enterprise, VisualAge gives you everything you need:

- Language of choice
- Platform of choice
- Interoperability
- Component management

- Extended support
- Partnerships

The key to VisualAge's power is the integration of extended support and customer partnerships with visual programming, languages, platforms, and tools. This is how VisualAge provides you with an end-to-end, proven solution to your application development needs.

Language of choice

VisualAge brings object technology and visual programming to the most popular industrial-strength OO languages, C++ and Smalltalk. For each language, VisualAge offers a complete end-to-end solution. You can construct parts and applications in the language most appropriate to your project, with all the benefits of Object Technology and a common Visual Programming paradigm.

Platform of choice

Today's computing environment requires that your data, application logic, and GUI are available wherever and whenever they are needed, and in the operating system you prefer. Each VisualAge language offers you a range of tools for accessing data wherever it resides in your enterprise and for communicating across your systems. You can use VisualAge products to develop your applications in OS/2, Windows, or AIX, with more platforms coming.

Within the language of your choice, most components are platform independent, so that parts and even entire applications developed on one platform can be ported to another. And development skills are platform independent, too: your developers can create Windows applications one day and OS/2 applications the next.

Interoperability

With OOAD, you have many paths to interoperability, including the easy construction of client-based front ends for your existing server applications. Other paths include dynamic data exchange, a broad range of communications parts, data access parts, and parts that wrapper existing C and COBOL applications. SOM support enables parts to communicate with one another across different languages. And DSOM extends SOM across a distributed environment.

Distributed objects enable a single application to run logic or store data transparently on the desktop, on intermediate servers, or on enterprise servers. You can connect to CICS or IMS systems with VisualAge. And soon, with Smalltalk for MVS, you can use the same business objects across your entire system, from the host to the workstation.

Whatever platform you choose to run on, the VisualAge products fulfill the promise of Open Systems. And IBM is dedicated to providing still more interoperability as part of its committment to a reliable client/server solution.

Component management

Whether your application development team is large or small, VisualAge component management makes it easy for them to share code components and to track changes as they develop new applications. In addition, the extensive VisualAge class libraries make it easy to reuse existing classes and parts.

Extended support

VisualAge gives you access to a wide range of traditional product support sources, such as help lines, online bulletin boards, and lists of frequently asked questions (FAQs). But our VisualAge support goes far beyond this. It's not just support; it's a solution.

You have access to extensive training and educational offerings in products and related technologies. Offerings include high-level overviews of Object Technology and its methodologies, as well as offerings in various approaches to application development with VisualAge. Courses are available world-wide.

Partnerships

Customer partnerships with IBM give you a full spectrum of access to the best of the VisualAge solution. IBM Consulting and Professional Services provide world-wide assistance in such areas as training/mentoring, designing and building reusable componentry, setting up multi-user development environments, and prototyping.

VASTeam members are independent vendors who have become VisualAge specialists to provide you the best in wide-ranging services. In addition, through the Object Connection, IBM brings you into partnership with independent vendors who build "plug-and-play" VisualAge parts for almost every business purpose.

Proven Results

VisualAge is already proving itself as a reliable end-to-end solution across a spectrum of industry sectors. In the past two years, many companies like yours have adopted IBM's pioneering VisualAge products as their strategic application development solution.

Whether their language of choice is Smalltalk or C++, these companies are achieving greater responsiveness to the needs of the marketplace, greater flexibility in developing applications, a quicker time to market, and an overall greater productivity.

Smalltalk case study—an automobile manufacturer: One of the world's largest automotive companies wanted to improve the inventory and finance management systems for their regional offices and dealers. This would improve time to market for their auto parts and speed the customer repair cycle.

Their overall strategy called for a robust application system that would manage their large centralized distribution facility. They wanted the business logic available at client installations where the auto parts were needed, and they also wanted continuous update of key corporate data on their centralized databases. This required a staged migration to a new client/server environment.

Results: Using VisualAge, the company developed all the required function within their aggressive schedule, yielding a distinct competitive advantage. According to their Vice-President of Information Systems, "VisualAge . . . gave us a methodology that I think is pretty revolutionary from the standpoint of classic business application development."

Smalltalk case study—an insurance company: A medium-sized insurance company wanted to streamline its delivery of new products. To successfully target products in the rapidly-changing insurance market, the company required a flexible and sophisticated use of database statistics.

To make this possible, they needed an effective client/server environment that could port business rules from mainframe into client applications, and provide online access from GUI clients to existing host databases. In this way they could tailor their new insurance offerings for precise market segments.

Results: As a result of using VisualAge, the company is now deploying applications at lower expense and in a shorter timeframe. They can deliver new varieties of products because the cost barriers are disappearing. According to the Vice President of their Personal Lines System, "I think the industry is going . . . to see that object-oriented technology is not just the wave of the future; it's here already."

C++ case study—a transportation company: A major transportation company wanted to speed the coupling of freight cars into trains and reduce misrouting. They needed to upgrade the way they tracked the locations and inventories of the cars. Their freight yard operators, located in control booths, needed all the flexibility and user-friendliness that a computer system can offer.

The operators required small, single-task applications that were entirely mouse-driven, to be used with one hand while the other hand was free for additional tasks. And due to the large number of these small applications required, the company needed a fast prototyping-to-delivery cycle.

Results: With VisualAge, the applications came in on time and they met the user requirements to rave reviews.

C++ case study—a government contractor: A government contractor wanted to move its customers from command-line driven applications to a more up-to-date GUI system. This contractor specialized in providing software for organizations ranging from municipalities to the Federal government. Because their customers ranged from the largest to the smallest, the software had to be scalable. And it required a simple design structure to accomodate maintenance and programmer turnover.

Many of the contractor's customers required complex contract-management software. The software had to clearly model customer business rules and governmental regulations. It also required strong accountability on changes and sign-offs.

Results: Using VisualAge, the contractor was able to develop a well-designed, elegant application system that fulfilled all requirements and was fully scalable. And they developed the system on time and on budget.

VisualAge: how it all fits together

The overall story is really quite simple: whatever your language and platform of choice, IBM brings you an integrated development environment based around powerful visual development tools.

The story begins with the Smalltalk edition of IBM's VisualAge. Smalltalk is a logical place to start because it is a pure object-oriented language. Once the Smalltalk edition was established as an industrial-strength application builder with rich functionality, IBM started porting the best of the new techniques and technologies to the world of C++.

The roots of C++ in C make it a popular hybrid object-oriented language that is familiar to existing C programmers. In the same way as Smalltalk optimizes for rapid application development, C++ optimizes for performance.

Once the two editions of VisualAge were established award-winning products, IBM turned its attention to extending the base products with even more powerful function and strengthening their position as power tools for enterprise development.

Both editions of VisualAge combine visual programming and an integrated development environment with the latest in client/server advances. With available parts that extend the base environment, and with a wide range of user services, VisualAge adds up to an application development solution for the '90s and beyond.

Integrated development environment

At the heart of each VisualAge product is an integrated development environment, which consists of the object-oriented language of your choice, combined with extensive part and class libraries, browsers, debuggers, and a state-of-the-art visual application builder.

Visual application builder

The visual application builder is far more than just an interface builder and prototyping tool. It's a power tool for creating industrial-strength applications, many of which do not require you to write any code at all. The VisualAge Composition Editor gives you a visual work surface on which to create your application. You simply select parts from an extensive library and make the appropriate point-and-click connections. Need a new part for a unique purpose? You can create one using the other tools of the integrated development environment.

Parts include everything you need to build GUIs, together with a broad range of application logic. It's easy to create additional application logic using the Smalltalk Script Editor or the C++ code editor. And the composite components you build can be wrappered as individual parts and used "as is" from application to application.

You can add new parts to the VisualAge palette, whether from IBM or from independent vendors, to greatly expand the business function in your applications. And new extensions to VisualAge are available all the time, both from our growing family of “Object Connection” partners and IBM’s own continuous enhancements to the base environments.

The parts palette: VisualAge parts are intuitive. They look like real-world objects. And, in fact, they closely model your existing business objects and practices. The VisualAge parts palette includes categories such as buttons, lists, windows, external functions, and database functions. Inside these categories are parts such as entry fields, sliders, ordered collections, program starters, database queries, and stored procedures.

Parts on the palette can be used over and over. So can the composite parts you build from your basic parts. It’s easy to modify parts and have your new parts display on the palette. You can even create new categories to suit your needs.

Database access: The VisualAge products provide native support for IBM DB2/2 databases, including a complete suite of functions like server-enforced referential integrity, stored procedures, and industry-standard SQL. In addition, VisualAge can be extended to access a range of IBM host databases.

The Smalltalk editions include far-ranging native support for ORACLE databases, bringing you the data access you need on the platform of your choice. Even if you must access multiple databases in your current environment, VisualAge can make the connections with its Open Database Connectivity (ODBC) interface for Smalltalk. ODBC enables you to create applications that access all your data sources, from spreadsheets to sophisticated relational databases like INFORMIX and SYBASE.

Communications and transactions: The VisualAge communications and transactions parts mask the complexity of communicating among system components and even to remote systems. You can access program logic and data by using a wide variety of popular communications protocols and APIs. For Smalltalk, these include APPC, CPI-C, CICS, ECI, IMS, MQSeries, NetBIOS, RPC, TCP/IP, and EHLLAPI (3270 terminal emulation).

Multimedia: Add multimedia to existing applications, or create state-of-the-art new applications. VisualAge Multimedia gives you access to industry-standard interfaces for playback of audio and video files. The multimedia parts include a rich set of visual controls to custom-design your own multimedia applications.

VisualAge in the C++ environment

In addition to the Visual Builder, the C++ development environment includes a rich set of tools such as a Data Access Builder to quickly construct useful data access parts, and a Visual Browser that enables you to edit your source code, as well as look at graphical relationships between program components, online documentation for classes, and types of program objects. Other tools include a state-of-the-art compiler and linker, a language-sensitive editor, a debugger, and performance analyzer.

The environment also includes the IBM Open Class Library, a comprehesive set of building blocks. These classes serve as the basis for all the VisualAge C++ parts, and provide a rich range of function in reusable, reliable, extensible objects.

To get you started quickly with VisualAge C++, the environment includes “Project Smarts,” a set of application frameworks you can use to build your own. These preconfigured applications lead you through the process of learning the development environment. In addition, the VisualAge Workframe enables you to fully customize your work environment, integrating tools of your choice with your VisualAge tools, and using the “Build Smarts” dialog to switch quickly between the most common build options.

VisualAge in the Smalltalk environment

In the Smalltalk edition, the Visual application builder is seamlessly integrated into the development environment. The applications you construct are actually Smalltalk objects, and they compile and link dynamically, so that no additional compile/link tools are needed.

The environment also includes a Script Editor that guides you through the construction of common, useful Smalltalk scripts, and a Public Interface Editor that helps you define the interfaces your VisualAge parts use to communicate with one another.

Browsers give you full access to the underlying Smalltalk code. The Organizer window helps you track your development work and get started using the tools, and a full-function debugger helps you analyze and debug your applications.

A rich variety of parts are available from IBM to extend the base Smalltalk environment. These are mature, tested parts that are easy to integrate into your applications. They include the many database, communications, and transaction parts mentioned earlier. In addition, **External function parts** enable you to wrap your existing C or COBOL code for use in VisualAge applications. **VBX support** does the same for Visual Basic components and applications.

Need reports generated from any of your Smalltalk applications? The VisualAge **Reports feature** provides a dozen report parts to help you create, format, preview, and print attractive and meaningful reports.

The **VisualAge AS/400 Connection** lets you work with AS/400 system resources as visual parts, including database files, data queues, and data areas. Using visual construction techniques, you can connect these parts into powerful client/server applications. You can make use of distributed and host server programming.

With the **VisualAge IMS Connection**, you can move still further into the world of client/server development by communicating between object-oriented client applications on your workstations and traditional System/390 programs on the host. The IMS Connection wraps host transactions into Smalltalk business objects for easy messaging in your workstation or LAN environment.

VisualAge and client/server enablement

VisualAge paves the way to an effortless and powerful client/server computing environment. Even with industrial-strength heterogeneous systems, VisualAge can help you make the connections you need. From the host to the workstation, all around the LAN and at any location where you need your data to run, VisualAge offers you an end-to-end, proven, integrated solution.

Existing Client/Server Assets: The applications you build with VisualAge provide easy access to your existing data and business logic wherever it resides in your system. External function parts make it easy to call existing C or COBOL code from any VisualAge application. With the database parts and communications/transactions parts, the AS/400 Connection and the IMS Connection, you can leverage your existing system resources. VisualAge gives your developers easy visual access to a wide range of communications and database protocols.

Cross-Platform Compatability: Most VisualAge classes are platform-independent; they exhibit a wide range of portability. It is easy to develop on one platform and run in another, or to exchange parts between platforms. You can also repackage or recompile your applications for different runtime platforms.

Distributed Computing: Distributed Computing is the wave of the future, and the VisualAge products are already riding that wave. Distributed objects help developers deal with the complexity of heterogeneous two- and three-tier client/server architectures. By enabling free distribution of function and data based on developer design considerations, distributed objects enable a high degree of scalability and robustness.

VisualAge Distributed enables you to easily develop distributed Smalltalk applications with true local/remote transparency across your system environment. You can partition your applications in the ways that best fit your business needs and you can protect them with industrial-strength security.

VisualAge **SOMSupport** enables C++ objects and Smalltalk objects to communicate with one another, so that applications are no longer limited by language or location. And by adding DSOM, applications can be composed of objects that can reside on different servers, or in entirely different systems.

The Object Connection

The Object Connection program gives you a shortcut to developing the complex and specialized parts that match your business needs. Through the Object Connection, many large and small vendors make specialized parts for VisualAge. These modular, self-contained parts fit seamlessly into the integrated development environment. Object Connection parts include business graphics, chart parts, frameworks for desktop publishing, multinational translation parts, and many more.

IBM encourages vendors to join the Object Connection program and provides the support they need to enable their parts for the VisualAge environment.

The industry-standard solution

IBM's VisualAge family of products conform to the latest industry standards, including the C++ ANSI standard, the proposed ANSI Smalltalk standard, and OMG's Common Object Request Broker Architecture (CORBA). This means that components, parts, and development tools adhere to the highest standards of quality and reliability in the industry. You can have confidence in the lasting value of your applications and your development tools.

VisualAge and you: making the connection

With the VisualAge products, IBM brings you powerful new tools for all your enterprise application development needs. VisualAge connects you to your data and existing applications wherever they reside in your enterprise, from the desktop to the host. And VisualAge connects you to powerful support, services, and partnerships. VisualAge: a true end-to-end solution.

Want to know more?

For further information on IBM's VisualAge products, call the following numbers:

In North America
    1-800-IBM-2279[1]
Outside the U.S. and Canada
    1-919-254-4760
In Europe, the Middle East, and Africa
    33-92-11-50-11

You can also append CompuServe under VisualAge (GO VISUAL).

For VisualAge information on the World Wide Web, go to the IBM Software Home Page:

http://www.software.ibm.com/

and do a search on the keyword **goVisualAge**.

2 After January 1, 1996, call 1-800-IBM-3333 ext. GS059.

IBM®

This publication may refer to products that are not currently available, or may not be available in your country. IBM product plans are subject to change.

This publication could contain technical inaccuracies or typographical errors.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries: IBM, AIX, "A Powerful New Vision of Programming," AS/400, CICS, DB2/2, IMS, MQSeries, OS/2, SOMobjects, "The Object Connection," VisualAge.

The following terms are trademarks of other companies:

C++ (AT&T Corporation);
INFORMIX (Informix Software);
ORACLE (Oracle Corporation);
SYBASE (Sybase, Inc.);
Visual Basic (Microsoft Corporation).

Windows is a trademark of Microsoft Corporation.

IBM's VisualAge products and services are not associated with or sponsored by VisualEdge Software, Ltd.

G325-0801-02

```
+++EDF004W USERDOC tag found within USERDOC and ignored. (Page 10  File: TECH
NOGY SCRIPT)
DSMMOM397I '.EDFDTYP' WAS IMBEDDED AT LINE 820 OF 'TECHNOGY'
+++EDF072E PROLOG tag used more than once - tag ignored. (Page 10  File: TECHNOGY
 SCRIPT)
DSMMOM397I '.EDFPRLG' WAS IMBEDDED AT LINE 821 OF 'TECHNOGY'
+++EDF002W DOCPROF tag found outside PROLOG. (Page 10  File: TECHNOGY SCRI
PT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 822 OF 'TECHNOGY'
+++EDF002W TITLE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 827 OF 'TECHNOGY'
+++EDF002W LIBRARY tag found outside TITLE. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 828 OF 'TECHNOGY'
+++EDF002W TITLE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 828 OF 'TECHNOGY'
+++EDF002W TITLE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 829 OF 'TECHNOGY'
+++EDF002W TITLE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 831 OF 'TECHNOGY'
+++EDF002W DATE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 832 OF 'TECHNOGY'
+++EDF011W PROLOG end-tag found outside PROLOG and ignored. (Page 10  File: TEC
HNOGY SCRIPT)
DSMMOM397I '.EDFEPRLG' WAS IMBEDDED AT LINE 836 OF 'TECHNOGY'
+++EDF009W Duplicate ID PHONE was ignored on FN tag. (Page 9  File: TECHNOGY S
CRIPT)
DSMMOM397I '.EDF@ID' WAS IMBEDDED AT LINE 470 OF '.EDFFTNT'
DSMMOM397I '.EDFFTNT' WAS IMBEDDED AT LINE 1576 OF 'TECHNOGY'
DSMBEG323I STARTING PASS 2 OF 2.
+++EDF004W USERDOC tag found within USERDOC and ignored. (Page 10  File: TECH
NOGY SCRIPT)
DSMMOM397I '.EDFDTYP' WAS IMBEDDED AT LINE 820 OF 'TECHNOGY'
+++EDF072E PROLOG tag used more than once - tag ignored. (Page 10  File: TECHNOGY
 SCRIPT)
DSMMOM397I '.EDFPRLG' WAS IMBEDDED AT LINE 821 OF 'TECHNOGY'
+++EDF002W DOCPROF tag found outside PROLOG. (Page 10  File: TECHNOGY SCRI
PT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 822 OF 'TECHNOGY'
+++EDF002W TITLE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 827 OF 'TECHNOGY'
+++EDF002W LIBRARY tag found outside TITLE. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 828 OF 'TECHNOGY'
+++EDF002W TITLE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 828 OF 'TECHNOGY'
+++EDF002W TITLE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 829 OF 'TECHNOGY'
+++EDF002W TITLE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 831 OF 'TECHNOGY'
+++EDF002W DATE tag found outside PROLOG. (Page 10  File: TECHNOGY SCRIPT)
DSMMOM397I '.EDF#CNTX' WAS IMBEDDED AT LINE 832 OF 'TECHNOGY'
+++EDF011W PROLOG end-tag found outside PROLOG and ignored. (Page 10  File: TEC
HNOGY SCRIPT)
DSMMOM397I '.EDFEPRLG' WAS IMBEDDED AT LINE 836 OF 'TECHNOGY'
```