



IBM System *i*™

Moving RPG to an SOA Framework; A Step by Step Overview

Alison Butterill
butteril@ca.ibm.com

*i want stress-free IT.
i want control.
i want an *i*.*

Simplify IT

© 2006 IBM Corporation

IBM System *i*



Acknowledgements and Disclaimer

- **Acknowledgment:**
 - This presentation is a collaborative effort of many IBM presenters
 - This presentation includes some material from *Paul Tuohy of ComCon, Dublin, Ireland* and *Susan Gantner of Partner400, Toronto, Canada*.
- **Disclaimer:**
 - The information contained in this document has not been submitted to any formal IBM test and is distributed on an as is basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customers' ability to evaluate and integrate them into the customers' operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.
- **Reproduction:**
 - This presentation is the property of the IBM Corporation, ComCon and Partner400. Permission must be obtained **PRIOR** to making copies of this material for any reason.

*i want an *i*.*

© 2006 IBM Corporation

Why SOA?

Does this sound like applications in your shop?

- Have difficulty in adapting applications for business changes
- Functionality within applications is not modularized to support reusability
- High costs, time consuming and lacks integration across business units
- Application portfolio misaligned to business strategy
- Proprietary applications are fragmented and unable to “talk to one another”
- Lack enterprise-wide, cross-data sharing capabilities
- IT resources are tied up with no bandwidth for new projects
- Uncertainty about the long-term role of the existing portfolio
- Difficulty linking applications to modern technologies

Does Implementing SOA Help?

Why Modernize?

The interface is changing.

- The preference is for GUI as opposed to green screen.
- Integration with new architectures requires modular code

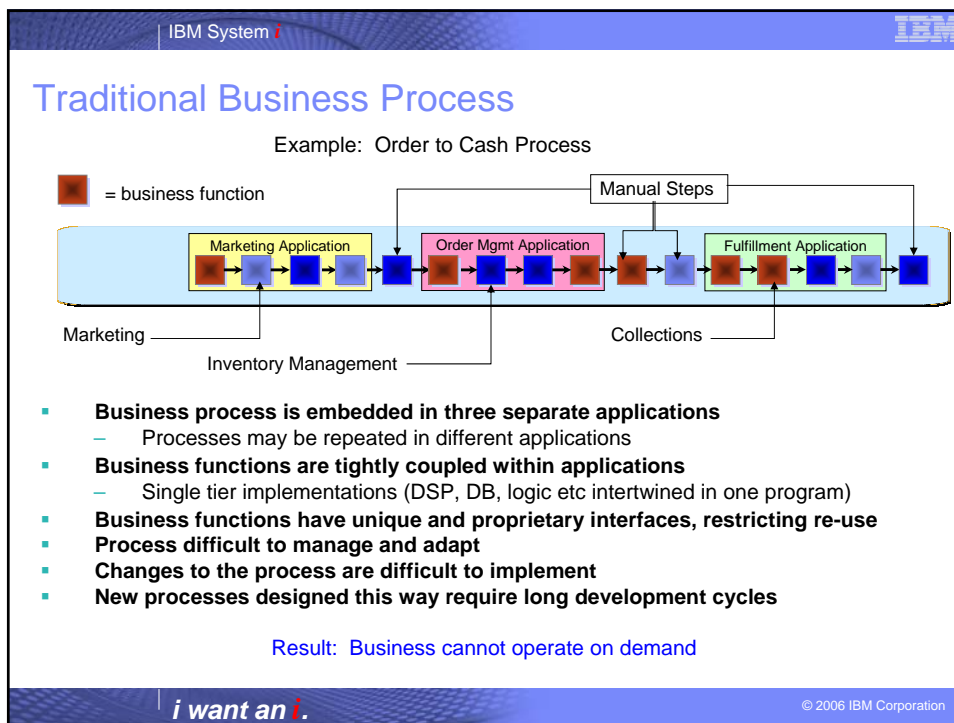
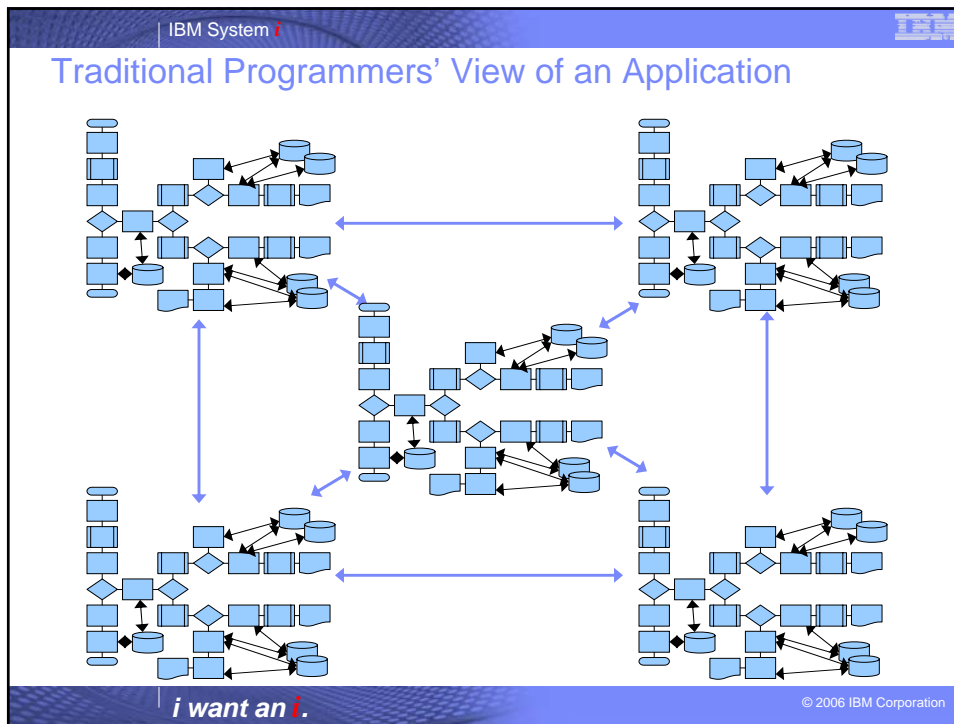
The Choices

- Get new applications.
- Modernize current applications.

Considerations

- Protect investment in current working code.
- Protect investment is staff who understand the *business* application.
- Protect investment in staff who understand the code.
- Reduce the maintenance overhead.

Does Implementing SOA Help?



IBM System i

Step 1: Customer Self Service

- Customers now order online using a web browser and the Internet
- Business partners can order using a web service call from their own process
- Customers are better served
- Reduces requirement for additional staff to take more orders


i want an i. © 2006 IBM Corporation

IBM System i

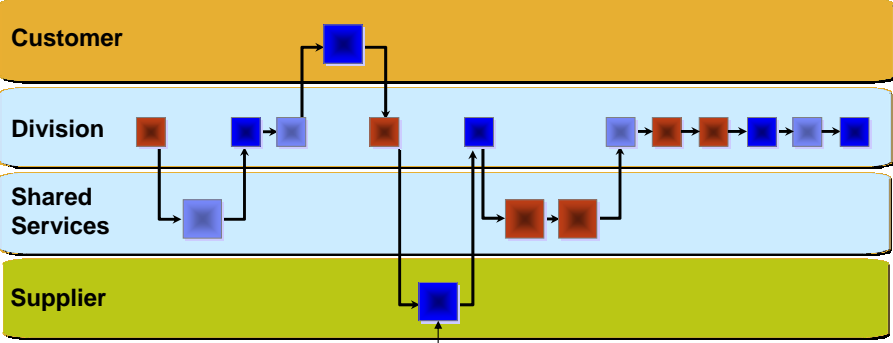
Step 2: Remove Redundant Code

- Common business functions are shared across the enterprise
- Marketing, Billing, and Receivables are handled uniformly
- Enterprise can scale across divisions and lower costs

i want an i. © 2006 IBM Corporation


IBM System 

Step 3: "Just in Time" Inventory

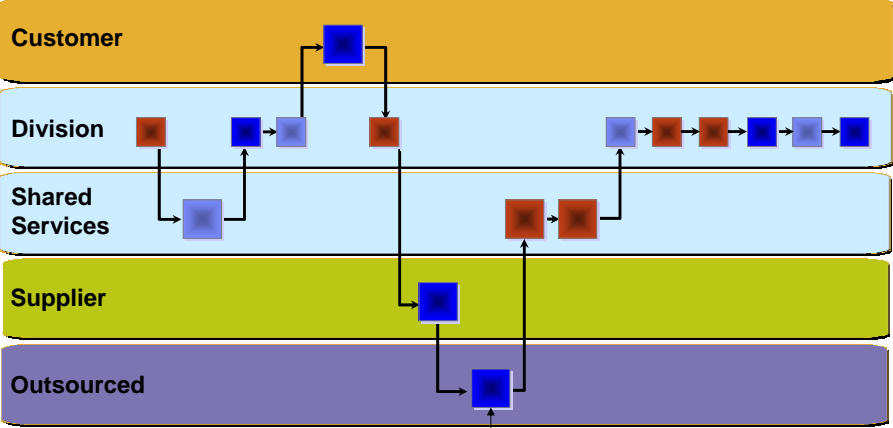


- Minimize or eliminate inventory management function
 - Suppliers responsibility to ship when inventory gets low
- Costs are reduced because less inventory is needed
- Inventory servicing is better because of supplier integration with the process

i want an i. © 2006 IBM Corporation

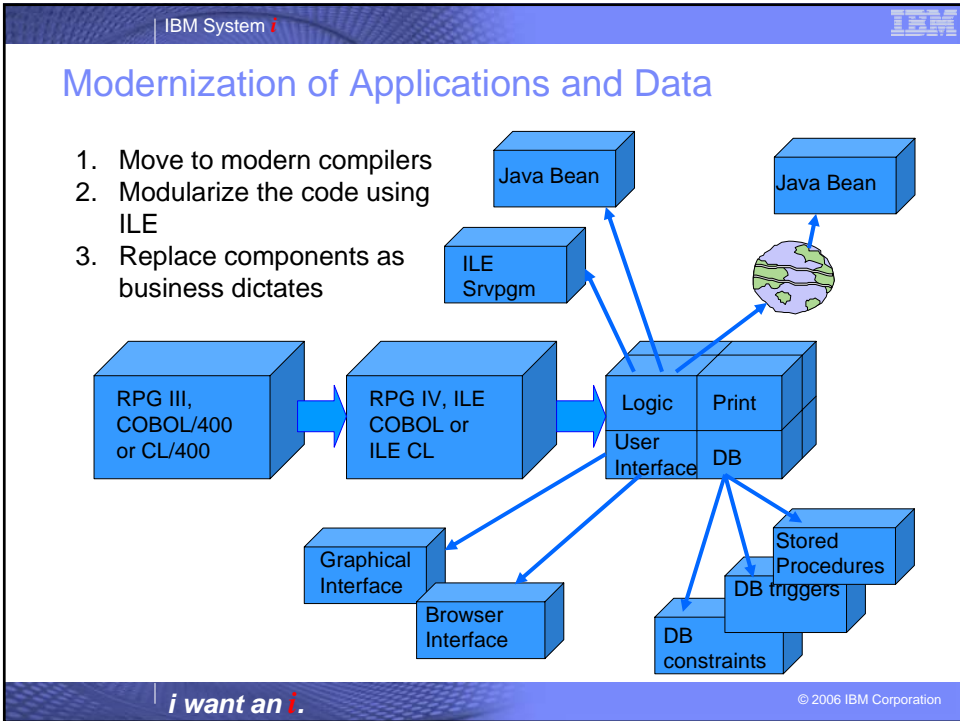
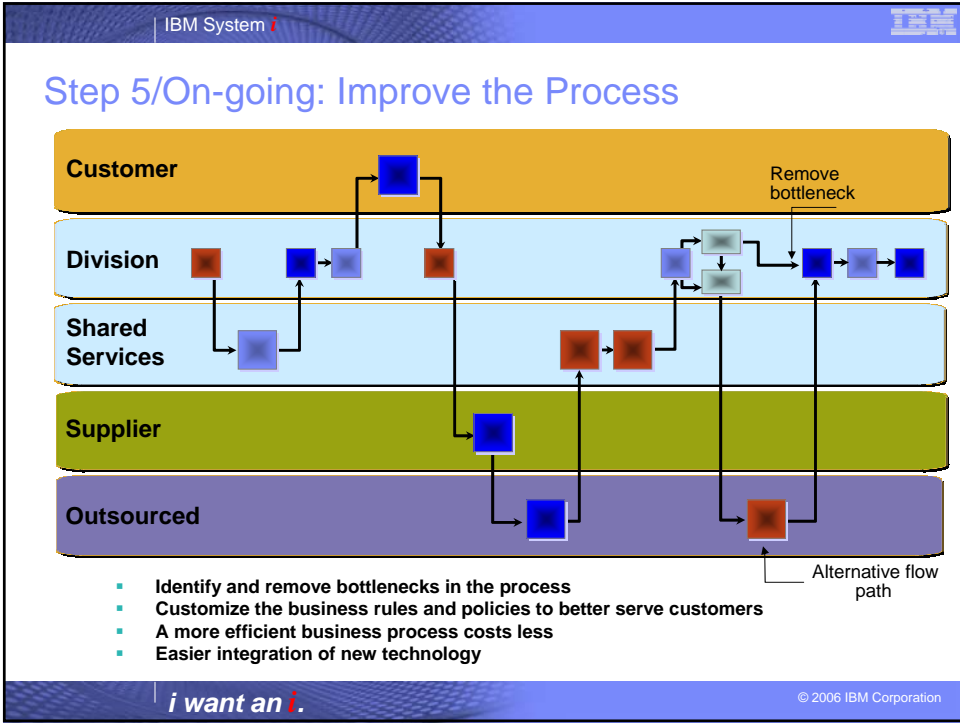
IBM System 

Step 4: Examine Viability and Off-Load as Required



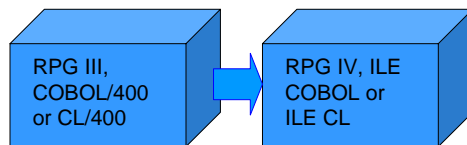
- Shipping is not a core competency
- Shipping companies (e.g. FedEx, DHL, UPS) have more capabilities
- Reduce shipping infrastructure and overhead costs
- Improve customer satisfaction - shipping/handling costs more accurate, maybe reduced

i want an i. © 2006 IBM Corporation



Modernization of Applications and Data

1. Move to modern compilers
2. Modularize the code using ILE
3. Replace components as business dictates



RPG

- Procedures
- Local variables
- Integration with Java
- Extended names, expanded limits
- Ability to work with XML files

COBOL

- Call to procedures
- Expanded limits
- Ability to work with XML files
- Current language standards

CL

- New logic control commands
- Expanded number of files supported
- Variables

Conversion to Modern Compilers

- **CVTRPGSRC – RPG III syntax to RPG IV syntax**
 - Converts to new specification layout
 - One conditioning indicator per line
 - Arrays converted to D-Specs
 - CODE/400 has the CVTRPGSRC option
 - Can view converted source prior to “saving” to the server
- **COBOL/400 to ILE COBOL**
 - Recompile as ILE COBOL
- **CL to ILE CL**
 - Recompile as ILE CL

A Word on Conversion

- **CVTRPGSRC only does a *syntax* conversion**
 - It will work – almost 100%
 - No logic flow changed
- **COBOL and CL re-compiles**
 - No logic flow changed
- **Third party conversion tools offer re-engineering:**
 - Redefine data structures' fields by reordering them naturally, converting from/to positions to actual lengths, and indenting subfields.
 - Redefine *LIKE DEFN-defined fields in the D-specs.
 - Redefine C-spec-defined fields in the D-specs.
 - Redefine any of the ADD, SUB, MULT, DIV, Z-ADD, Z-SUB, MOVE, MOVEL, etc. operations as free-form EVAL operations.
 - Convert Operation Codes to their free form equivalent.
 - Perform case conversion to either lower case or mixed case.
- **Third party and local conversion GREATLY increase the likelihood of testing requirements.**
- **Important to understand the tool and its conversion process**

Conversion – Vendor Tools

- **The programs have already been converted to RPG IV.**
- **Using the RPGWIZ conversion command in the RPG Toolbox from Linoma Software.**
 - Converted to Mixed Case.
 - Converted to Extended Factor 2.
 - Beware! You may not get the same results.
 - Field definitions moved to D Specs.
- **Using a local conversion program.**
 - Added a copy directive for a standard H SPEC.

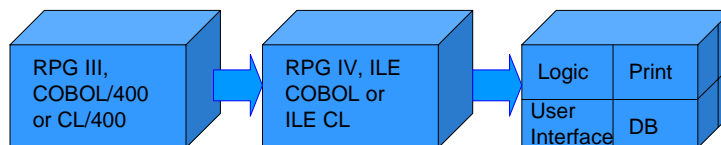
```
H Debug DatEdit(*MDY/) Option(*SrcStmt:*NoDebugIO)
```

After Conversion - The Next Step

- **Start using BIFs.**
 - BIFs offer powerful coding methods.
 - Prepare you for later techniques
 - Subprocedures – writing your own BIFs.
- **Get rid of conditioning RPG Indicators (left and right)**
 - No longer required.
 - Take the opportunity to make indicators self explanatory.
- **Use Prototyped Calls.**
 - Have the compiler validate your parameters.
 - Get rid of parameter lists.
 - Make calls self documenting.
- **What about Free Form RPG?**

Modernization of Applications and Data

1. Move to modern compilers
2. Modularize the code using ILE
3. Replace components as business dictates



Modular Design

- Code reuse
- Improved quality of code

Maintenance

- Less maintenance repetition
- Easier to determine location of required changes
- Compiles are faster?

Distribution of business function

- Can distribute code as required

IBM System *i*

Integrated Language Environment

OS/400

ILE

JOB

OPM

- Dynamic or static binding
- One environment integrates all ILE languages
 - both traditional and procedure based
- Environment provides run-time behaviour
- OPM still available

i want an i.

© 2006 IBM Corporation
Copyright IBM Corporation, 1999, 2001

IBM System *i*

Binding: By Copy and By Reference

*PGM A

*MODULE A *MODULE B

CALLB B CALLPRC P2

RPG CL

Bind by Copy

Bind by Reference

*SRVPGM S

*MODULE X PRC X

*MODULE Y PRC Y

*MODULE Z PRC P1 PRC P2

Multiple Call Entry Points

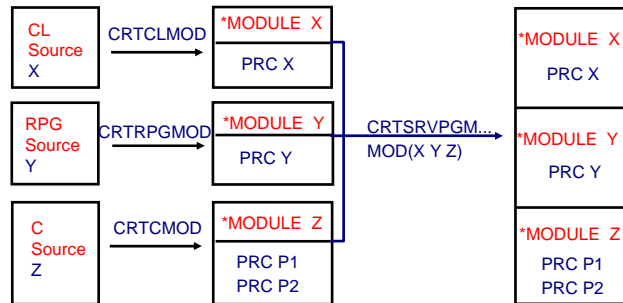
- Both are static (bound) calls
 - static calls are significantly faster than dynamic
- Completion of reference bind on 1st call to *PGM A
 - dynamic bind to *SRVPGM S
 - then, all bound calls to 'S' are static

i want an i.

© 2006 IBM Corporation
Copyright IBM Corporation, 1999, 2001

Service Programs

- *SRVPGM object type
- A collection of commonly used modules packaged together
- A kind of "subroutine library"
- Uses fast static call - static bind by reference
- Cannot be called dynamically
- Multiple entry points



RPG IV Subprocedures

- This is the RPG IV name for a User-Defined Procedure or BIF
- Subprocedures can:
 - Define their own local variables
 - This provides for "safer" development since only the code associated with the variable can change its content
 - Access Global variables
 - Those defined in the main body of the source
 - This includes access to any files defined in the program
 - Be called recursively
- The Compilation Unit can have any number of subprocedures
 - Each must have its own prototype
- By "Compilation Unit" we mean those source lines processed by the compiler in a single compilation
 - This includes any /COPY members

IBM System i

Taking Inventory

DATABOROUGH
legacy enhancement solutions

- Impact Analysis
- Structure Charts
- Data Flows
- Data Modelling
- Pseudo Code
- Word & Visio Export
- Business Rules
- Drill-down Inquiry

REDP4046 - Modernizing and Improving the Maintainability of RPG Applications Using X-Analysis Version 5.6

i want an i. © 2006 IBM Corporation

IBM System i

Modernization of Applications and Data

1. Move to modern compilers
2. Modularize the code using ILE
3. Replace components as business dictates
 - User interface – Web GUI, Rich client

ReFacing

- Translation of data stream

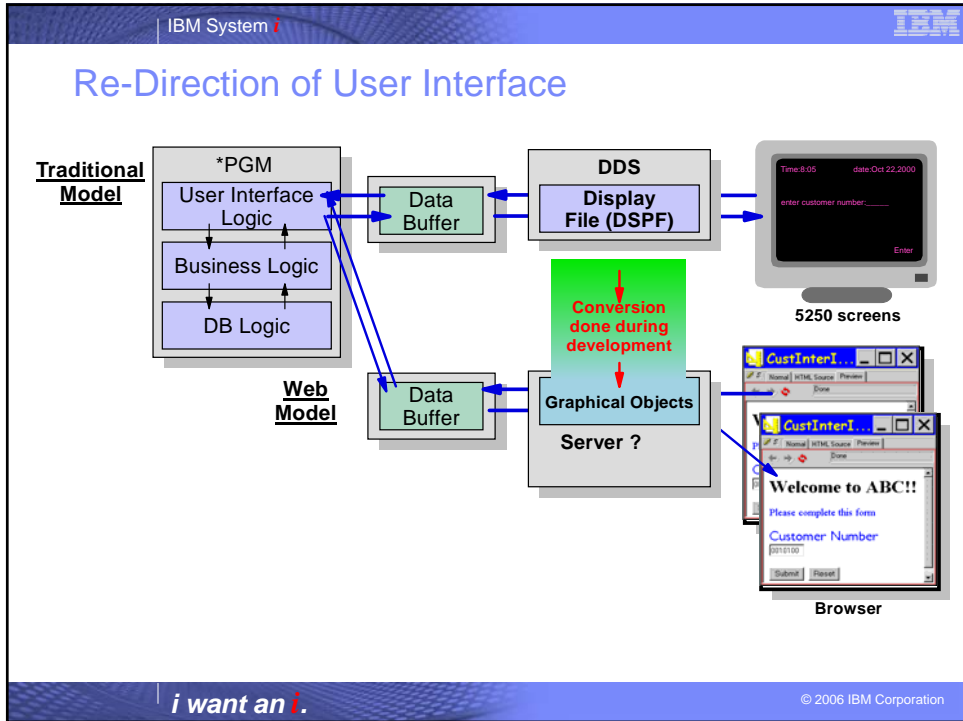
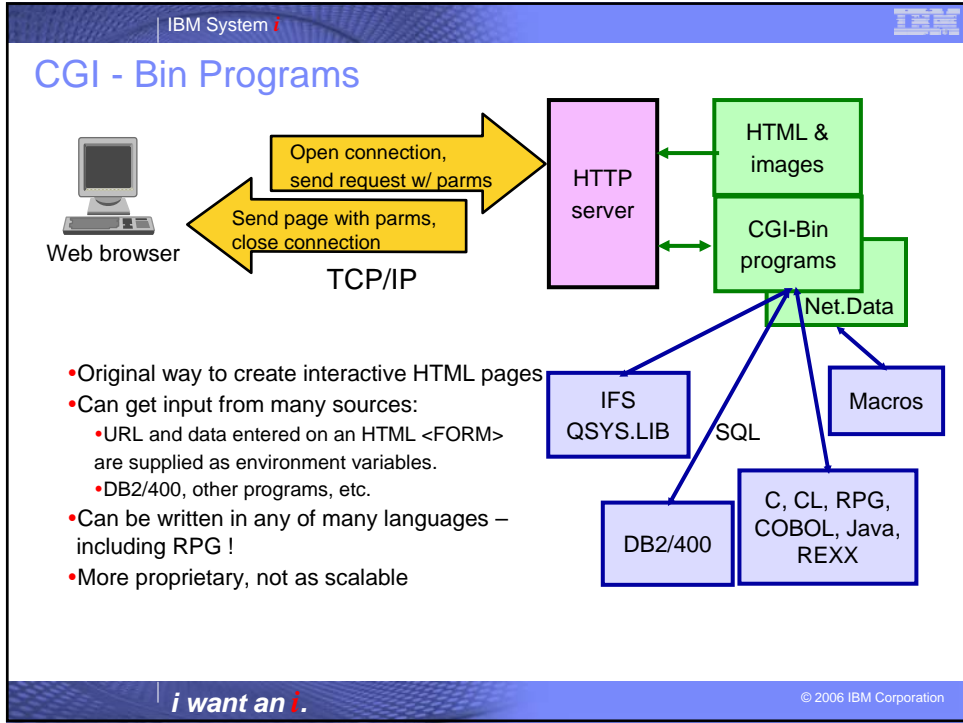
Re-direction

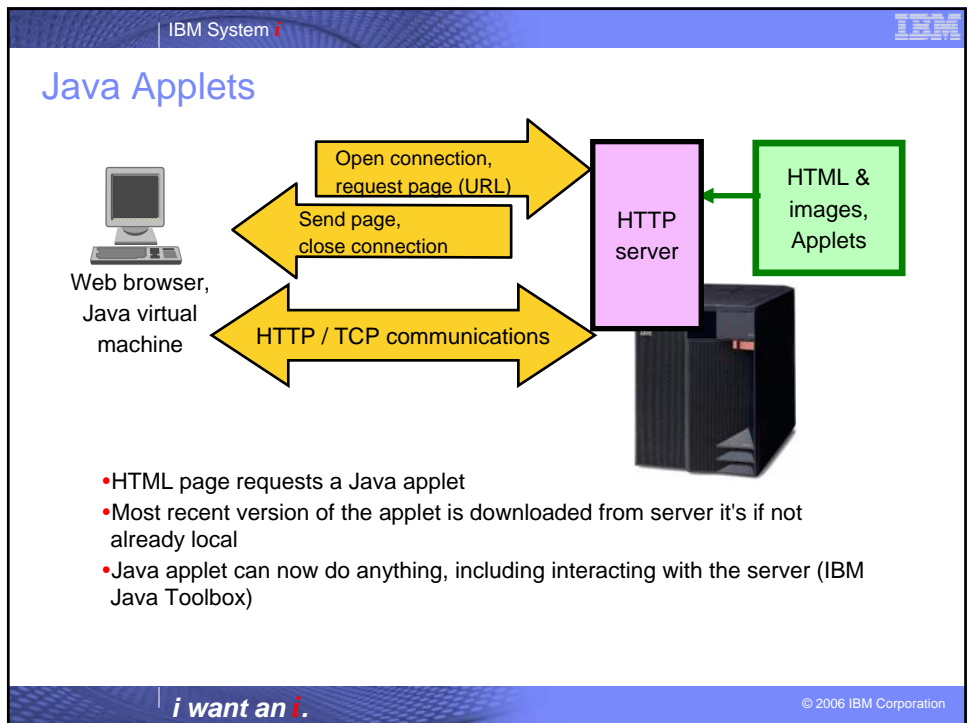
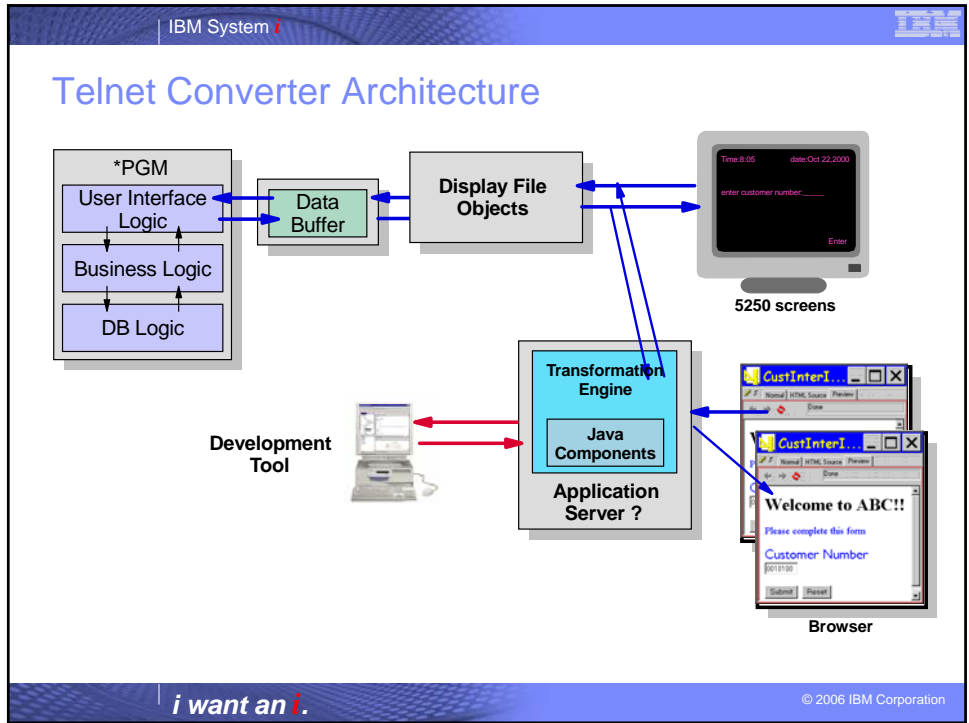
- Alternate interface with re-direct intercept

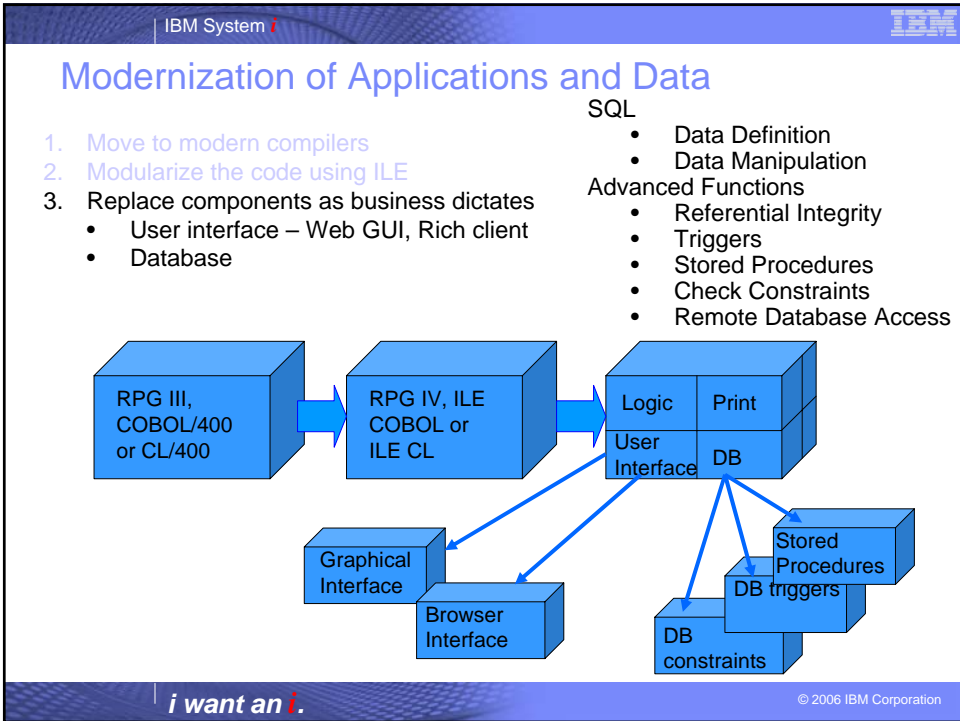
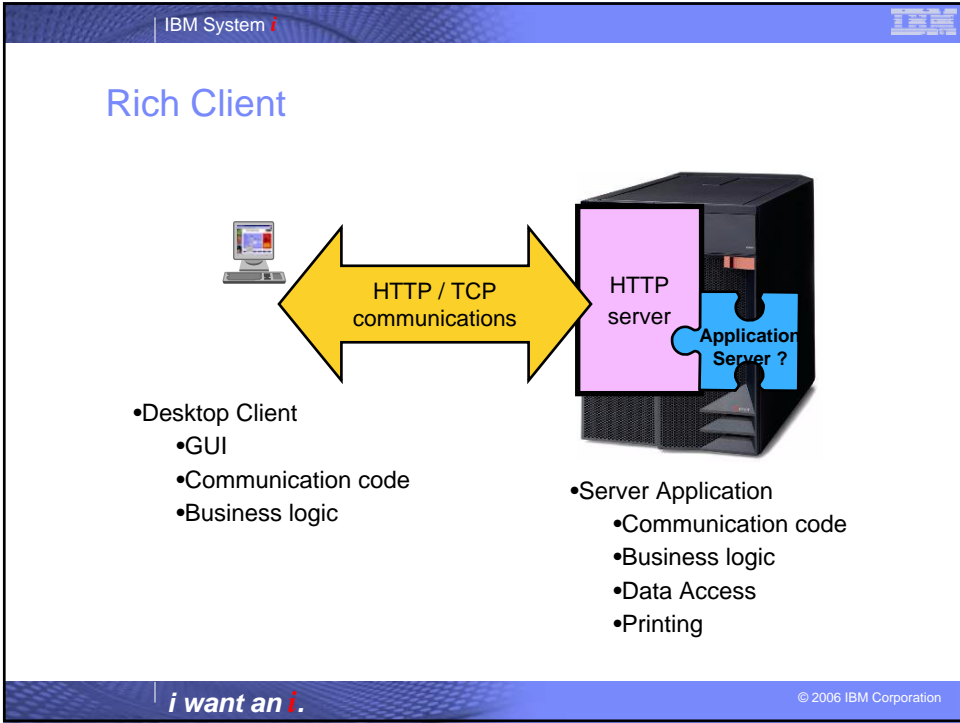
Crafting of new

- CGI-bin
- Java – JSP, JSP, rich client

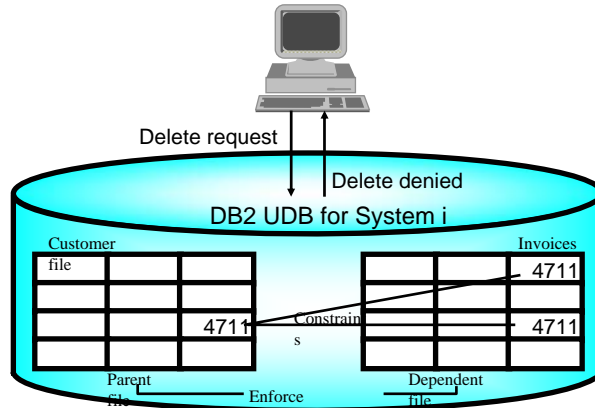
i want an i. © 2006 IBM Corporation







What is Referential Integrity?



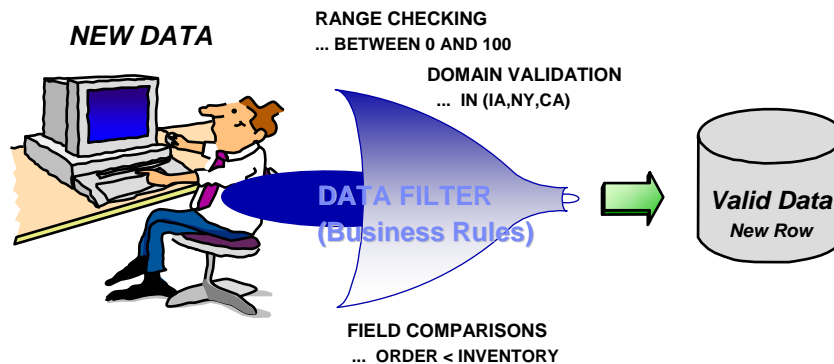
- A capability provided by the database management system to ensure
 - logical consistency of data values between files
 - validity of data relationships
 - robust enforcement of integrity constraints

i want an i.

© 2006 IBM Corporation

Check Constraints

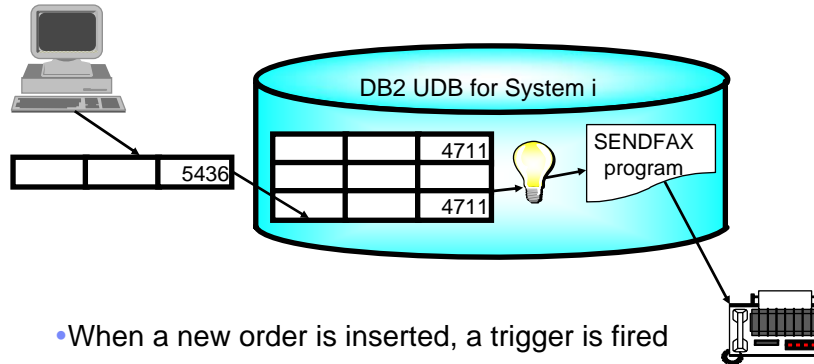
What are they?



i want an i.

© 2006 IBM Corporation

Triggers: An Example



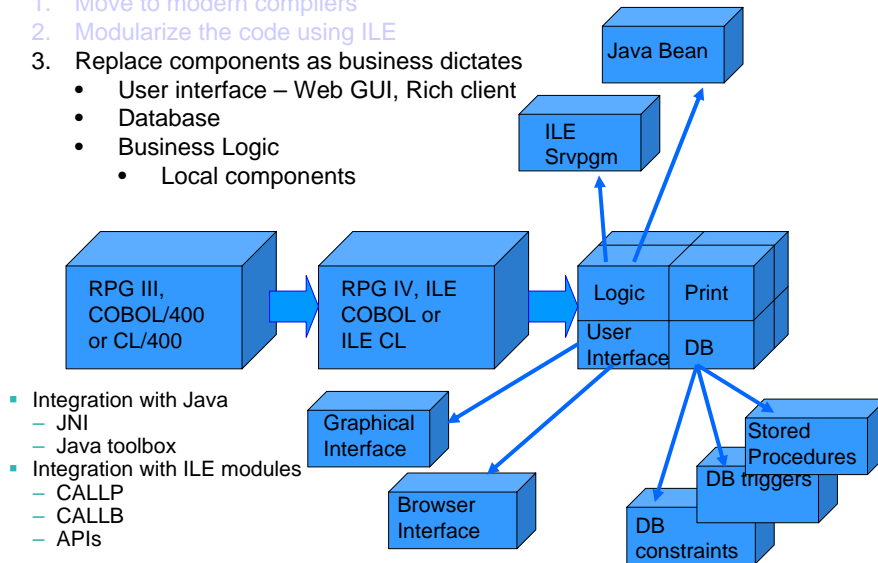
- When a new order is inserted, a trigger is fired
- Trigger retrieves information about order and customer
- A confirmation fax is automatically sent

i want an i.

© 2006 IBM Corporation

Modernization of Applications and Data

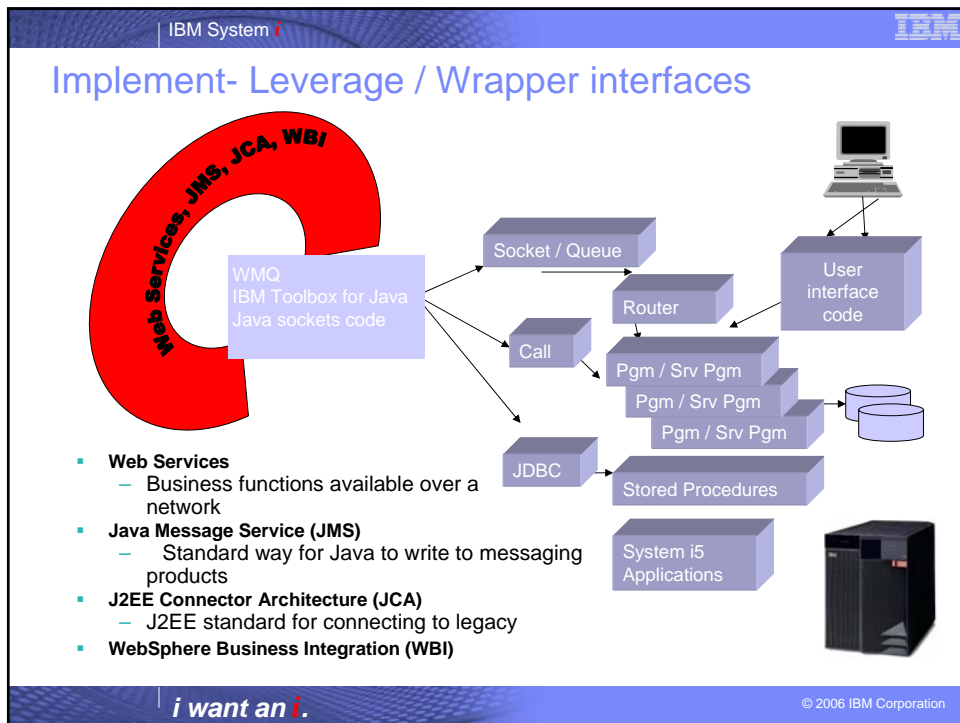
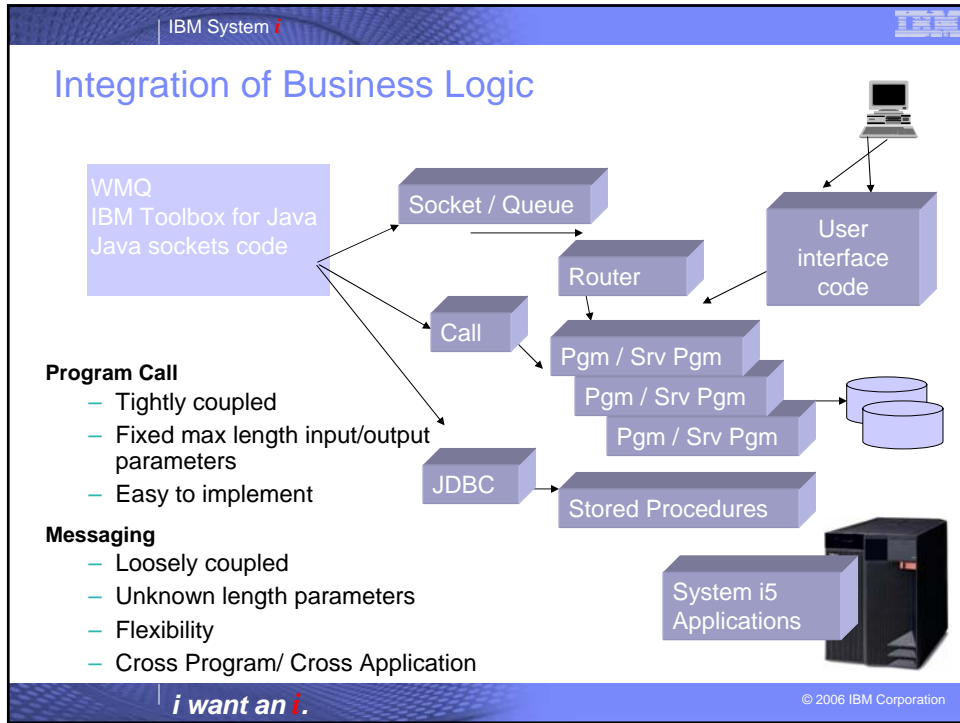
1. Move to modern compilers
2. Modularize the code using ILE
3. Replace components as business dictates
 - User interface – Web GUI, Rich client
 - Database
 - Business Logic
 - Local components



- Integration with Java
 - JNI
 - Java toolbox
- Integration with ILE modules
 - CALLP
 - CALLB
 - APIs

i want an i.

© 2006 IBM Corporation



IBM System i

Modernization of Applications and Data

1. Move to modern compilers
2. Modularize the code using ILE
3. Replace components as business dictates
 - User interface – Web GUI, Rich client
 - Database
 - Business Logic
 - Local components
 - Remote components

Web Services

- Java
- RPG, COBOL
- Other

XML

- Transfer of data

Transaction Processing

i want an i.

© 2006 IBM Corporation

IBM System i

Integration of Applications

Problem:

- Integrating software applications across multiple operating systems, programming languages, and hardware platforms is
 - Difficult
 - Not something that can be solved by any one particular proprietary environment
- Traditionally, the problem has been one of tight-coupling
 - One application that calls a remote network is tied strongly to it by the function call it makes and the parameters it requests
 - Fixed interface to access remote programs or data, with little flexibility or adaptability to changing environments or needs
- Web services technology allows applications to communicate in a platform, and programming language independent manner

Solutions:

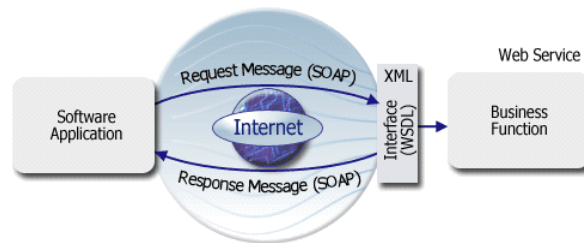
- Web Services
- Other XML
- Other Solutions

i want an i.

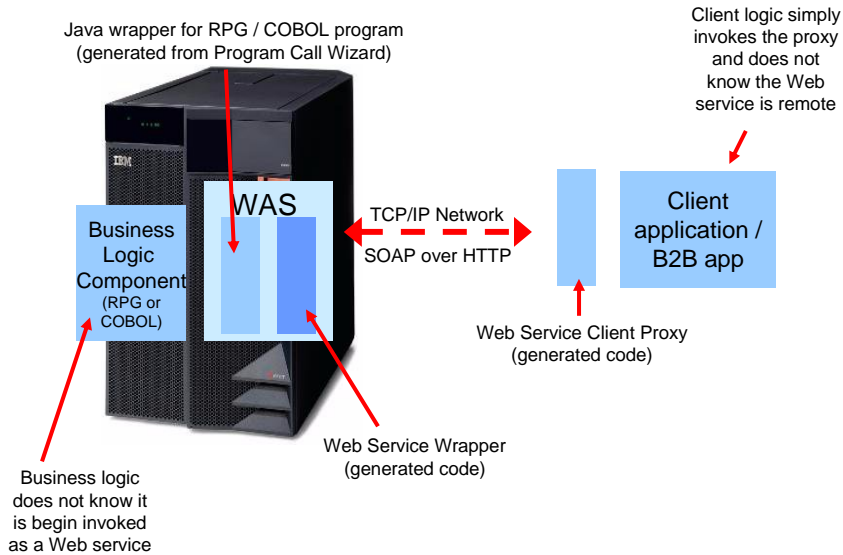
© 2006 IBM Corporation

What are Web Services?

- Web services are self-contained software components, with a well-defined interface
- Extensible Markup Language (XML) is a standardized language used for interface description
- Web services use protocols based on XML to describe
 - operations that can be executed
 - data that can be exchanged with other programs or web services



Overview With System i



IBM System i

Summary

1. Move to modern compilers
2. Modularize the code using ILE
3. Replace components as business dictates

i want an i.

© 2006 IBM Corporation

IBM System i

Trademarks & Disclaimers

© IBM Corporation 1994-2005. All rights reserved.
 References in this document to IBM products or services do not imply that IBM intends to make them available in every country.
 The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	IBM (logo)
AS/400e	iSeries
e (logo) business	OS/400
IBM	

Lotus, Freelance Graphics, and Word Pro are registered trademarks of Lotus Development Corporation and/or IBM Corporation.
 Domino is a trademark of Lotus Development Corporation and/or IBM Corporation.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.
 Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
 Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
 ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.
 UNIX is a registered trademark of The Open Group in the United States and other countries.
 SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.
 Other company, product and service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

i want an i.

© 2006 IBM Corporation