

Store Integrator for Partners

Document Number (Store Integrator for Partners.DOC)

Version Code: (V1.0.0)

July 14, 2004

Address questions to:

Boyd Dimmock
(919) 301-5713
bkd@us.ibm.com

Contributors

- Raja Das, Architecture, SWG
- Boyd Dimmock, Architecture, RSS
- John Fluke, Architecture, RSS
- Mike Johnson, Architecture, RSS
- Michael Maximilien, Architecture, RSS
- Bill Noonan, Architecture, RSS
- Mark Ring, Architecture, RSS
- Paul Ruocchio, Architecture, RSS
- Rick Shryock, Architecture, RSS
- Gaines Whitley, National Retail Services Center
- Ken Wilson, Architecture, SWG
- Ginny Wise, Architecture, RSS
- Fil Yeskel, Architecture, SWG

Archival Requirements

None. This document is permanently in use and has no logical end or completion date. Old versions will be replaced and not archived.

Document Availability

This document is available to the Retail Store Solutions personnel on the Architecture data base.

Quality Records

Record Name	Available	Retention Period
Review Record.	Maintained and controlled by the document owners	Until Superseded
Approval Record.	Maintained and controlled by the document owners	Until Superseded
Notification Record.	Maintained and controlled by the document owners	Until Superseded

Summary of Changes

Changes resulting in document revisions will be summarized in this table in reverse chronological sequence. Revision bars (|) will highlight the text changed in new document versions.

Version	Approval Date	Change Description
V1.0	June 14, 2004	Initial SI White Paper

TABLE OF CONTENTS

1	Store Integration Framework Value	4
1.1	Introduction	4
1.2	Objectives	4
1.3	Retail On Demand Focus	5
1.3.1	IBM Middleware	7
1.3.2	Store Integrator	8
2	Store Integration Framework Reference Architecture	9
2.1	Enterprise	9
2.2	Store	13
2.3	Store Integrator	15
2.3.1	Device Software Components	16
2.3.2	POS Terminal	17
2.3.3	Store Integrator POS Application Extension Facility	18
2.3.4	In-store Controller	19
2.3.5	Store Web Applications	20
3	Solution Lifecycle Management	22
3.1	Introduction	22
3.2	Solution Architecture	23
3.2.1	Solution Architecture Strategy	24
3.3	In-Store System Management Strategy	25
3.3.1	Management Sub-Agent	26
3.3.2	Management Agent API	26
3.3.3	Management Agent	26
3.3.4	Management Application	27
3.3.5	Solution Install and Upgrade	27
3.3.6	Operations Administration	29
3.3.7	Monitoring and Troubleshooting	31
4	Solution Opportunities	32
4.1	Customer Loyalty / Relationship Management	32
4.2	Price Checker Function	34
4.3	Kiosk Integration with POS	34
4.4	Returns Management	34
4.5	Virtual Inventory	34
5	Performance Capacity and Retail Environments	34
5.1	Traditional retail environments	34
5.2	Web application measurements	35
5.3	Connection Bandwidth	35
5.4	Hardware configurations	36
5.4.1	4690 OS Terminal:	36
5.4.2	4690 POS Controller/Terminal:	36
5.4.3	In Store Processor / Web server:	36
5.5	Data Currency	37
5.6	Response times	37
5.6.1	Traditional POS response time requirements	37
5.6.2	Remote data access for new business function	37
6	Store Integration Framework Glossary	38

1 Store Integration Framework Value

1.1 Introduction

IBM is delivering an architecture for bringing web technology based solutions to the retail industry, building on the strengths of IBM retail store products and middleware and embracing the strengths of our business partner network. IBM offers a software framework of building blocks for integration of heterogeneous systems with POS function and the seamless mapping of store operations into the enterprise. The framework focuses on enhancing solutions that deliver a satisfying shopping experience to consumers, empowering employees and streamlining store operations. A key value is delivering a common view of the retailer to the consumer based on the following four components:

- support for multiple touch points for store services, including new consumer-interaction devices;
- integration of store operations with enterprise processes, data and applications;
- linking store systems with multi channel retailing; and
- extending custom, legacy point of sale applications using Java and web application programming paradigms.

This whitepaper defines the architectural components which describe this new offering and focuses on IBM Store Integrator as a key deliverable.

Audience

This document is intended for IBM business partners to use to understand the Store Integration Framework. It will be useful for architects, software developers and technical marketing support. This information lays out IBM's vision of the solution and provides technical detail on the solution components and building block products.

Scope

The scope of the document is to include the broader solution and selection of products for retail including the central enterprise location, the store, and access to other internet resources. While the future direction is always an important factor, the focus of this document is on function and features which are planned to be generally available within the next 12 months.

1.2 Objectives

The objective of this retail store framework is to meet the needs of the retail industry for innovative solutions, and to address the pain points in today's environment. The fundamental forces that are shaping the retail industry drive the architecture. The architecture provides solutions to address these four key issues.

- **Consumers:** Today's consumers are more value driven than ever before. They are technology accepting, and use technology to make informed purchasing decisions. They have high expectations for both the value of the goods and the shopping experience. They are less loyal to a specific store chain or brands.
- **Competition:** The competition between retailers is a fight to survive. The industry is being shaped by consolidation of chains within mega-chains and blurring of

market segments. Retailers are investigating ways to differentiate themselves from their competitors and to draw consumers based on one-to-relationship building. Retail competes within multiple segment categories (supermarket, toy store, etc.) as well as multiple format stores (Supercenters, drive through supermarkets). Retailers compete but on low price as well as service model. The competition as low cost provider is driving retailers in general to higher value and deeper service strategies to compete.

- **Productivity:** The name of the game in store management is productivity. New business systems for stores must come with both real ROI as well as strategic advantage. Retailers are looking for technology which helps their store workforce be more efficient. Excellent store management is the key to store performance, so retailers are looking for technology (store management tools) to help drive the best processes into the store operation. Process integration is important as a tool to simplify store operations, to make them more standard and reproducible, and to integrate store operations with new enterprise solutions.
- **Technology:** Retailers are looking for ways to use technology to provide reliable, accessible systems to be able to quickly react to market initiatives. Standards are key to availability of solutions that integrate between best of breed suppliers. Emerging technologies are beginning to transform the instore experience for both shoppers and employees. Kiosk and self checkout are becoming widely deployed. Consumer service devices are being piloted to demonstrate value in building one-to-one relationships between retailers and their shoppers and to make the shopping experience more appealing. Other technologies such as cashless payment systems and RFID tags to streamline the supply chain are being investigated for future store solutions. Digital signage has potential to transform the in-store messaging to consumers and is being widely evaluated.

1.3 Retail On Demand Focus

The Retail On Demand EBO focuses on three actions to deliver value to retailers. These are the thrust to

- 1) provide a satisfying shopping experience to consumers,
- 2) to empower employees and
- 3) to streamline store operations.

These three initiatives are supported by a series of scenarios which add value to the store operations and enterprise integration.

The value to the retailer will be the enhanced customer service and efficiency advantages provided with the Store Integration Framework. The functions which are enabled will need to provide real return-on-investment for the retailer. This could include increased sales due to enhanced customer loyalty and assisted shopping (improved customer conception of value), or accounting savings in cash management for return transactions. The Store Integrator component of the framework is an enabler which reduces the knowledge and skills that a solution provider must have in order to extend the function of IBM's legacy POS systems. It will help a retailer to extend the life and capabilities of a preexisting 4690 Operating environment.

Building shopper loyalty is provided by delighting the customer with their in-store experience. Shoppers expect more information about products and special deals that apply to them specifically. Providing for promotional offers on a one-to-one basis can increase the shopping basket size as well as consumer satisfaction. Consumer service devices and in-store kiosks provide interfaces to reach the shopper with relationship building information.

Empowerment of employees includes reaching into the store with easily useable training, access to personal data, and communications to the employee to improve productivity.

Applications can span providing information for assisted selling and clienteling, as well as store operations such as store layout and stocking data.

Streamlining store operations can provide savings in better management of stock deliveries to stores, and better recognition of stock in the store. Simplifying the cost and time of adding new application functions into the store reduce the cost of running the store.

The architecture to support these scenarios includes components that are part of the larger network of operations for the retailer. The architecture is built from components that deliver User Interaction services, Application services, Data Management services, Enterprise System Management for Stores and Store Integration services.

The architecture definition is driven by scenarios which demonstrate the concepts of integrating web access with in-store systems as well as consumer interaction technology. Additional business applications should be included in solutions available to the customer, either from IBM services or through partnerships with ISVs encouraged by this framework.

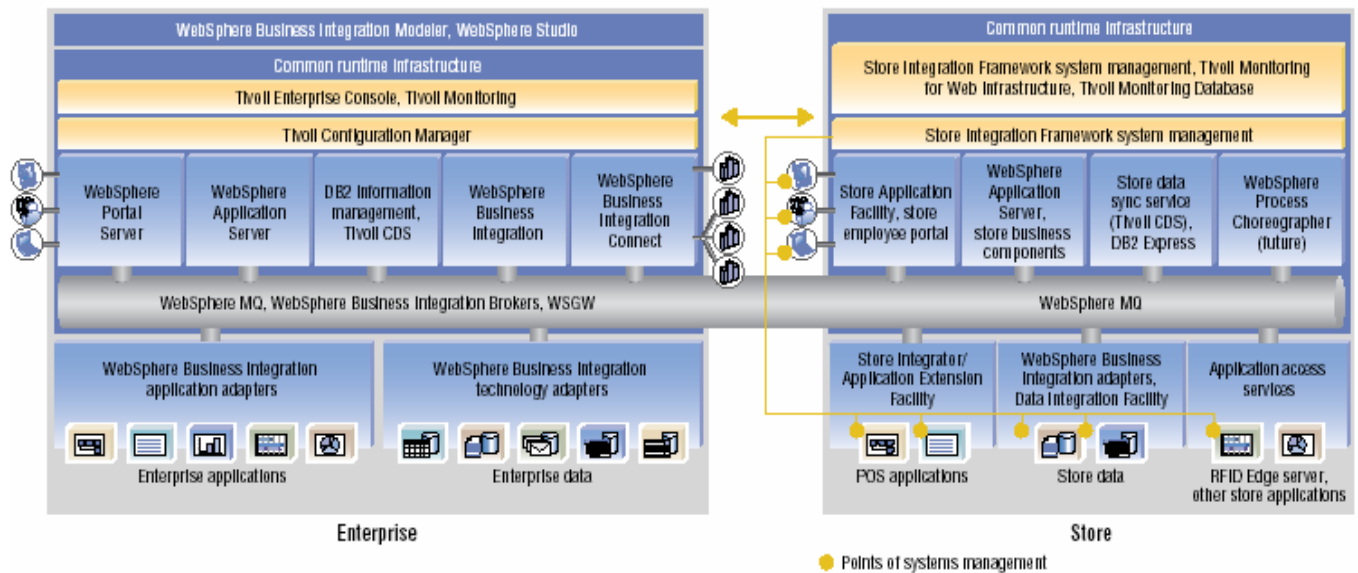
Enterprise scope scenarios

- Movement of transaction data to WBI (Websphere Business Integration) and work flow processing
 - Data mining of transaction data using WSC (Websphere Commerce) infrastructure
- System management of the store
 - Deployment of software and data using TCM (Tivoli Configuration Manager) and DMS (Distribution Management System)
 - Start / Stop of application
 - Monitoring of the store
- Integration of web enabled transactions with in-store POS transaction sales
- Personalized marketing application while the consumer is shopping (partnership) driven by an enterprise offer management system.
- Management of returns for financial accounting and fraud prevention using enterprise data mart
- Inventory extension by ordering merchandise for the consumer within the store
- Management of Gift Registry including integration with the POS transaction
- Delivery of employee portal for personal data, training, and corporate communications

Store scope scenarios

- Integration with POS - Consumer service device
 - Portable shopper, Line buster (pay at POS)
 - Self Checkout
- Promotion integration (get price for this customer)
- Integration to POS for Transaction processing (payment)
 - Simple - Customer kiosk
 - Complex - Fuel, Pharmacy, RFID solutions
- Websphere Application Server in the store for non-POS applications
 - Video rental, Pharmacy
 - Operator productivity console
- Independent Hardware Vendor devices (System Management, client support)

Figure 4: Technology solutions that deliver IBM Store Integration Framework architecture
Current IBM software offerings



1.3.1 IBM Middleware

The objective for Retail on Demand EBO is to bring value to the retail environment by delivering solutions - including hardware, software and related services – that help to transform the store with on demand capabilities. To deliver this operating environment in the store, IBM offers a software infrastructure that, at its very core, includes the following components:

- Websphere MQ for standard, assured-message delivery for all applications within the store.
- WAS for Retail Outlets for the server J2EE programming environment in the store, including the Java Messaging Service and Network deployment capabilities. Function includes facilities to simplify the installation process for retailers to roll this function out to thousands of stores easily and with the minimum of hands-on experience required.
- Tivoli for Retail for managing the network of stores from the enterprise. Software distribution for maintenance and store monitoring are the most basic requirements for this environment.
- Data Management capabilities require the continuous flow of many critical data elements between the store and the enterprise. A variety of solutions may be required to meet the retailers needs for caching small amounts of data in the store, synchronizing the use of enterprise data between multiple store functions, and the process of real time data flowing back to the enterprise. This capability includes the WBI (Websphere Business Integrator) functions for formatting and processing daily transaction data collected from the stores.
- Data mining at the enterprise will integrate store and web sales data for analysis by common tools and use by retailers for advancing their business. This facilitates a common view of the consumer for the retailer, regardless of the technology path servicing the consumer.
- Portals for human resource interfaces for employees such as access to benefits information. This also provides for easily accessible training and availability of sales assistance tools for employees.
- Store management portals to deliver key data and process management assistance for the role of store decision making.

1.3.2 Store Integrator

The objective of the Store Integrator offering is to deliver on the promise expanding to web based, WSAD (Websphere Application Development) programming in the store while preserving existing in-store software systems. It provides for the integration of new store operations with customized point of sale processing. The software solution requires a framework to extend store operations.

The Store Integrator delivers the architecture and implementation for the following:

- Consumer Service Device software components, device data exchange functions, and common support for IHV provided devices
- Consumer Service Device reference application logic which provides a seamless solution for consumer assistance in the store.
- Components for extending legacy POS applications using Java
- Components that enable integrating the POS operation (legacy POS) and retail web applications.

- Capability to support multiple instances of terminal POS business logic running on a 4690 controller
- Interchange of POS data between store and enterprise in a timely / real time manner using XML standards
- Systems and Network management to facilitate the control and monitoring of the complete “store” environment and to do this over its life-cycle, and in a standards based, vendor neutral, non-intrusive way.

2 Store Integration Framework Reference Architecture

2.1 Enterprise

The enterprise component model shown in **Error! Reference source not found.** is a very high-level overview of the major elements of the retail enterprise. Highlights of this chart include:

- **User Interaction Services:** The user interaction services establish the relationship between the user and the store applications, both local and enterprise, through intuitive interfaces. They create interactive venues such as employee sales portals or consumer self-checkout kiosks. The services allow the tailoring of access to meet end-user needs, including in-store shoppers, online consumers, sales associates and store managers. This particular service is the key for integrating people, delivery, experience, and resource services to provide the following:
 - Aggregation and Collaboration: The bringing together of people, functions, and information into a single integrated experience for the end user.
 - Personalization: Tailoring the content of the user experience to their role, their context, and their personal preferences
 - Access: Access is about letting the user work with the portal any time, any where, from any device and without changing the user experience.

We have a similar user interaction service inside the store and that also provides similar functions.

- **Application Services:** The application services deliver a foundation on which to build new applications and to integrate new solutions and technologies into the store. It includes the attributes and components of the Websphere Application Server. It provides the following capabilities:
 - Support of J2EE, XML, Messaging and Web Services programming model
 - Autonomic administration and management
 - High transaction throughputs
 - High availability and failover support
 - Advanced scalability support
 - Advanced Service Oriented Architecture capabilities

The Websphere Application Server is also deployed in the store and is the infrastructure for building integrated business processes which optimize the retail operations.

- **Process Services:** The key capabilities of process services are process choreography, state management and staff services:
 - Process choreography is the execution of process flow logic that orchestrates activity that is either in the form of application-to-application or application-to-people interactions. Such logic is external from an application and, typically, initiated by one of two events - human activity (e.g., the entry of an order via a portal interaction) or an application's activity (e.g., detecting a low inventory condition). In either case, it is the placing of a message on the transport that triggers the process. From the standpoint of the initiating application, it appears as if its adapter has detected that an event has occurred in that application and has pulled data from it. It does not realize that a process has been invoked. Currently, process flow is expressed as collaboration (WebSphere InterChange Server), or as Flow Definition Language (WebSphere MQ Workflow), or as both. Strategically, these will converge to the single industry based standard, Business Process Expression Language (BPEL). In many cases, process integration needs to include application-to-people activity. This is accomplished through State Management and Staff Services.
 - State Management makes it possible for a process to change its behavior depending on where it is in the execution of a flow. For example, in an order processing system the action of 'cancel order' service would be different depending on whether the order has only been "entered" or has been "entered and paid for" (say, via a charge card). States are user defined and are maintained as part of the choreography.
 - Staff Services allows human interaction within a process. For example, in a purchasing system a manager's approval might be needed when the purchase amount exceeds \$10,000. This facility is provided through the creation and management of work lists. A list can 'feed' multiple people (e.g., on a list of prescriptions that are being filled by a group of pharmacists) or it can be directed to a single individual (e.g., a list of purchases that need the approval of a department head). Here, portal technology can be used to provide the needed people-to-application interaction.
- **Information Integration Service:** Synchronize data, in all formats and contexts, virtually everywhere across the enterprise. Provide a consistent, unified view of business information. Within databases, across applications, on the Web and at the POS. The information integration services in conjunction with data access service (WBI Technology Adapters) provide a data federation capability for the enterprise. The data federation capability provides a single point of access and control of data.
 - Transparency in access; all data in the enterprise appears to be a single source of data.
 - Heterogeneity
 - Integrates data from diverse sources

- Structured, XML, unstructured, messages, bulk, web, etc.
- Function and extensibility: SQL/XQuery plus backend specific functions
- Autonomy: Non-disruptive to data sources.
- Performance: Distributed queries.

Data is exchanged between the stores and the enterprise

- **Enterprise Service Bus:** An architecture construct for end-to-end integration of distributed software applications and components. It is a core component of service oriented architecture. It has the following characteristics:
 - Mediation service responsible for routing and transformation (for impedance matching between services).
 - Event services supporting publish and subscribe capabilities.
 - Transport service providing Synchronous/Asynchronous, Persistent/Non-persistent, and Loosely-coupled/Tightly-coupled messaging
 - Standards based
 - HTTP/HTTPS with option for WS-Reliable Messaging
 - JMS, JAX-RPC, SOAP
 - WS-Security, WS-Policy, WS-Addressing
 - Etc.

A similar bus notion exists at the store level which provides the infrastructure for integration at the store.

- **Application Access Services:** Connectivity is critical in establishing an integrated, enterprise-wide system from which companies can extract information from disparate sources, such as different packaged and legacy applications, technologies, and mainframe systems. The WebSphere Business Integration Adapters help address this need for connectivity with one consistent approach. All adapters run on the same common framework, ensuring that they can be reused with different integration engines. The common framework also allows greater speed and ease of adapter deployment, requiring less skill to install a broad range of adapters.
- **Community Integration Services:** The community integration service provides the business to business (B2B) exchange capabilities. It supports various forms of transports, data formats, protocols, transformations, etc and it can be thought of as an application to application connectivity between trading partners. It could be an existing connection using EDI, or newer protocols such EDI INT (AS1/AS2) or industry defined protocol such RosettaNet. The WBI-Connect product part of the Websphere portfolio provides this functionality. WBI-Connect provide document exchange services with a rich set of operational tools to manage the trading community. It provides a highly focused, process level integration so it can derive the most possible value from its service-level-agreements (SLAs) – rather than simply conduct business with a trading partner, can have a far more granular trading relationship that identifies the profitable parts of the B2B exchange and maximises them for mutual benefit.

- **Monitoring Services:** This supports both process-level and IT-level monitoring in an integrated fashion. Most IT departments deal with thousands of issues every day. All of them have their own practices for resolving problems ranging from first-in, first-out, to prioritized by how badly the resources are broken (red, yellow, black). While this certainly helps to resolve problems, the IT team frequently does not understand the relationships between these processes and the business and is unable to prioritize what they do to have the greatest impact to the bottom line. Hence integration of IT-level monitoring and business-level monitoring is a key capability and is provided by the IBM solution.
 - IT monitoring allows visualization of IT environment in business terms and management of service levels to business objectives
 - Process monitoring allows business leaders to understand in real-time the state of key business processes
- **System Management Services:** System management service provides the basic capability of install, configure and operation administration capabilities. This is a key aspect of the store solution and is captured in the chapter dealing with Solution Lifecycle Management.
- **Model Services:** This service is the build-time capability which allows a user to model business processes, simulate and analyze them, before actually putting them into production. This way, by the time a process is put into production it is nearly optimized. The service provides these capabilities:
 - Business requirements are clearly defined and documented
 - Design the way processes will work, including 1) Model current processes (as-is) and 2) Plan future processes (to-be)
 - Simulate and Analyze processes
 - Identify ROI
 - No programming required

The WBI Modeller facilitates anyone in the organization, whether it is a business analyst or I/T specialist, to capture a business process and clearly define and document the process. Both the current (as-is) and the future (to-be) can be modelled and compared using the simulation and analysis capabilities of WBI Modeller. Comparing these processes will identify ROI opportunities in the business by changing the process as modelled. This is an enterprise only capability.

Store Interaction Reference Architecture

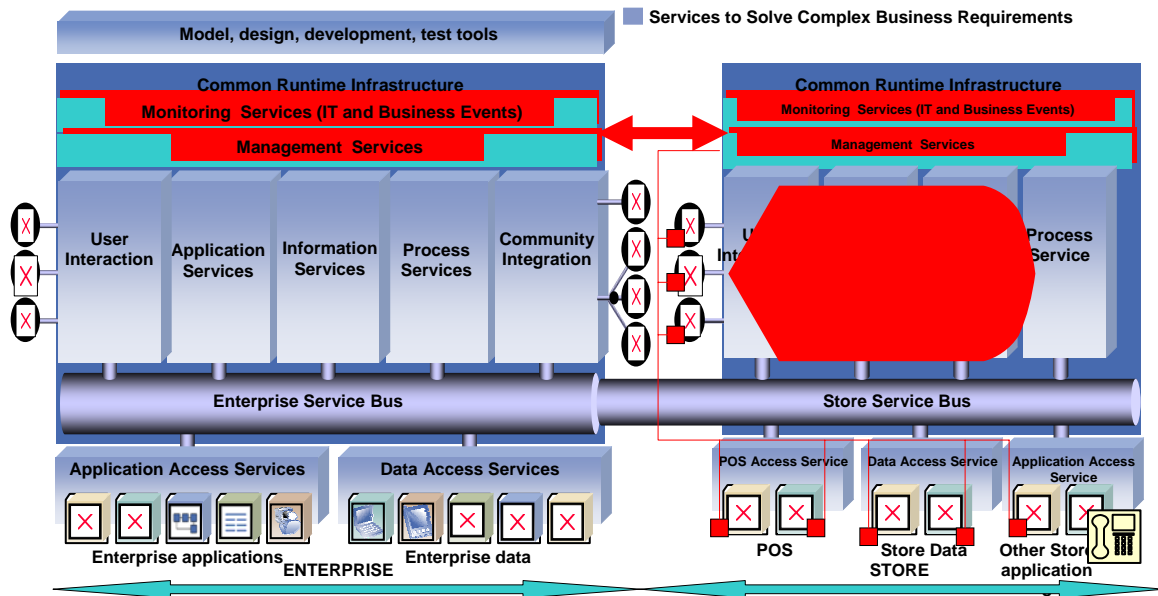


Figure 2: Enterprise and Store Component Architecture

2.2 Store

The Reference architecture includes a similar model for the store represented in Figure 1 where the store is served by a web application server.

- **Store User Interaction Services** : This service is the key for integrating people, delivery, experience, resource services to provide the following:
 - Aggregation and Collaboration: The bringing together of people, functions, and information into a single integrated experience for the end user.
 - Personalization: Tailoring the content of the user experience to their role, their context, and their personal preferences
 - Access: Access is about letting the user work with the solution any time, any where, from any device and without changing the user experience.
 - Retail unique peripherals: Point of sale data exchange is provided in a standardized manner to simplify integration with browser based applications. In the store, this access can include a determination of the consumer device location.

- **Store Application Services**: The application services deliver a foundation on which to build new applications and to integrate new solutions and technologies into the store. It includes the attributes and components of the Websphere Application Server as well as specific retail business function enablement, provided by IBM or IBM business partners. When deployed in the store, the WAS platform will be tuned to provide scaling appropriate to the store's use, the configuration for the reliability of the

network, and system management for widely distributed WAS platforms. It provides the following capabilities:

- Support of J2EE, XML, Messaging and Web Services programming model
- Autonomic administration and management
- High availability and failover options
- Advanced Service Oriented Architecture capabilities including interfaces into virtualized checkout business logic.

The Websphere Application Server deployed in the store provides the infrastructure for building integrated business processes which optimize the retail operations. It also provides the web-based technologies to enhance the shopping experience of the consumer via a browser user interface.

- **Store Information Integration Service:** The information integration services provide a data federation capability for the store and enterprise. The data federation capability provides a single point of access and control of data. Data is exchanged between the stores and the enterprise.
- **POS Access Service:** The Business logic that manages the customer checkout process in the store is often highly customized and tuned to the retailers' business needs. It is critical to the store's operations. At the same time, retailers need to integrate new function into the store in such a way that it is totally congruent to the sales operations. The Store Integrator will provide Java services that will allow external components to reuse the sales logic seamlessly. The Application Extension Facility (AEF) API's expose this capability. This allows new programming to be independent of the knowledge of the intricacies of the sales transaction while taking advantage of its configuration, data use, and interfaces to the enterprise.
- **Store Data Access Service:** Data in the legacy store system may not be easily accessible with standard access capabilities of Information Services. RSS Data Integration Facility (DIF) is a program product that currently provides "TLOG adapter" function, as well as access to the 4690 OS file system. As retailers expand their operations with additional sales opportunities (ie. Fuel, instore kiosks, Starbucks) or new customer interaction models (ie. Home shopping, portable shoppers, self checkout), they want to manage these opportunities with the same processes, procedures and data that they use for their main checkout system. The AEF addresses the integration of business logic and its immediate data. To complete the integration infrastructure, the DIF expands the capability to share all data that is managed by the POS system.
- **Application Access Service:** Application Access services are value-add components which can be used to extend the function of the in-store operations. These components are composite functions of the underlying AEF capabilities, and will enable simple integration with POS. Unique store requirements such as access to RFID content may be supplied by this component.

From a technical operations point of view, the network between the retail store and the enterprise is the key enabler of sales integration. Falling network costs and improved

connectivity services have very quickly enabled a new generation of applications to be considered. However, there must be careful design and build of store-level applications which will continue to operate within the limitations of many retailers' external networks. Before proposing full integration between POS and enterprise-level applications, acceptable response time characteristics must be studied and appropriate engineering done to insure that response time.

2.3 Store Integrator

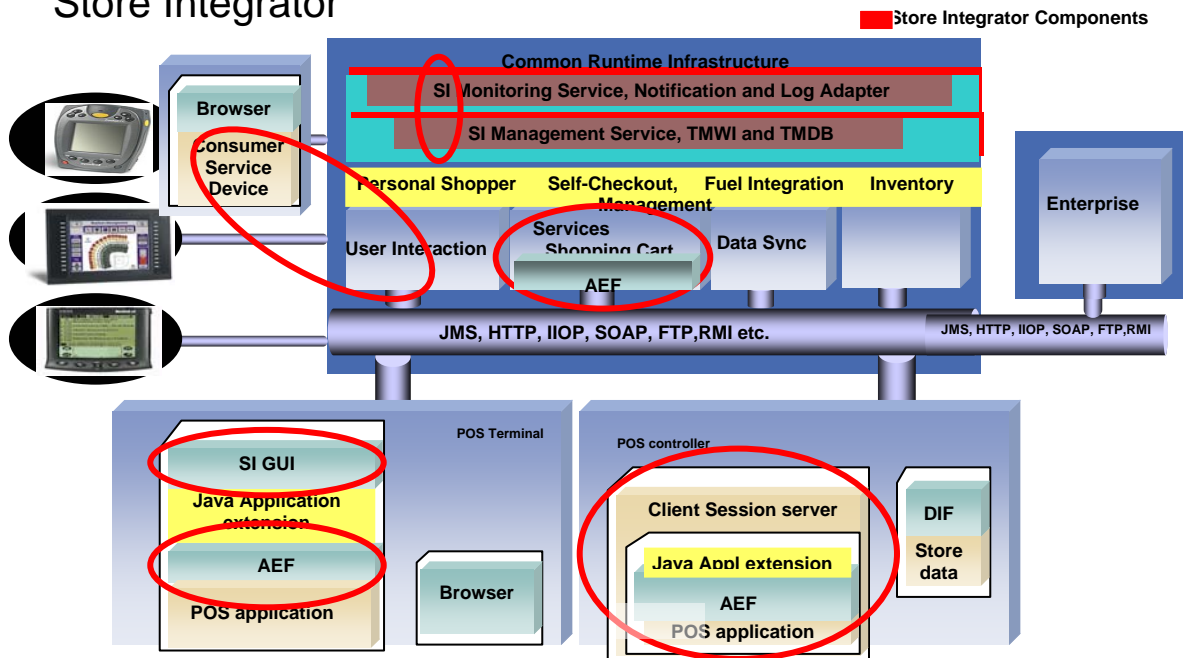
As a key component of the IBM Store Integration Framework, IBM Store Integrator allows retailers to integrate store and Web-based applications with their 4690 ACE, SA, or GSA Point of Sale systems in response to today's evolving retail environment. The consumer shopping experience has changed dramatically. Integrated multichannel retailing is becoming the norm. While the Web and new technology have empowered consumers and opened new venues for evaluating and purchasing merchandise, the store remains the hub of the customer experience.

Store Integrator provides the following components:

- Provides a common Java API to integrate store and Web-based applications with POS. Business Partners and Independent Software Vendors can use this API to interface their applications once with ACE, SA, and GSA.
- Eliminates the need to maintain duplicate price files or payment facilities. Other applications can now just reuse the pricing and transaction logic in POS.
- Enables the extension of existing ACE, SA, or GSA POS applications using modern Java development tools.
- Provides an enhanced XML-based GUI toolkit to develop simple, intuitive consumer and operator POS interfaces with upgrades available for SureVision and Astra PRPQ customers.
- Enables integrated management of the store based on the Java Management Extensions (JMX) standard.

The components of the Store Integrator architecture and their relationships are illustrated by the following diagram.

Store Integrator



2.3.1 Device Software Components

A Consumer Service Device is primarily targeted for use with server based applications based on the framework. The role of the device in these applications is to provide a relatively small and inexpensive device that can be used to display server generated content viewed with a web browser on the device. All of the application business logic and access to POS, e-commerce and enterprise application data is addressed by the back-end applications and the framework. The primary objective of the device software components is to provide a relatively full function browser, support acquisition of data from the locally attached retail POS devices such as a scanner and magnetic stripe reader, and support required deployment, operation, management and security functions to allow the easy setup, use, maintenance and protection of the device and its data in the retail environment.

Given the objective of low cost, the Consumer Service Device hardware may be limited with respect to its performance and capacity. Given these limitations, it is important that the software components occupy as little space as possible and operate as efficiently as possible with the given resources. At the same time it is also important that the device software components are as portable as possible to allow them to be supported across a broad set of devices. In addition, components such as the Java Virtual Machine, browser etc. should be chosen to support this embedded environment and allow selective configuration to ensure that only the system components required for the intended application need to be included.

2.3.1.1 Browser

There is a requirement for a browser in each consumer touch point, including the POS terminal node and the mobile kiosk. The browser provides the basic UI for remote business logic and web provided content. The minimal requirements for the browser are defined in the standards section of this document.

2.3.1.2 Interface to Consumer Service Device Peripherals

The solution requires a way for the business logic on the web application server to be able to receive the data read from the peripheral I/O devices attached to the consumer portable device. The devices supported for Store Integrator release 1 are the Magnetic Stripe Reader (MSR) and the bar code scanner. These are non-standard devices to a normal browser HTTP interface; however, the solution will provide a pass through capability with no browser modifications required. The Store Integrator web application device access will be based on a Custom Java Server Page (JSP) tag per device. This tag will use JavaScript communicating with a non-visual applet which uses JavaPOS device controls to read the devices. JavaPOS interfaces are required to be available on the device for the devices supported.

In order to implement this support on a device, the device must provide/include the following standards-based functions:

- A Java Virtual Machine
- JavaPOS device interfaces for the POS I/O
- A browser with Java applet support and Javascript to Java applet communication capability
- The local device components of the Web Application Device Access (WADA) (available from IBM)

A device enabled with these capabilities is then able to interact with the applet components of the Web Application Device Access (provided as part of SI and used with compliant applications) to provide access to the local device I/O from the web application. This allows a common interface from the device to web applications. The advantage of this solution is that web applications can be developed so that they support any device that supports this architecture. The applications do not have to change based on the particular I/O device brand attached or the platform running on the device.

2.3.2 POS Terminal

2.3.2.1 Legacy POS

The legacy POS component represents the customized checkout engine in the store. It may be running in a register with or without a local disk. The component manages the business logic for the point of sale transaction including computing price per line item, taxes, receipt tape printing, total cost of transaction, receiving and processing payment, and constructing the raw transaction data records. 4690 Applications which are supported with release 1 are Supermarket and ACE.

2.3.2.2 Store Integrator GUI for 4690 OS

The SI Graphical User Interface (GUI) component of the SI framework provides a high function touch user interface that is easy to use and customize. This component requires the standard 4690 OS terminal platform and uses the local Java graphics support. It is not a component available for the consumer service device.

The SI GUI meets the needs of retailers who want to migrate from the existing simple 4690 POS line display operator interface. The value is migration to a full touch environment gradually without making major changes to the underlying business logic or having to maintain two applications. This enhanced GUI tool enables retailers to develop simple, intuitive consumer and operator interfaces for ACE, SA, and GSA using XML. No programming is required.

Those retailers who have invested in the SureVision GUI PRPQ to create GUIs can protect those interfaces as they move to this new enhanced GUI. The 4690 Java GUI/ASTRA PRPQ version 4.0.1 has been enhanced to take advantage of the SI Application Extension Facility. When operating in this Store Integrator-enabled environment all existing ASTRA functionality will be preserved and ASTRA's user extension mechanism will use the new AEF APIs.

Details are available in the IBM [Store Integrator GUI Programming Guide](#).

2.3.3 Store Integrator POS Application Extension Facility

This Store Integrator feature allows existing functionality within IBM General Sales Application, IBM Supermarket Application, and IBM SurePOS Application Client Environment for 4690 to be accessible through a set of APIs for use remotely by other applications such as self-service fuel, gift registry, special orders, and Web shopping.

This component provides the interface for a Java application extension to integrate new business logic with the standard legacy business logic managing item sales, etc. The term 'extension' in this name refers to new business logic which can be integrated as an extension of the legacy POS. This is not to imply that the AEF itself is necessarily extendable by third parties.

The interface will provide multiple capabilities:

- **Data Provider** - This component provides the interfaces that allow the application to programmatically register to be notified of specific events which are occurring in the sales transaction. Appropriate data associated with the event will be supplied. Examples of events are Operator sign on, Loyalty card entered, and Item scanned.
- **POS Automation** - This component provides the capacity to extend the operation of the checkout engine by automating the supply of input.
- **Device Access** - This component provides the APIs that are used by an extension to gain access to input device data streams such as data from the keyboard and scanner. This API will support the ability for an extension to consume a device event or to pass it on to the point of sale application.

Future capabilities are planned for “Service provider.” This component provides the interfaces for a service to register to replace / supply function needed by the checkout engine. Examples of services exposed are 'Operator Authentication' and 'Tax Calculation'. The implementation to this interface is likely to be invasive to the legacy POS application. (Not Release 1 function).

Details are available in the [SI Programmers Guide](#).

2.3.4 In-store Controller

2.3.4.1 IBM Data Integration Facility

The Data Integration Facility consists of a flexible messaging component, a transaction data distribution component, and standard messages for the selected solutions that we implement. The messaging component consists of a set of Java classes which can be customized for different parameters, message types, message requests, and message replies. The messaging component encapsulates the details of the underlying messaging infrastructure and provides a flexible architecture that allows for future growth and change. The communications component is designed to allow multiple infrastructures to be used; however the initial plan is to use MQSeries Everyplace for messaging. The transaction data distribution component is built on top of the messaging component and provides support for sending transaction records to a central host from legacy POS applications using industry standard data formats.

As we implement solutions on top of the Store Integration Framework, we will also deliver the standard message formats used by these solutions. This will allow other POS vendors and applications to use and integrate with these solutions.

Details are available in the [IBM Data Integration Facility Programming Guide](#).

2.3.4.2 4690 Client Session Server (CSS)

This Store Integrator component supports the browser based programming model for a wireless, portable device, allowing the device to provide the user interface for POS checkout business logic. The checkout business logic executes on the 4690 store controller, remote from the UI. The UI is provided by a browser component managed by the WAS business logic (described below).

The CSS provides an environment to run multiple instances of the terminal Point of Sale (POS) Application on a Store controller. Further there is not a need to have real POS I/O on the controller. The POS Terminal Applications running in the CSS do not need to be modified and should not know they are running in the CSS.

The Client Session Server will be used in combination with the SI Application Extension Facility (AEF). The CSS will interface with the AEF and is available for 4690 OS. The AEF will have the ability to start CSS components on more than one controller in the store. Multiple controllers will be required to support 100 terminal sessions for 100 consumer devices.

The POS Terminal Applications are supported by the following services:

- Drivers which support the same interface as our existing POS I/O drivers (while redirecting the I/O)

- File Services accessed through “R:” which is the (Remote I/OAccess Method) resource manager that allows terminals to access files on the controller disks. For example, R:C:\adx_ipgm\file.dat or R:D:\somefile.dat.
- Handle the Application Loader interface used to load IOP tables
- X and Y Ram disk support which are terminal RAM disks, some systems also provide Q: enhanced RAM disk - bigger, long file names (through VFS)
- PC: hardfile support

Details are available in the [Store Integrator Programming Guide](#).

2.3.5 Store Web Applications

The architecture provides for store web applications integrated with Point of Sale. It is dependent on a web application server, and includes specific Store Application Framework components that are core to enabling the solution.

2.3.5.1 Web Application Server for Retail Outlets

This architecture will require the Websphere Application Server. This platform will provide the programming environment for the business applications accessed through in store consumer service points.

IBM provides a package of WAS for Retail that supports J2EE programming with EJB 2.1 standards. Modularity and software reuse are some advantages of the use of EJB objects to develop and extend the business logic of our applications. Third party EJB objects are supported as long as they use EJB 2.1 or lower, have been tested with WebSphere 5.0 and are self-contained (that is, contain all the prerequisite classes or libraries in the JAR/EAR files).

Other functions supported include web container, Performance Monitor Interface Service, DynaCache, Trace Service, Logging Service and JMX administrative services.

2.3.5.2 Store Application Framework

The Store Application Framework provides for the integration of store applications using the J2EE programming environment. The SAF deliverables include business objects and tools and education for their use that facilitate a POS integrated solution. Components are also provided for a system management dashboard to consolidate store management data.

SAF will provide tools to assist in the use of a browser interface to peripherals available to in-store users. To use these tools, the consumer device must support the appropriate exchanges for the provided JSPs and server side business. These applications expect I/O data such as scanner input to be provided into a web application via a customized JSP tag. Any device client that matches these criteria can be integrated with the SAF framework components.

Access to POS business logic

The SAF provides the component to expose the Application Extension Facility Java APIs locally for web applications. This support requires a 4690 Controller with Store Integrator installed to reside on the same network.

Store Integration Framework Applications / Common Business Components (CBC)

The Store Integration Framework defines a new IT infrastructure on a service oriented architecture built with loosely coupled business components which address the following requirements:

- Heterogeneous platforms and programming environments
- Evolutionary, not revolutionary
- Support stateless and stateful business components
- Support for multiple component topologies
- Support an event model

This architecture provides for these applications to have access to appropriate reusable business components, such as, a Shopping Cart business component that can be used programmatically to integrate new function with in-store point of sale applications. A shopping cart object would provide a simplified AEF interface with a ‘cart’ paradigm which maps to the physical world of a ‘shopping cart’ with functions to add or remove items from the cart, to get a list of items in the cart, and to see the current subtotal for the price of items in the collection. A shopping cart object is a key building block for ISV provided shopping assistance using the in-store business logic for POS.

Common Business Components provide application services which can be composed into business processes to enable integration and deliver business functions. Business applications are a collection of such components and processes that are customized (and composed) to deliver business functions. Packaged ISV applications can:

1. Incorporate such Business Components within their application services, and
2. Expose their functions as Business Components which may be a collection of services.

The J2EE programming model enables such components to be developed and deployed as coarse-grained, loosely-coupled, and instrumented/managed software components.

IBM provides the base products (e.g., WAS, DB2, Tivoli, MQ), retail-specific utility Services (e.g., AEF APIs), and Common Business Components as reference implementations as well as base functions to accelerate time-to-value for partners, services teams, and customers.

IBM’s intent is to publish interfaces and to promote standardization at the API (services) level as well as at the Business Component construct with exposed services. IBM’s investment in building such CBCs are based on:

1. High-value, but non-core-competency for the ISVs,
2. Packaging as a higher-level “API” set, e.g., shopping cart on top of POS access APIs, and
3. Mechanisms to acquire and populate application data.

3 Solution Lifecycle Management

3.1 Introduction

A solution consists of one or more business processes, any of which may span one or more applications, middleware and system environments. When we talk about managing a solution, the idea is not only to manage the software components that have been developed to provide the solution but also to manage the middleware and the underlying system environment too. In the context of store integration framework, the definition of *solution lifecycle management* deals with providing answers to the following questions:

1. How can we install, deploy, monitor and manage all components of a solution, including software components, middleware and system environment, over its lifecycle?
2. How can we do the lifecycle management remotely over low-bandwidth connectivity to the store?

As a store solution is constructed, tested, deployed in production and finally disposed of, a number of activities take place. The figure below illustrates the progress of the solution over the lifecycle.

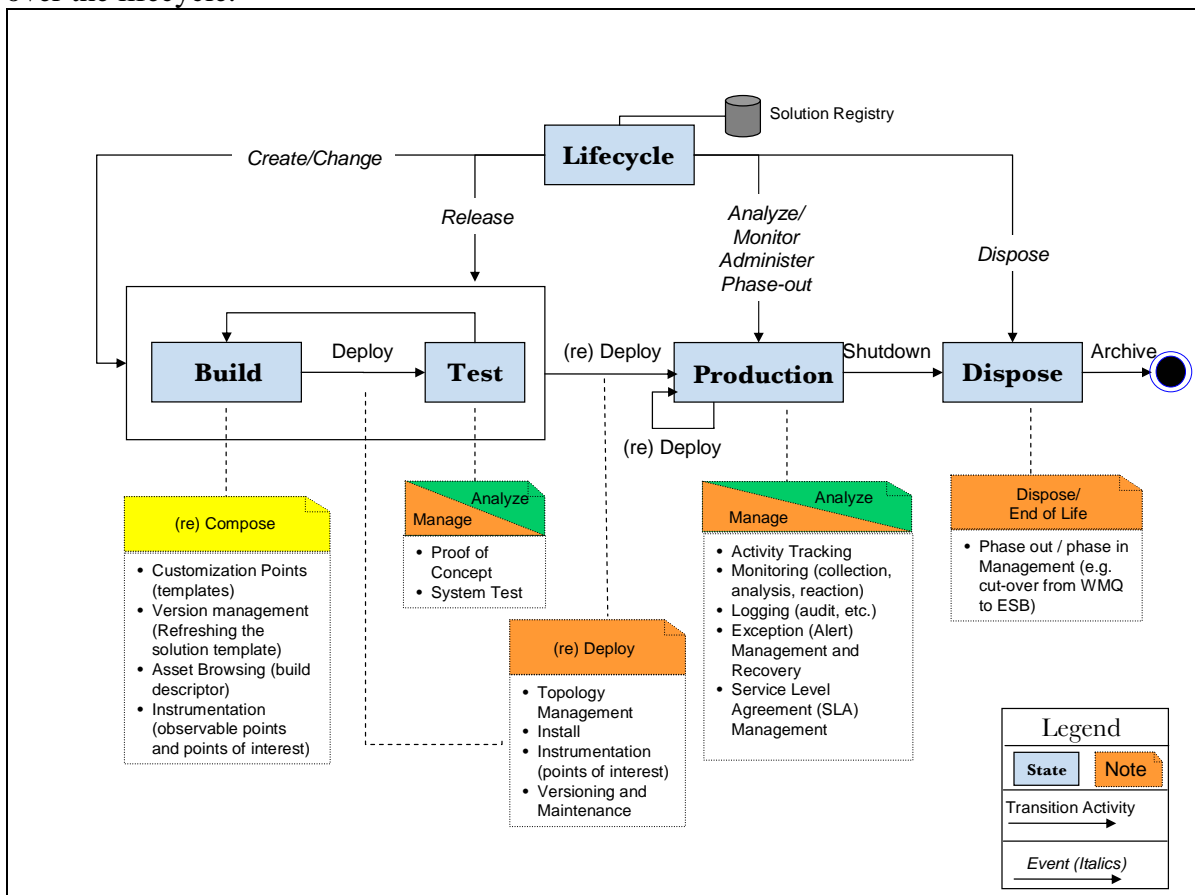


Figure 1 : Solution Lifecycle – Major Phases and Capabilities

To provide the lifecycle management of the store solution, we need to work within the context of various products from Software Group and Retail Store Systems. The Store

Integration Framework provides certain integrated store solutions. It also is a solution enabler, providing services that can be used by partners to provide a complete store solution. As a result, the solution lifecycle architecture should provide functionality for managing the lifecycle of a) the solutions provided directly by the Store Integration Framework and b) solutions that are enabled by the framework, but not directly produced by it.

This document captures the overall architecture of the lifecycle management of the store integration framework solution. The build and the test phase of the solution lifecycle are not covered in this document. The details of the in-store solution are captured in another document, and we concentrate primarily on the enterprise view of the solution.

3.2 Solution Architecture

Figure 2 shows the enterprise view of the system management architecture.

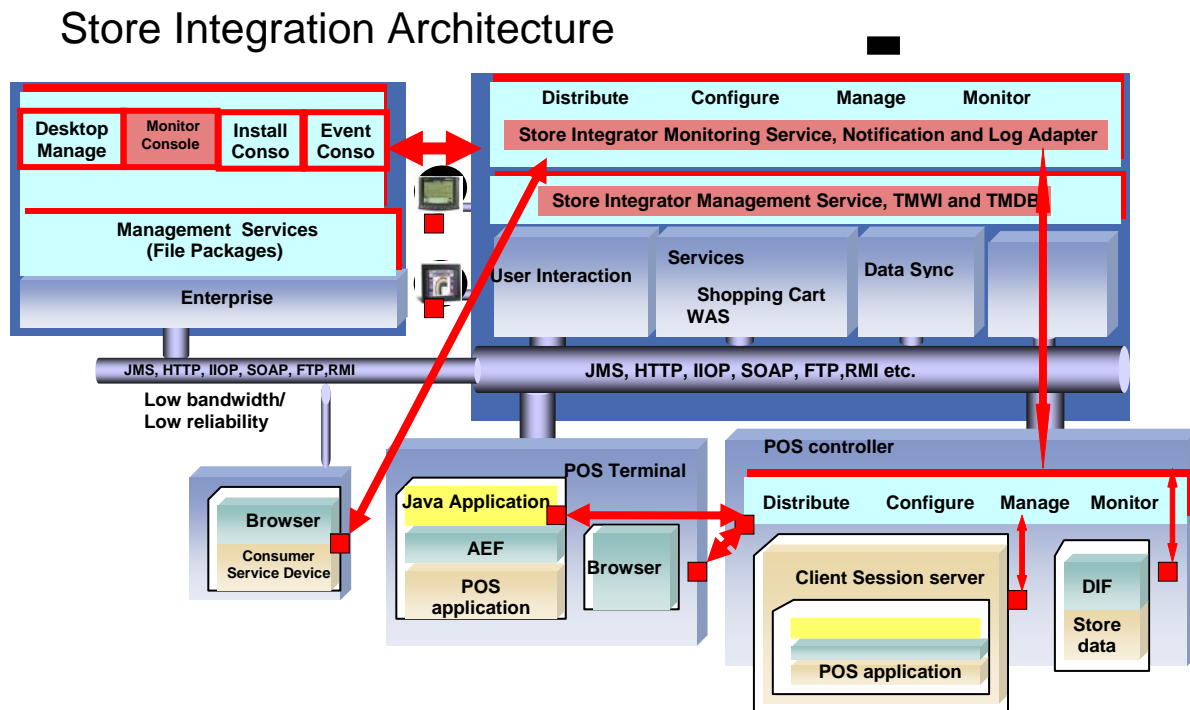


Figure 2: Lifecycle Management Architecture

The lifecycle management problem objectives can be broadly divided into 3 parts. All three objectives need to be fulfilled remotely over a low bandwidth and an un-reliable network.

- **Install, Configure and Update:** The objective is to distribute and install all software related to the store, from the enterprise. There are a number of different software components that get installed, configured and maintained over the

lifecycle of the solution in the store. Our install solution has to deal with these different components, their different behaviors during the installation and configuration process, the different system environments in which the components execute, etc. Further, from the enterprise point of view (and also the store) we need to have a single tool, even though we may be using different tools and code underneath the covers for each of the different components.

- **Operation Administration:** The IT skills available at the store-level are non-existent. Hence, the objective is to manage and administer all the components in the store from the enterprise. Once again we have to deal with multiple components both software and hardware and their different management interfaces. We need to wrap the individual management interfaces of the components and expose it to the enterprise tool.
- **Monitoring and Troubleshooting:** The objective is to provide capabilities that will allow the monitoring the health and the status of both software and hardware components at the store from the enterprise. Capabilities are provided that allow an enterprise operator to pin-point any problem that may occur with the hardware or software component at the store. Also, infrastructure is capable of sending out alerts and notifications in the case of malfunctions of the any of the components.

3.2.1 Solution Architecture Strategy

System management is key to the success of a store operational system, and is a differentiating factor in solutions from IBM. To implement capabilities such as software distribution in the store, Store Integrator includes JMX Managed Beans (Mbeans) for each particular device and component. Implementing support in this way allows the device to participate in features such as store level monitoring. The Store Integration Framework will provide a set of “standard” MBean Interfaces, and where applicable - implementations of those Interfaces. IHVs/ISVs are expected to implement those interfaces where applicable, and define and implement their own interfaces as required to surface their unique functionality. These implementations, whether of existing interfaces, or IHV/ISV unique, will plug into the systems management infrastructure in exactly the same way as those MBeans provided by Store Integrator itself. Details are available in the [IBM Store Integrator Programming Guide](#).

3.2.1.1 Consumer Service Device

System Management for the Consumer Service Device handles maintenance and operational issues for both the base device and locally attached peripherals. Store Integration Framework based systems management provides the JMX sub-agents which allow the compliant device to participate in the overall management architecture. The JMX sub-agent is a Java application which provides for system discovery, proxy support and logging of a given device in the framework management infrastructure. The system will be automatically discovered by the store management agent, and the output of standard logging interfaces such as log4J and syslog on the device will automatically be directed to the central logging server of the store. In addition, there will be a CIM to JMX mapper that will map CIM inventory data up through the JMX store management infrastructure.

Most Consumer Service Devices will ship with a preloaded software stack so that downloading the execution environment will not be required for every boot sequence. However, for maintenance of the software, the devices will recognize the need for a reload and then automate the download of a new image.

3.2.1.2 Store

System Management for the store considers the maintenance and operational issues for the Websphere platform in the store as well as the integrating with the existing legacy point of sale terminals and controllers. Tivoli is capable of providing system management for the distributed network of WebSphere platforms. Tivoli currently supports an agent for the 4690 OS.

Support required from Tivoli includes software maintenance distribution, configuration management, and problem reporting / monitoring capabilities. The IBM Store Integrator also includes a store dashboard to allow for local system management reporting / monitoring.

3.2.1.3 Enterprise

System Management for the Store Integration Framework solution considers the maintenance and operational issues for connection and data sharing between the store and the enterprise. This solution will integrate with the function provided by Tivoli for this purpose.

3.3 In-Store System Management Strategy

The Store Integration Framework provides a solution to facilitate the control and monitoring of the complete store environment. A mid-level manager model is employed with all systems and devices in the store interfacing to a single collection point in the store. This collection point is the focus of all outside management actions, and is in-turn responsible for executing any and all management actions across the components that make up the store. Each system or device in the store contains a management sub-agent that interfaces with the management agent in the store via the standard management agent API. A local or remote management application then interfaces with the management agent representing the store. This application can be a simple browser application or a gateway into a more extensive systems management solution such as those provided by Tivoli.

To participate in this management structure, a device must include the standard management sub-agent and implement a certain set of standard management functions, such as software distribution, inventory etc. These agents are implemented according to the JMX (Java Management Extensions) specification. The JMX specification describes a general management framework for the Java environment. It is a multi-tier architecture that includes an instrumentation layer, MBeans; an agent layer, MBeanServer; and a management layer. MBeans (Managed Beans) which meet certain standard definitions must be implemented on each device for it to participate fully in the management framework. In order to implement this support on a device, the device must provide/include the following standards-based functions:

- A Java Virtual Machine

- Standard Store Integrator Framework device sub-agent and Mbean definitions
- MBeans that implement the appropriate systems management interfaces

A device enabled with these capabilities is then able to interact with the management agent infrastructure (provided as part of Store Integrator) to provide a common method for systems management throughout all devices within the store.

This architecture defines four roles for Systems Management. All Systems management code will fall into one of these roles, and will adhere to the defined responsibilities for that role.

3.3.1 Management Sub-Agent

- Located in every component that makes up the store environment.
- Provided by the individual component developers.
- Conforms to the implementation defined by the “Management Agent API”.
- Defines manageable attributes and exposes them for interaction to the environment.
- Defines functional entry points as needed and exposes them for use within the environment.
- Defines and issues Notifications.

3.3.2 Management Agent API

- Provides the implementation for communications between the “Management Sub-Agents” and “Management Agent(s)”
- Handles notifications from “Management Sub-Agents”, and forwarding to “Management Agent(s)” when appropriate.
- Manages filters for and forwarding of notifications to “Management Agent(s)”.
- Supports the concept of general monitors for elements of a “Management Sub-Agent”
- Exposes a collective management interface representing all “Management Sub-Agents” for use by all “Management Agents”.
- Provided by SI product.
- Provides a point of implementation and control for “store-wide” policy. For example: software distribution, Notifications, Monitor control,
- Normalizes data for public consumption

3.3.3 Management Agent

- Is a user of the “Management Agent API” for interacting with “Management Sub-Agents”
- Central point of contact for its corresponding management application
- Works as a communication proxy for the “Management Application”
- It is the single presence of the Management vendor’s toolset in the Store Integration Framework environment
- Point of management control for everything in the Store Integration Framework
- Provided by Tivoli, ThinkVantage, or others

3.3.4 Management Application

- Located in the enterprise or the store
- Provided by Tivoli, ThinkVantage, or others

3.3.5 Solution Install and Upgrade

The solution install process has a number of key steps which are performed at various times by various role players.

	What	When / Who
Software Installation	Loading the files and configuration for products that a solution depends on. Associated with base middleware products such as WebSphere, MQ, and DB2. Most often accomplished through Install Shield programs.	Normally provided once by the solution deployer, and then infrequently modified via patches or upgrades.
Software Distribution	Moving files from a central location to a potentially large number of distributed nodes in the network.	For example, large companies may use special tools such as Tivoli Configuration Manager in order to efficiently install copies of applications to hundreds of machines at a time across the network.
Solution Deployment	Focused on providing the implementation files and configuration for the software components that have been specifically developed to execute business processes associated with a solution.	Used by solution developers during development and testing, testers during test cycles, and solution administrators when deploying into production. Used in incremental fashion whenever any aspect of the solution implementation changes.

Solution distribution, installation and configuration will be performed using a combination of Tivoli Configuration product and various scripts and programs.

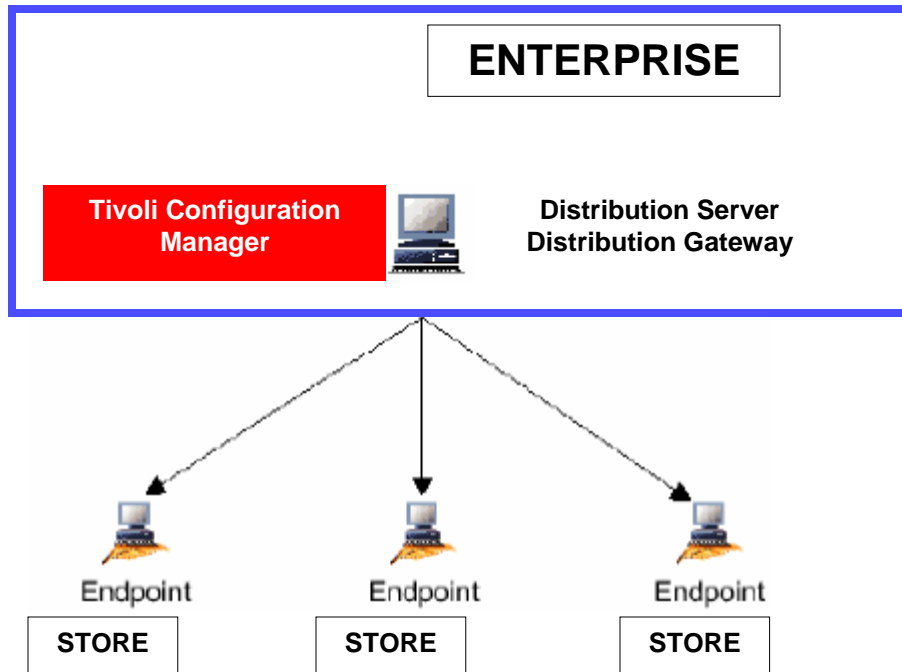


Figure 3: Store Integration Framework: Solution Deployment

For very large installations, the distribution server and the gateway server may have to be separated. In the short term, this solution has to deliver the following scripts or programs to help with the installation process:

- Install scripts for the middleware products i.e. Websphere Application Server, DB2 and Websphere MQ. These scripts will be developed in the context of the Tivoli Configuration Manager framework. Any install or update image greater than 20 MB, should be shipped to the store via CDs. During a new installation, we may decide to pre-install the middleware stack and using the remote mechanism to configure the products.
- Scripts to distribute the applications running on the in-store processor. These include things like jar, ear and war files for applications running on WAS, ddl scripts for applications running on the database server, etc.
- Scripts to deliver software stack running on the POS server. In most cases this will be the 4690 based applications and also the 4690 operating system. These programs will be delivered from the enterprise to the in-store processor, after which the in-store software distribution system will take over.
- Scripts to deliver the software stack running on the webpad devices and also the RFID Edge Server. Again, the software gets delivered to the in-store processor, after which the store system management infrastructure takes over.
- Installation/configuration scripts for applications running on our middleware products. These include the following:
 - For the applications running in the Application servers, there need to be Java Command Language scripts to install and configure them.

- For the applications running on the database server and MQ server, applications script need to execute that are associated with the products which were delivered to the in-store processor during the delivery of the complete application.

3.3.6 Operations Administration

This section describes the management of the store from the enterprise. The management of the store is based on a combination of Java Management Extensions (JMX) and Service Management Framework (SMF), which is IBM's implementation of the OSGi Standards. JMX is a java standard and is the preferred way to manage and monitor Java applications. The Websphere Application Server supports JMX.

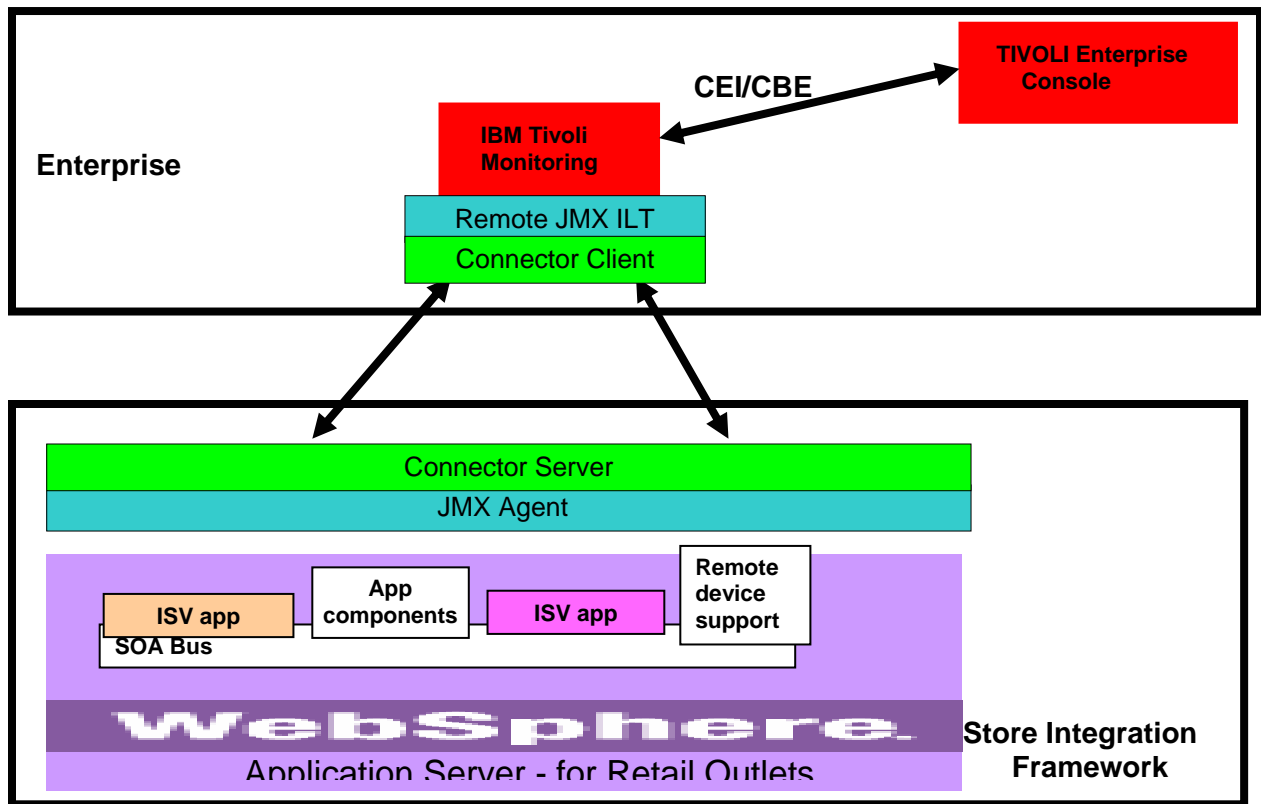


Figure 5: JMX enabling of IBM Tivoli Monitoring

A conduit has to be provided for IBM Tivoli Monitoring (ITM), one of the key components for management, to understand and interact with a JMX interface. The conduit interfaces between the JMX Agent and ITM, so that management information can flow freely. The Remote JMX ILT (as shown in the figure above) is such a component; it enables a Resource Model to retrieve resource instance property values from a JMX instrumented resource, i.e. the application. The ITM engine instantiates the Remote JMX ILT and it executes in same JVM as the ITM engine. The Remote JMX ILT connects to and submits requests to a JMX Agent that is running in a JVM other than the ITM engine JVM, thus the *remote* nature of accessing the JMX Agent.

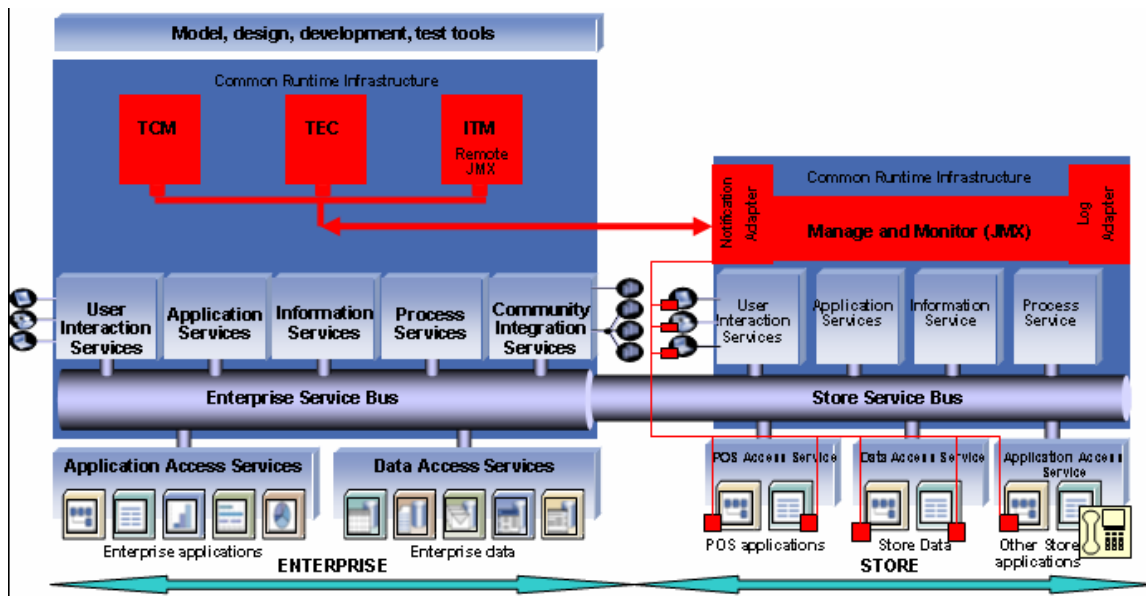


Figure 6: Manage and Monitor

Figure 6 shows the management infrastructure. There are various components in the store and the management infrastructure has to be capable of handling all the different components and environments. At the enterprise level, the management of the various store components is done using various Tivoli product; IBM Tivoli Monitoring and Tivoli Enterprise Console. The various management tasks include:

- **Websphere Application Server:** The application server has to be managed remotely. This is done using the TMWI (Tivoli Monitoring for Web Infrastructure) product. Tasks performed are:
 - Start, stop, restart, and retrieve the status of the Web servers.
 - Monitor key performance and availability of virtual servers hosted by the Web server
 - Forward critical events to Tivoli Enterprise Console
- **IBM DB2 Server:** The DB2 sever installed in the store has to be managed and this is done by using the Tivoli's Proactive Analysis Component (PACs) for DB2, TMDB (Tivoli Monitoring for DB2). Tasks performed are:
 - Start, stop, restart, and retrieve the status of the database servers.
 - Monitor key performance and availability of database servers
 - Forward critical events to Tivoli Enterprise Console
- **Linux on the in-store processor:** Another Tivoli PAC for Linux, TSAfl will be utilized to monitor the health of the operating system.
- **WAS Applications:** For each of the WAS based applications, MBeans (instrumentation layer conforming to JMX) will be available and interaction with the application will be done via the MBean interface. These MBeans should be able to interact with ITM.

- 4690 OS and Applications: These components are managed via the MBeans being developed as part of the Store Integrator solution. They will plug in into the Store Integrator Management Framework, and be managed via ITM from the enterprise.
- Store Devices: Webpad and RFID Edge Server have to be managed as part of the infra-structure. MBeans will be developed as part of the solution that will wrap the OSGi drivers and management will be performed via these MBeans.

3.3.7 Monitoring and Troubleshooting

This section describes the monitoring aspects of the system management. The monitoring of all components part of the store integration framework solution is based on JMX. The various standard JMX monitors will be used by the components.

Notifications from the software and hardware components need to get generated both for troubleshooting and also for health monitoring purpose. Various log files (infrastructure and applications) that get created by the components need to be monitored and events have to be generated from them. A number of components deployed in the store participate in the monitoring infrastructure and will generate events and notifications that get propagated to the system management application at the enterprise. They include:

- IBM Tivoli Monitor (ITM) for WI: The ITM for WI product from Tivoli is used for monitoring the logs that are generated by the Websphere Application server and it automatically generates events and notifications that flow into the enterprise console.
- ITM for DB: This is the component that monitors the logs generated by the database server and it in turn generates events that can be consumed at the enterprise.
- Logfile Adapter: The logfile adapter is used to monitor the various application logs and it generates events that flow into the enterprise console. The logfile adapter is delivered as part of the Store Integrator Product.
- Notification Adapter: When the JMX monitors associated with any component is triggered a notification will be generated. All notifications inside the store are collected at a central location and persisted. These notifications need to be propagated to the enterprise and this is done using the notification adapter. The notification adapter is delivered as part of the Store Integrator.

The logfile and notification adapter generates events that are consumed at the enterprise by the Tivoli Enterprise Console (TEC) application. TEC is responsible for generating the actual notifications and these are sent to the relevant administrative staff.

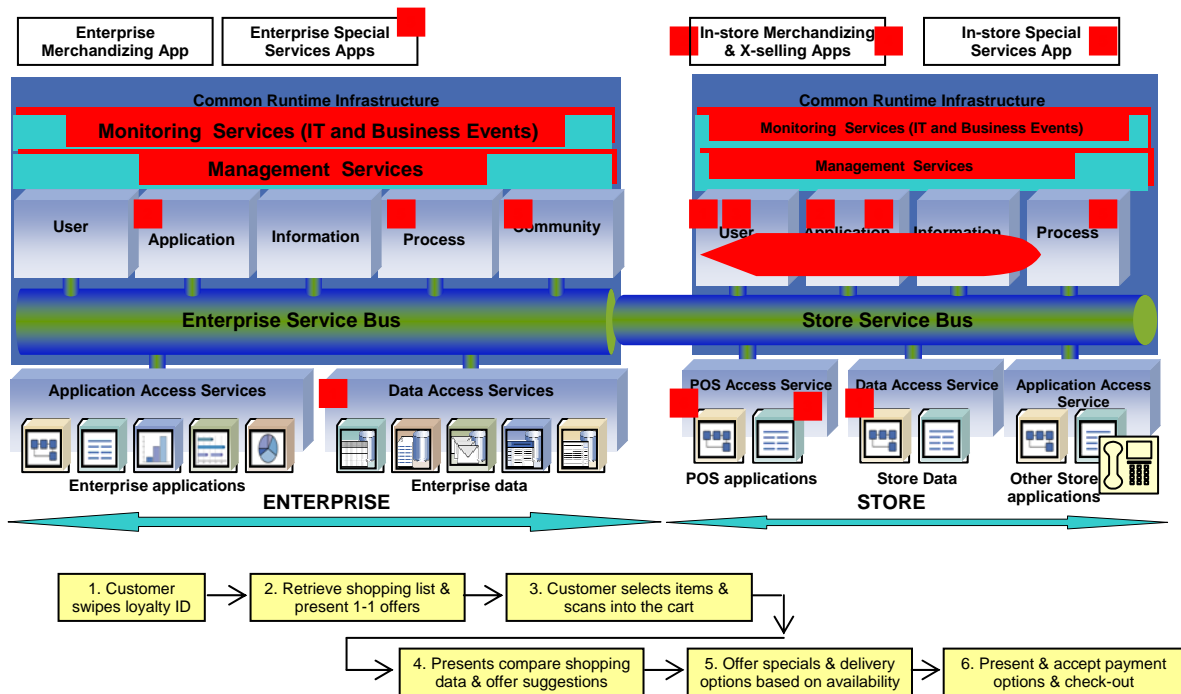
4 Solution Opportunities

The reference architecture for total solutions is built on the family of products that make up the Store Integration Framework. Here are some definitions of the solution opportunities that have evolved from this analysis. These need to be married to the segment and opportunity evaluations coming from marketing; however they demonstrate an ability to move quickly in some areas which have evident industry interest. These scenarios are a walkthrough of how each solution will use the framework

4.1 Customer Loyalty / Relationship Management

One use of the consumer service device is to assist the shopper and to encourage increased purchases based on the immediate availability of information, shopping aids, and/or promotional offerings. The key to modifying purchase behavior is to provide up sell / cross sell to the customer, while they are still shopping and prior to approaching the checkout. The following scenario illustrates the interaction of the Store Integrator Framework components to provide these services.

Example: New Shopping Experience



1. Customer scans a loyalty ID

- The User Interaction Service enables the customer to enter the loyalty card number from multiple device types (Magnetic Stripe Reader, keyboard input, barcode reader, or biometric)

- User profile is retrieved through Data Access Services from Store Data (shopper's data is cached in his/her "home" stores) or from enterprise data if the customer is visiting a new store. The Store and Enterprise Services Bus makes this 2-step search easy to implement for the developer.

2. Retrieve shopping list and present 1-1 offers

- The Cross Selling app built on the in-store Application Service (e.g., could be done through an "Offer" Business Object) is invoked through either input from a kiosk or a Shopping Assistant PC
- The Cross Selling application retrieves data and present static offers (daily specials, etc.) based on the in-store data
- The Cross Selling application uses Application Service on the enterprise side to develop dynamic, 1-1 special offers (e.g., an offer made specific to the shopper based on a combination of multiple data sources)

3. Customer selects item

- The User Interaction Service enables the customer to (1) scan an item to add it to the virtual shopping cart, and (2) browse and/or select the cross-selling items presented

4. Present comparison shopping data and offer suggestions

- The in-store Merchandising Application, retrieves and presents comparison data (e.g., comparing various brands of appliances to assist shopping decisions), or it can retrieve and present companion data (e.g., recipe or health eating guide) based on the customer's shopping selections
- The Catalog data could be in store or in enterprise. The framework enables transparent access to either data source.

5. Offer specials and delivery options based on availability

- For items that require special ordering (not in stock) or special services (non-SKU items, such as measurement or delivery/installation services), the in-store or Enterprise Special Services Application can use the Process Service to execute a process flow, e.g., place order, schedule delivery and installation, etc.
- The special services could be provided by a third party contractor, and the scheduling can be done through a B2B connection through the Community Integration Service, e.g., connecting to a Web Service API on the partner's web site.

6. Present and accept payment options and check-out

- Consumer moves to POS to identify themselves and to handle tendering for their transaction. POS business logic accesses the held transaction and automatically plays back all items selected.

- The Payment service in the in-store POS application is exposed for access via the Store Service Bus to present and accept payment services. The "Sales Transaction" Business Object, which could be exposed as an Application Service or made available from the POS application, is used to process and close the transaction.
- No rescanning is required. Tendering is handled simply and using established store procedures.

4.2 Price Checker Function

SI includes a web service type access for calculating the exact price that will be applied when an item is presented for purchase at the POS. This service supports many back office and enterprise process which require the up-to-date store price. The criteria for creating the price response is to include any price delta / calculations which are included in the POS price calculation. These factors must include in-store price file data, store promotions, coupons, and/or customer loyalty influencers. The service is also able to process a list of items in a single request to maximize response time and minimize network load for providing an accumulated web cart.

4.3 Kiosk Integration with POS

There are high value, specialty kiosks that retailers and kiosk providers would like to have integrated with the in-store point of sale systems. Store Integrator provides the functionality to do this integration for 4690 applications. The requirement for integration includes pricing, payment, and sale transaction processing. Clearly, IBM's POS presence in the US market represents an attractive leverage point for third party kiosk solution providers, and the framework provides the necessary interfaces for integration.

4.4 Returns Management

There is great potential ROI for helping retailers to manage returns. Returns management requires a transaction database to be established at the enterprise level to provide specific information on the original transaction. Retailers are interested in doing analytics about returns. Also, it is critical to manage the refundability of returned items.

4.5 Virtual Inventory

There are solutions for Virtual Inventory which allow web ordering from the store and having the merchandise delivered. This transaction ideally can be included in the purchase of other items in the store by tying the sale to an in-store POS transaction. This capability would be based on the Store Integrator GUI and 4690 applications to tie internet purchases through an enterprise inventory database into IBM POS.

5 Performance Capacity and Retail Environments

5.1 Traditional retail environments

Stores served by the Enterprise

The design point is up to 1000 stores per enterprise or sub-enterprise.

Customers per store

Historically, store information technology has not been concerned with the actual number of shoppers in the store, but only the number who are actively checking out or waiting to do so. For portable or per-cart devices, however, the number of customers in the store at a time becomes a consideration. We estimate the extremes of our range to be 5 to 100. That is, our smallest system would support at least 5 wireless devices and our largest 100. (The lower bound of 5 comes from the consideration that even the smallest retail store does not want to have to upgrade hardware when expanding from 2 mobile devices to 3, for example. 100 is the reasonable marketing requirement for the number of assisted shoppers regardless of the number of shoppers in the store. No retailer would purchase enough devices to cover every shopper in the busiest hour of their busiest day of the year.)

Consumer service devices per store

The design point is to support up to 100 consumer service devices in the store.

Multiple 4690 Controllers may be required to provide adequate CSS resource to access to legacy POS business logic for this configuration. It is likely that an additional controller will be required per each 50 persistent connections for a consumer access device to legacy POS function provided.

5.2 Web application measurements

Pages served per transaction

For stationary POS interface, per-item screen updates are assumed to be handled under local Java application control. For the consumer service interface, when an item is scanned, the screen will be updated through a refresh of the web page coming from the server. Per-transaction screen changes depend on the transaction type, but are generally on the order of 10. This figure is based on the average number of items per transaction since the screen will need to be updated at least once for each item scanned. There will also be screen updates for the scanning of the loyalty card and finishing the transaction.

For acceptable performance, we are targeting the size of the web page to be 10K bytes.

For the consumer service web apps the GUI responsiveness may include visual content other than the full page rewrite response. Either a single spinning globe or a transmission status bar that gives information on how much data has been downloaded and how much is left to download would add satisfaction to the consumer experience. See the [Consumer Service Application Specification](#) for details.

5.3 Connection Bandwidth

Store to Central

- 128 Kbps from store to central is assumed.
- This implies that not many shoppers can have simultaneous connections outside the store

- Connectivity options available to retailers: (availability varies by location)
 - Fractional T1
 - 128K
 - 256K
 - Cable (type 3 Mbps down, 150 Kbps up)
 - 2-way Satellite (type 500 Kbps down, 150 Kbps up)

Store Server to Clients

- 10 Mbps legacy ethernet LAN between wired POS terminals and store server LAN utilization by point-to-point traffic should not exceed 10%
- 802.11b to mobile devices (11 Mbps)

5.4 Hardware configurations

The aggregated cost of the deployed hardware throughout the retailer’s stores is a critical success factor for a solution. The recommended minimum hardware capacity for store systems is documented below.

5.4.1 4690 OS Terminal:

The terminal using SI Application Extension Facility has the minimum requirement of a 300 Mhz processor and 96 MB memory. When selecting new terminal hardware to allow for future growth using Java, it is recommended to choose a 1GHz processor with 512 MB of memory.

The Store Integrator GUI requires the ICEsoft AS Browser for IBM 4690 OS, Version 5.4 and a minimum 500 MHz processor with 192 MB of memory for a single video display. Dual video displays require a minimum 850 MHz processor with 192 MB of memory. In addition, minimum terminal RAM with produce images is 224 MB.

5.4.2 4690 POS Controller/Terminal:

The minimum requirements listed in this section represent the minimum supported configurations for use with Store Integrator AEF extensions. The actual hardware requirement for an AEF-extended application is highly dependent on the size of the base application and the extension code itself. Therefore, the minimum supported hardware may not be sufficient for some applications. When selecting new hardware or upgrading existing hardware for use with Store Integrator and allowing for future growth in functionality, the following hardware is recommended:

- Terminals: 1 GHz processor with 512 MB or more of memory
- Controller: least 2.4 GHz processor with 512 MB or more memory
- Platform: IBM 4690 OS V3R3 CSD1 or later

5.4.3 In Store Processor / Web server:

The in store processor which provides the web application server environment and necessary store databases requires the following as a minimum configuration:

- Server: 1 GHz Pentium III, 1GB Memory, minimum of two disks
- Platform: WAS for Retail and IBM Retail Environment for SuSE Linux V1 or later, or SuSE Linux Enterprise Server 8 service pack 3 or later

5.5 Data Currency

Data currency is measured by timeliness of data from the real time operation in the store to availability at the enterprise level. The following requirement applies only to stores with persistent connections.

Spec name	Description	Requirement
Timeliness of central data	Time from transaction complete until the data appears in the central RDB	2 minutes

5.6 Response times

Key to the retailer’s satisfaction with the store systems is adequate response time for servicing the customers. At checkout, the response time to scanning will impact the transaction time, which directly impacts the length of the lines for service. When we move to consumer service devices, the acceptability of performance is critical to expecting the customer to use and interact with the device.

5.6.1 Traditional POS response time requirements

These are summarized in the chart in the Appendix. In this section, we focus on the new requirements associated with Consumer Service devices and the framework.

5.6.2 Remote data access for new business function

General GUI Responsiveness Requirements

- 1 second response time for most simple data entry or screen-changing requests.
- For tasks that take more than 1 second to complete, the program must provide some visual acknowledgment that a request was accepted (e.g. Hourglass or change in cursor)
- For tasks that take more than 5 seconds to complete, the program must display a “processing” message.

For rare, exceptional tasks that take more than 15 seconds to complete, the program must display a progress bar or other estimated-time-to-complete indicator.

6 Store Integration Framework Glossary

- **ASTRA** - Advanced Store Touch Retail Application; a Java GUI package for existing legacy 4690 POS currently delivered as an RPQ.
- **CBC** – Common Business Components are services provided to implement the Services Oriented Architecture and provide value in the store operating environment for integration of ISV applications.
- **CSS** - Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colors, and spacing) to Web documents.
- **CSS** – Client Session Server (CSS) is the component of the Store Integrator which hosts the 4690 terminal business logic in a virtual session on the controller to provide the services necessary for integration and reuse of the custom checkout process.
- **Consumer** - the retailer's customer. The person who shops and will be using the in-store wireless service device.
- **Customer** - IBM retail customer of store systems IT equipment and software. Examples are Food Lion, Wal*Mart and The Gap.
- **Consumer Service Application** - One of the Store Integrator reference implementations based on Websphere application server which provides the consumer with ability to create a POS transaction while shopping. Also referred to as the **Shopper Assistant** or **Portable Shopper**.
- **EJB object** - An EJB is an Enterprise Java Bean, a distributed, persistent, software component. It is a Java object that is known in the networked environment and is easily accessed remotely.
- **ITM** – IBM Tivoli Monitor is the enterprise management tool recommended to provide a full functioning monitoring capability for the network of stores.
- **IXRetail** - The NRF endorsed industry standard definition for XML schema to be used for retail business data exchange
- **J9** - Embedded J2ME JVM supplied by IBM.
- **JavaPOS** - the NRF endorsed industry standard definition for the API for Java access to POS devices.
- **JAXM** - Java API for XML messaging specification, which is used by the Data integration component for messaging.
- **Linux** - Open Source Unix-like OS.
- **MQ Series** - IBM SWG product for guaranteed messaging. MQ Everyplace is a Java version which also runs on the 4690 OS.
- **Personal Pricer** - One of the reference implementations based on Websphere application server which provides a customer / coupon / offer based price per item scanned.
- **POS Application Extension Facility API** - Base architectural component of Store Integrator which provides the Java API for integrating new function with legacy POS.

- **RSS** - Retail Store Solutions, the IBM division focused on building and marketing IBM's retail store solutions.
- **Store Integration Framework** - Name for IBM software strategy to extend the legacy POS, integrate the store operations with the enterprise, and support consumer access devices. The Store Integration Framework includes infrastructure services and business components to enable a total solution for Assisted Shopping and Personal Pricer.
- **SDI** - Strategic Driver Initiative, the term used to identify the Java design and implementation for the RSS peripheral device support.
- **SOAP** - Simple Object Access Protocol is the standard for interfacing between web services using XML messages.
- **SSL** - Secure Socket Layer is a TCP/IP protocol used for transmitting secure data that is encrypted with a public key. It is a standard supported by many browsers for private documents.
- **SureVision** - Java GUI for existing legacy 4690 POS currently delivered as an RPQ.
- **URI** - Uniform Resource Identifiers (URIs, aka URLs) are short strings that identify resources in the web: documents, images, downloadable files, services, electronic mailboxes, and other resources. They make resources available under a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail addressable in the same simple way.
- **URL** - See URI
- **Web application server (WAS)** - from an architectural view point, the solution for Consumer Service Devices requires a J2EE compliant platform to support Java Server Pages and servlets. The Store Integration Framework solution will require an IBM WebSphere platform for the solution. There may be configurations where this requirement could be met with a non-IBM platform (i.e. BEA) or an open source alternative.
- **WebSphere 5.0** - IBM's premier development tool kit and application server environment for deploying web services. It provides for integration of distributed processing between J2EE and Microsoft .NET environment. It also integrates MQ Series Integrator for message brokering and business process integration technology from CrossWorlds. It supports SOAP and WSDL standards.
- **WebSphere Express** - Minimalist version of WS 5.0.x that requires smaller footprint and less administration than WS 5.0.x.
- **WebSphere Application Server for Retail Outlets** - A specially priced package which mirrors WAS Base function for installation into retail stores.
- **WSDL** - Web Services Description Language is used to identify the characteristics of the web service to the network.
- **XML** - eXtensible Markup Language, a tag language for handling distributed data.