Best Practices and Tools for Creating ATG Commerce Sites

ATG Professional Services

May 2005

atg **Professional Services**

Contact

ATG
25 First Street
Cambridge, MA 02141
617.386.1000 phone
617.386.1111 fax
www.atg.com

atg

Table of Contents

1 Introduction

ATG Services would like to congratulate you on your selection of the ATG Commerce Software platform. ATG Services works with our customers and partners to help design and implement solutions based on ATG technology. Our mission is to help you exploit the full potential of your ATG solution and get the maximum return on the investment you've made.

ATG Services has unparalleled, real-life experience with ATG technology. Bringing together a wide range of capabilities - from project management, to engineering consulting, to dynamic design, to deployment support – ATG Services helps insure the success of your implementation, launch and application management. Our proven methodology is based on our experiences and our best practices, distilled from successfully managing hundreds of ATG projects.

The ATG Commerce application serves as the foundation for your online store. It contains everything you need to manage your product database, pricing, inventory, fulfillment, merchandising, targeted promotions, and customer relationships.

ATG Commerce is available in two versions. ATG Consumer Commerce is used for developing standard business-to-consumer (B2C) online stores. ATG Business Commerce is used for sites oriented more toward business-to-business (B2B) uses.

ATG offers a flexible, modular delivery model that can be adapted to the specifics of your commerce project - whether we're working along side your team, your partner team, or taking full responsibility for the project ourselves. Whatever the engagement, our goal is the same: to accelerate the time to implementation and maximize the contribution of ATG technology to your most important business goals.

The ATG methodology is not a silver bullet - but an organized process to help make it happen based on our prior experiences. At ATG, we strive to keep it simple. Partnership, Products, People and Process – these are the cornerstones of successful implementations. This document helps to address two of the four components – Process and Product.

ATG Professional Services works with our customers and partners to help design and implement commerce solutions based on ATG technology. Our mission is to help you exploit the full potential of your ATG commerce solution and get the maximum return on the investment you've made.

We have been in the business of assisting our customers in successful implementations since 1991. No one knows our software better than we do, and no one knows how to implement them more efficiently or more effectively. Our years of experience have enabled us to perfect an implementation approach to help ensure your success. The Adaptive Delivery Framework (ADF) Software Implementation Methodology is our recommended approach for maximizing the return on your investment - by ensuring a quick return on your implementation, and also doing it accurately as well. For further information regarding the ADF, please refer to the ATG Professional Services Implementation Methodology and Best Practices Customer Care Guide.

We know our part:

- Making your objectives – our objectives
- Providing leading practice solutions
- Identifying process improvements
- Providing complementary skill sets
- Introducing and helping you to support new technologies
- Effectively training your staff
- Managing project scope and expectations

- Transferring expertise
- Ensuring you are self-sufficient after live

An experienced partner during an implementation can mean the difference between success and failure. The Adaptive Delivery Framework Methodology represents a tried and proven process in implementing ATG software. Our five-phase methodology allows you and your project team to view the implementation not as an overwhelming task, but as five manageable phases, each with milestones, available deliverables and tangible benefits.

"Internet-oriented marketing matures to the point where search engines, email, instant messaging, and web logs all morph together to create compelling, always-on, and potentially fully mobile online interactions with savvy marketers." David H. Deans, Managing Director, GeoActive Group USA

"In the world of web sites, we'll see a major movement from first generation sites (where the enterprise focuses on making the technology just work) to next generation sites (where the enterprise focuses on making the site provide value to the audience). It's already starting with some of the leaders in key indicator industries, such as retail, insurance, healthcare, government, and higher education. The leading organizations now focus on designs that enhance the audiences' experience and are reaping tremendous benefits." Jared Spool, Founding Principal, User Interface Engineering

In terms of capabilities, ATG's personalization technology, which essentially creates an authoring and implementation environment for sophisticated client-driven workflows, is a unique fit with our customer's vision for creating customer-centric site experiences. No other product available provides the sophistication or the effectiveness of ATG's scenario model for creating business rules and scripts to enable effective segmentation-based and evolutionary marketing.

Often the terminology around personalization, scenarios and marketing campaigns are used interchangeably. ATG views it this way: personalization comes first, including the capture of customer profile behavior and buying behaviors that are required for a seller to become "personal" with a buyer. Without this information there is no ability to become personal or customer centric because you have no idea who the customer is or what they are interested in. Second, scenarios are the underlying ATG centric technology that enable both the capture (recording) of customer behavior (what they are looking at, what they have purchased, etc.) as well as the ability to modify the page display real time based on the unique behaviors of the customer. Lastly, marketing campaigns are the result of the use of personalization and/or scenarios based on the data collected (often analytics are used here to "mine" the data) and market segmentation used to define a target audience, map a product offering of interest to that target market, and finally present the product to the customer.

Follow up research should be used to determine which products are most effectively being sold to which target markets. This helps to facilitate the creation of the marketing campaigns (using the scenario engine) as well as provide results regarding the effectiveness of each campaign.

In summary:

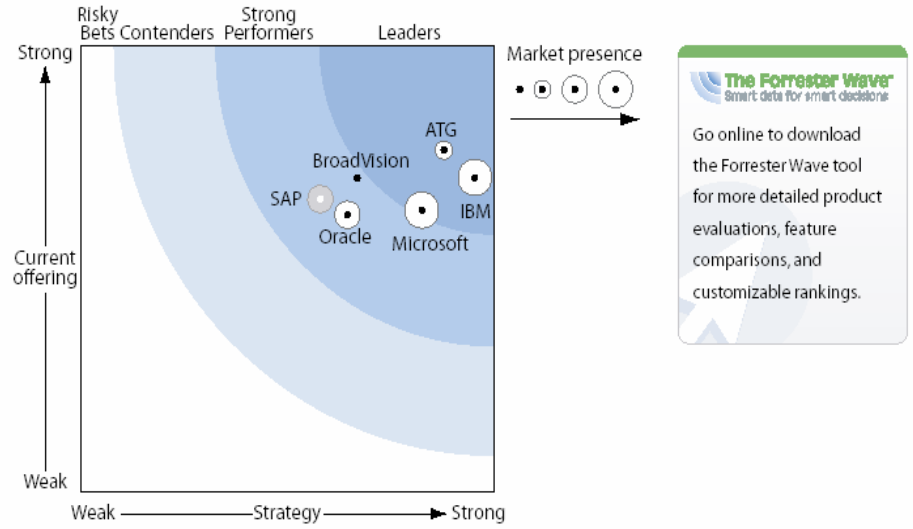
- Well constructed ATG sites can influence how much customers spend and how loyal they remain.
- A key to maximizing the value of ATG is to access the business goals of the site and the organizational and technical support it requires to achieve the business goals.
- ATG customers have reported double-digit gains in revenues, customer satisfaction, and employee productivity), along with dramatic savings in customer acquisition costs.
- ATG's personalization and Scenario technology, in terms of its intent and capabilities, is a perfect fit for our customer's vision of creating a customer-centric, personalized site experience.
- Use of personalization, including simple "If-then" constructs, have presented no known scalability concerns.

Our own view of sophisticated marketing campaigns via personalization and scenarios, and the emerging view among our leading customers, is that the use of personalization should be built incrementally to allow effective learning, organizational adaptation and maximum effectiveness. By ranking potential marketing

campaigns by ease of implementation and potential return, there is an appropriate prioritization that limits the total cost (including organizational costs) while maximizing return. More simply, we have not yet seen a situation in which it was necessary or productive for a company to implement thousands of marketing campaigns. While the ATG technology and the business needs of our customers may well evolve to the point where such a large volume of campaigns becomes useful, no doubt in the same timeframe performance capabilities will keep pace with the increased demand.

ATG Commerce has been listed by Forrester as the highest rated commerce offering as of Q2 2005:

Figure 3 Forrester Wave™: Commerce Platforms For B2C Sites, Q2 '05



Source: Forrester Research, Inc.

The following documentation will assist you in gaining an understanding of the five phases of the Adaptive Delivery Framework Methodology, and how, through using the methodology in conjunction with ATG Professional Services best practices, you too can be successful.

atg Best Practice

Throughout the Adaptive Delivery Framework methodology you will find the best practice symbol. This designates a key process that we believe is important to consider on your implementation and are based on our years of experience. This is also a critical juncture in the project when ATG Professional Services should be engaged.

2 ADF Overview

2.1 Objective of the Methodology

ATG Professional Services has developed an Adaptive Delivery Framework (ADF) to ensure consistent, high-quality service delivery throughout all our project engagements. The ADF brings together expertise and best practices, including project templates, existing work samples, and lessons learned. The ADF is used to create repeatable delivery processes from project to project to provide a consistent look and feel for all ATG project deliverables. We have devised well-defined processes to mitigate risks. This includes identifying potential bottlenecks in order to implement preventative measures and contingency actions. The ADF outlines clear processes for managing change requests so you have a clear understanding of how requested changes will effect the time and budget of the project. This helps to avoid schedule delays and cost overruns.

The ADF is client focused, giving ATG the flexibility to meet your specific needs within a proven delivery framework. We apply only those project "modules" which are appropriate for your project requirements. We continually update this framework to meet changing market requirements.

The Adaptive Delivery Framework Implementation is constructed around five major phases. Although you can anticipate that there will be overlap among the activities during each of the phases, the phases must be performed in the order presented. For further information regarding the ATG ADF Methodology please refer to the ATG Implementation Methodology and Best Practices Customer Care Guide.

2.2 Phase I – Analysis

During the Analysis phase, you should gain an understanding of your current business practices and identify areas for improvements. The objective is to develop a comprehensive list of requirements. The requirements definition can be developed independently of any current working knowledge of the software.

Also during this phase, the core project team members gain a thorough understanding of the delivered software application. They gain this understanding by attending formal classes conducted by ATG and participating in the extensive classroom hands-on practice sessions. With the knowledge gained in training and the requirements defined, the project team prepares the Functional Specification Document. The Functional Specification Document maps the functionality of the software to the requirements of your organization. Any gaps between the functionality of the software to the requirements of the organization should be highlighted and documented.

Included in this phase are the following activities:

- Defined Business Requirements
- Finalized Education Plan
- Preliminary Project Plan
- Project Team Training
- Document gathering – strategy, objectives, system and operating requirements
- Data Conversion, Interface & Enhancement Requirements
- Information Architecture Review
- User Experience Page Layout and Navigation

Also included in this phase is the completion of the following available deliverables:

- Functional Specification Document
- Trained Project Team
- Preliminary Project Plan
- Content Management Plan
- Preliminary Architecture Plan
- Prototype Issues Document
- User Experience Wire frames

2.3 Phase II – Design

During the Solution Design phase, the project team will design the customer solution, including database schema design, repository definitions, integrations and other technical issues, as well as page layout, templates of the primary interface design, and technical architecture specifications. This phase is driven by project goals and user needs as defined by the Functional Specification.

Also during this phase, ATG will also refine the requirements, define user scenarios for all user categories, complete the detailed technical architecture (including all integration points), create feature and technical specifications, and create a feature prototype. Also, a detailed project plan for the implementation phase will be prepared.

The deliverables from this phase include:

- Detailed Architecture Specification
- Detailed Technical Design Documentation
- Data migration, QA plan, risk plan
- User input and acceptance
- Personalization & promotions definition
- Complete set of screenshots and/or clickable HTML prototype
- Test Plan Outline

2.4 Phase III – Implementation

The Implementation phase consists of the preparation of the new system and users for going into production with the new software. This phase includes the actual coding of programs to convert data from existing systems to the new system, configuration definition, company setup detail definition and report writing, as well as planning for the actual cut over to the new system.

Included in this phase are the following activities:

- Conversion Development
- Interface Development
- Custom Enhancement Development
- Directory structure
- Modules
- Build scripts
- Naming conventions
- Site Testing
- Build release QA
- Unit Testing

Also included in this phase is the completion of the following available deliverables:

- Interface Programs & Documentation
- Data Conversions & Documentation
- Reports/ Inquiries & Custom Views
- System Test definition and plans
- System Test Report/Validation
- Operational and deployment planning
- Test Plan
- Documentation

2.5 Phase IV – Testing

During this phase, the entire customer solution will be quality-control tested according to the QA plan created during the Implementation phase. During this period, our customer will perform Acceptance testing, while ATG will provide bug fixes.

- Integration Test – Back –end systems
- QA Load Test
- Performance test
- User Acceptance Test
- Web application code fully tested and ready for deployment

2.6 Phase V – Deployment

During the Production Rollout phase, the project team ensures that the completed system is fully functional, satisfies user requirements, and achieves business objectives. Most importantly, during this phase ownership of the new system passes from the project team to the users.

After the application is live, this phase provides you with a formal project close-out and review to ensure that the system operates efficiently and continues to meet the requirements of the users. Changes in your business environment and changes in ATG Software's products are monitored and evaluated on a regular basis to maximize the return on your investment in ATG products.

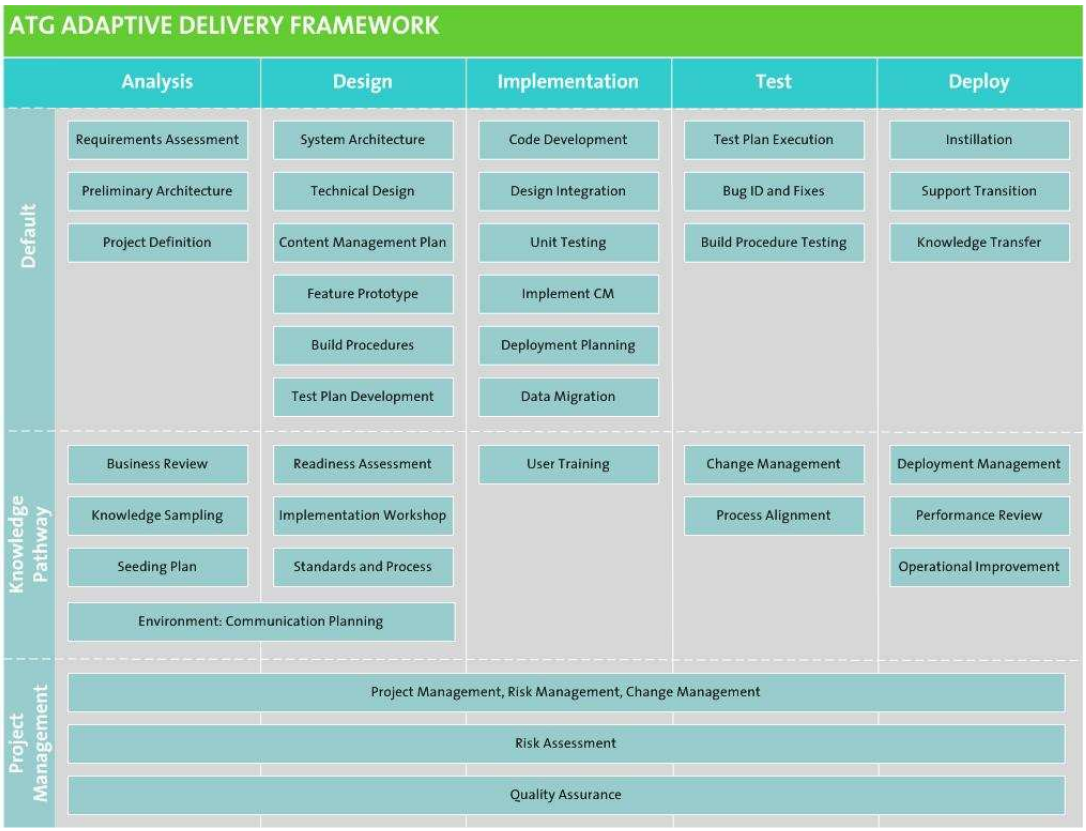
Included in this phase are the following activities:

- Production Cut Over Planning
- Acceptance Testing
- Production Cut Over Support
- Final Conversion
- Launch
- Post Implementation Review

Also included in this phase is the completion of the following available deliverables:

- Verified User Procedures
- Production Support Procedures
- Post Implementation Review Document
- Documentation for hosting provider or internal IT operations

3 Goal Setting and Planning for your Commerce Site



3.1 Narrative Overview

Prior to the project, your company spends time before the software is installed to ensure that the project incorporates the goals and expectations of the organization.

3.1.1 Sales to Implementation Transition

ATG has gained a lot of information about your organization and requirements during your software evaluation and selection process. You have gained a lot of information about ATG during the software evaluation and selection process. Perhaps time has transpired since the product and services surveys were completed. Assumptions may have changed during contract finalization. New members of your organization may now be involved with the project.

The very first step in Planning is to transition the information gained during the software evaluation and selection process to the project manager and key members of the implementation team. This is a very important step – as project issues later on can be avoided by ensuring proper expectations have been set by both sides.

A conference call or face to face meeting (depending on the size of the project) is appropriate. We recommend including your own project manager as well as ATG's Project Manager (and/or the system integrator) assigned to the project.

Key items/documents to be reviewed include:

- Identification of Project Manager and Client Project Managers
- ATG Master License Agreement
- ATG Customer Support Policy including hotline support number
- The implementation services proposal including project assumptions
- Business goals and roles of team members
- Project timeline or plan, if available
- ATG key contacts and roles (Account Executive, Project Manager, Education Coordinator, Regional Services Manager)
- ATG Electronic Customer Support
- Hotel accommodations, expense policies
- Billing practices
- Communication plans
- NDA in place?
- Master Consulting Agreement in place?

3.1.2 Goal Setting and Planning

One of the first major steps in the Planning phase is confirmation of your company's strategic business goals. Typically, your executive team defines these goals prior to licensing the software. ATG suggests that the executive team reviews the expected benefits of the implementation with the project team and identifies the issues the software is intended to resolve. At this point in the implementation cycle, goal setting and planning is often a top down process.

The risks to a successful implementation are determined by assessing the client's organizational readiness to undertake this type of project. Examples of high-risk areas, which directly affect the success of the implementation, include the following:

- Lack of prior implementation experience – contract for the appropriate skills
- Limited technical expertise – contract for the appropriate skills. Attend training. Use a mentorship project structure
- Lack of project ownership & accountability – effective project management and alignment
- Unclear or undocumented business goals – ensure proper time is given to the Functional Specification and the analysis phase
- Lack of or unstable technical environment – budget and determine proper hardware and configuration. Utilize a deployment specialist for the development environment setup
- Lack of dedicated project management time to the project – ensure full time project management is budgeted
- Lack of dedicated user time to the project – build the plans and timeline early. Communicate to the end users in advance of when their time will be required
- Lack of dedicated information systems time to the project – budget for and assume need for IS/network support

- Slow or bureaucratic decision making process – schedule weekly status meetings. Ensure a project Steering Committee is in place
- Unreasonable expectations – validate the timeline and effort estimates with ATG and/or your implementation partner. Create buy-in from the project team of milestones and deliverables
- Inexperienced project management - contract (or hire) for the appropriate skills
- Lack of proper production operational support – perform a deployment review prior to launch. Contract (or hire) the appropriate skills

ATG recommends that you assess the level of risk associated with your implementation. As areas of weakness are found, suggestions for addressing risk are discussed. High-risk implementations often require additional assistance from ATG Professional Services.

The project team confirms and documents the project goals as a method of quantifying the success of the implementation. The project goals should support your organization's strategic business goals.

Ensure your company's long-term vision is included in the Goal Setting and Planning phase. Include in the plan your technical architecture, internet strategy, workflow requirements and data inquiry and report writing needs and strategy. Review with ATG what needs to be done and how these can be accomplished with ATG or ATG Software's complementary products.

ATG also recommends that the project team develop a Project Charter, which formalizes the guidelines for conducting the project. This serves as the guide throughout the implementation to keep the project team within the established rules for conducting the project. In addition, you should define a Project Scope Statement to establish the boundaries of the project. Scope creep is a major factor that often increases project duration and expense.

The final step in this phase is to develop a macro plan for the project. In the macro plan you define high level target dates, responsibilities and available deliverables. Managing the three key variables of time, scope and resources (both people and dollars), and setting the proper expectation for each, is key to a successful implementation. ATG's standard is MS Project. Using this tool will allow you to easily share project plans with ATG.

3.1.3 Project Organization

atg Best Practice

The next step in the Planning phase is the building of the project team. A Steering Committee of executives and top management is essential to provide review and approval at critical points during the implementation (audit checkpoint). A Customer Executive Sponsor is also required as the primary liaison between the project team and the Executive Steering Committee.

ATG recommends that a full time project manager be dedicated to the implementation effort. Many projects have failed as a result of underestimating the importance of this position. This can be either an ATG or customer Project Manager. This position can be successfully split between ATG and the customer as joint co-Project Managers, and is recommended (best practice). The co-Project Manager structure can drive the responsibilities on behalf of both the customer and ATG.

If the customer is unable to provide a project manager and the technical team required, ATG works with a number of independent system integrators that should be used as the project lead, business analyst and user experience (UI) roles while ATG services is utilized for the technical lead positions (architect, technical lead, deployment and some development) of the project.

Adequate application and technical resources must be assigned to the project team to ensure success within the specified time frame. Especially on multiple product installs, technical project members often become "single threaded" and are the point of resource contention. Make sure proper technical resources are assigned

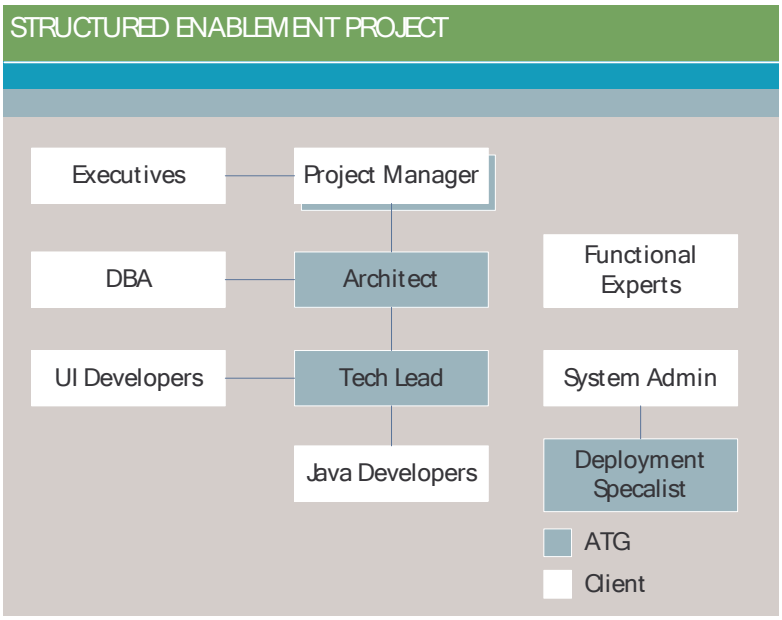
for multiple product installs for the work to be properly divided. A DBA or database administrator is a role within the customer IS department that can handle the setup and maintenance of the database schema. This position carries significant importance in ATG implementations. The assistance of key users throughout the implementation is also essential. This group comprises the "core project team" to be trained up-front, and will be responsible for the proper rollout and education of others.

Keep the "core" project team to a group of individuals within your organization who have the experience, authority, and commitment of their time to the decision-making and setup of the software.

User participation ensures that the system will satisfy their business objectives. The users make the decisions and approve key areas of the system setup. The project team should arrange for user involvement during this step to prevent project delays.

Communication is an important element in the success of any project. A formal method of communication between the project manager and the ATG Consultant is required. It is also imperative that communication channels exist between the project team and the members of the organization impacted by the project.

The following diagram depicts a combined ATG/client project structure (referred to as structured enablement):



3.1.4 ATG Education

Best Practice

ATG Education is responsible for developing and delivering training programs for all of ATG's customers, partners, and internal staff. We have trained thousands of developers in ATG and related technology. To examine our standard training programs and offerings, please see our education website at http://www.atg.com/en/education

ATG's educational offerings include:

- Instructor Led Courses: ATG's role-based curriculum is designed to address the needs of every member of your team - business managers, system administrators and developers. Courses are characterized by informative lectures and instructional labs. This curriculum is composed of 15 courses and a total of 28.5 days of instructor-led training.
- Students can take classes at any of ATG's training centers located a cities around the world. Locations are, Cambridge Massachusetts, New York New York, Chicago Illinois, San Francisco California, Reading UK, Paris France, and Frankfurt Germany
- Training can also be delivered at your site. This is a convenient and cost effective way to train groups of students at the same time. Onsite training provides the additional benefit of allowing the course delivery to be customized for your needs.
- Every technical course has at least 40% hands on labs to maximize your training experience
- ATG also delivers select courses in a Virtual class setting
- ATG Certified Professional Program: This program sets the standard for ATG knowledge within our customer and partner community. These computer-based exams are available at Sylvan Prometric testing centers around the world.

Ensure that your project team is properly trained leading into the implementation.

3.1.5 Project Kick Off

To introduce the project to the team members and others involved with or impacted by the implementation project, ATG recommends conducting a project kick off meeting. This orientation at the beginning of the project ensures that all project team members understand the project: the scope and objectives, organization and work effort, and the use of standards and procedures. The Executive Steering Committee, Executive Sponsor, project team members and other key users participating on the project should attend this meeting. Typically the Project Manager runs this meeting.

The agenda of the meeting includes the following:

- Presentation of your strategic business goals and objectives
- Presentation of the project goals, Project Charter, Project Scope Statement
- Presentation of the Implementation Methodology (approach)
- Presentation of the project budget, anticipated live date and expected resource commitments

As a result of this meeting each person in attendance gains an understanding of the scope of the project and the roles of the project manager and the project team. Keep in mind that the projected budget, anticipated live date and resource commitments may change as a result of the Analysis and Design Phases. The idea here is to communicate the high level expectations of the project. Keep it upbeat; bring in lunch or donuts and coffee to introduce the excitement and benefits of the project. If visual screen shots of the new site are available, show them to the team to help them get excited about the new site.

The Executive Sponsor should lead a general discussion to outline the benefits of the project and to emphasize the importance of the project to the organization. It is also important to communicate this information not only to the project team, but also throughout your entire organization. The Executive Sponsor should be aware of any issues that may be detrimental to the project.

As part of the meeting the project team reviews the macro plan (very high level). All changes to the plan require approval by the Executive Steering Committee.

3.1.6 Project Planning

The initial project planning presents the first opportunity for the core project team to plan and document the project in greater detail. The core project team is a group of individuals within your organization who have the experience, authority, and commitment of their time to the decision-making and setup of the software. This meeting provides the opportunity to outline the "administrative" aspects of the project, present and discuss project management concepts, establish regular meeting times, and define a method for recording and reporting project status.

It is important to note that Project Planning encompasses the entire project – all modules of the software to be implemented. Key decisions are made during the planning concerning time, scope, and resources. Multiple product installations/integrations require special consideration due to the complexities of managing and integrating many application areas.

The solution outline may be combined with macro design, or macro design and micro design may be combined. The main focus of the macro design is the creation of an ATG Commerce design that incorporates the requirements during the analysis phase. The project team may follow one ore more of the following design methods:

- System level (for example ATG Commerce interacting with external systems)
- Subsystem level (for example, interactions among ATG Commerce components such ASE and Content Administration)
- Use Case level (for example, site flows as identified in use cases)
- Component level (for example, interactions among JSPs and EJBs)

During the Project Planning your ATG Consultants will assist in the development of a project plan which will incorporate the detailed tasks. Decision makers from both the business and IT areas should attend the planning meetings. At this time, it is practical to plan through the completion of the Solution Design and Design phase of the project. Additional training is required before planning beyond this phase can be accurate.

A finalized education plan is also produced during the Design Phase detailing the project team enrollment requirements in ATG education courses. If training is to be performed on-site, training weeks are determined to assist ATG to schedule the appropriate training resource.

The project team should accomplish the following tasks as a result of the initial project planning :

- Document procedures for issue tracking, issue resolution, and change management.
- Update the project plan.
- Introduce the concept of contingency planning in the event of project slippage.
- Assess project team training requirements.
- Schedule prototype planning days by defining tasks and milestones for the prototype and for each application within the scope of the project.
- Obtain approval from the Executive Steering Committee on the project plan and the required resource commitments from other departments not directly assigned to the project team.
- Schedule the Business Requirements review meetings with your ATG Consultants.
- Schedule the first Risk Review

3.1.7 Technical Environment Readiness

A crucial step in the planning phase of an implementation that is often overlooked is the development of a plan for the purchase, installation and use of the necessary hardware. This plan includes planning for the servers and infrastructure required to run and scale you web site. The plan also determines the necessary electrical work to be done, communications and networking requirements, additional cabling, and other hardware requirements. If a hosting provider will be used make sure they are involved early in the process for planning purposes.

The following tasks are recommended:

- Order the hardware (Servers, PC's, RAM, hard drives, printers, UPS, etc.), computer supplies (paper, diskettes, etc.), contract services (electricians, etc.). Keep in mind that vendor lead times may impact the project schedule.
- Obtain the system software (Server and Network OS, DBMS, utilities, tools, etc.).
- Assign the computer operator and the system security officer. Schedule these individuals into the appropriate training classes (ATG technical/deployment) and involve them with your IT resources in the installation of the operating system, network operating system and the ATG software. Develop initial procedures to back up and restore the system during the implementation process.
- Gain an understanding of the current operational procedures that are within the scope of the project. Gather documents and reports that reflects the current system operations. Obtain a high level overview of the functions as well as any special processing requirements.
- If on-site training will be performed at your location, determine if the initial training will require special classroom consideration. If so, additional setup may be required (PC's, connectivity, memory upgrades, etc).

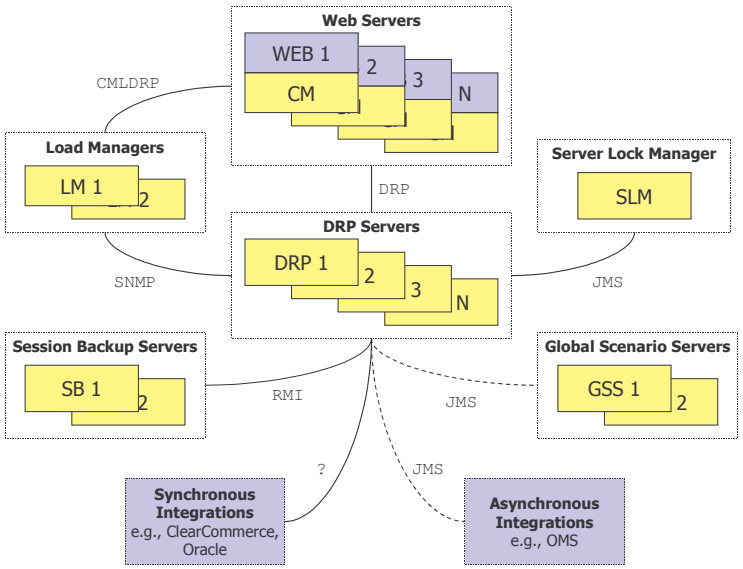
3.1.8 Environment Setup

Best Practice

This step includes loading and building of the development environment. The goal is to initially load the software to support the education and development activities of the project team. As part of the process, client development machines are configured for the project team.

To better understand the specified Dynamo configuration, it is important to understand the various application components that comprise a typical Dynamo installation. This section provides a generic overview and description of the different applications and services along with their basic requirements within a high availability environment.

When speaking about a multi-server ATG Dynamo installation, it is typical to talk about the number of servers required to handle client requests. In this context, it can be inferred that one is talking about a Dynamo Request Processor (DRP). In short, DRP servers are responsible for fulfilling dynamic page requests (usually .jhtml and .jsp). However, in a production environment, there are several other Dynamo component applications that play very important roles. The following diagram presents a high-level overview of these applications and how they coordinate within a multi-server environment. The following sections describe each component in more detail.



Dynamo overview. Items shown in yellow are ATG Dynamo components. Solid connection lines depict synchronous integration points. Dashed connection lines depict asynchronous integration points.

As noted previously, **DRP servers** are responsible for fulfilling dynamic page requests. This entails various services like page compilation, database connection pooling, page execution, etc. In short, this is where the bulk of the server activity occurs. When scaling a site, the number of DRP servers is tailored to meet the anticipated volume.

Generally speaking, there should be one DRP server for every 500-1000 concurrent sessions. Note that this range is heavily dependent upon several factors which are application dependent. Most notably, this metric is heavily affected by application complexity, code quality, and caching strategies.

The **Connection Module** is a Dynamo component that runs inside each web server. The CM is responsible for routing dynamic page requests to individual DRP servers. Additionally, the CM coordinates with the configured Load Managers (primary, secondary, etc.) to determine which DRP server should handle a new user.

For more detailed information see the chapters, Load Management and Integrating Dynamo with HTTP Servers, in the Dynamo Administration Guide.

A **Dynamo Load Manager** is a server instance responsible for managing the distribution of sessions and dynamic page requests across multiple DRP servers. Load Managers periodically poll each DRP server for status and current load information.

To meet high availability requirements, it is recommended that production sites run 2 or more Load Managers on separate systems.

See the *Load Management* chapter in the *Dynamo Administration Guide* for more detailed information.

Session Backup Servers are used to ensure a consistent user experience in the event of a DRP failure or an isolated hardware failure. In short, individual DRP servers can replicate a configurable subset of user session data to a particular Session Backup Server. In a failure scenario, user session data is migrated from a Session Backup Server to a surviving DRP instance.

There is a performance cost associated with enabling Session Failover, and it is not always worth the cost to implement. Based on requirements for each customer, this should be reviewed.

Server Lock Managers are responsible for managing distributed data locks when using Dynamo repositories with the "locked" cache mode.

To meet high availability requirements, it is recommended that production sites run 1 Server Lock Manager instance while maintaining a spare processor on a separate machine. In Dynamo 7.0, Server Lock Manager supports automatic fail-over capabilities, and it is recommended to have multiple Lock Managers to support this. In the event of SLM failure, locked caching is automatically disabled at each DRP server. This may increase traffic to the database, and conversely decrease site performance. However, the locked cache mode is used relatively infrequently, and SLM failures should be a rare occurrence.

For more detailed information, see the sections, Repository and Database Performance: Repository Cache Configuration and Configuring for Deployment: Enable the Repository Cache Lock Managers, in the Dynamo Administration Guide. Additional information can be found in the SQL Repositories: SQL Repository Caching section of the Dynamo Application Server Programming Guide.

Global Scenario Servers are a component of Dynamo's personalization engine. Individual Scenario Servers run within each DRP instance. As a rule of thumb, individual Scenario Server instances are responsible for executing synchronous scenario behavior; while in contrast, Global Scenario Servers execute asynchronous scenario behavior.

With regard to high availability requirements, since Global Scenario Servers execute asynchronous scenario blocks, system users will be minimally impacted in the event of a GSS failure. Nonetheless, it is recommended that production sites run 2 or more Global Scenario Servers.

For more detailed information, see the *Configuring Dynamo Scenario Server: Configuring the Scenario Manager* section of the *Dynamo Personalization Programming Guide*.

Any single developer should be able to work on any piece of a project at any point in time, without interfering with any other developers. For an ATG environment, this means an individual installation of ATG, an appropriate set of individual database schemas, and a robust set of stubs against which integrations can be tested. Developers must have complete control of their environment, so that moving from one activity to another can be accomplished quickly and without having to involve the infrastructure management group.

Given the need for completely isolated development environments (from QA), the common solution is robust desktops for each developer. These should be setup with a common base of applications and operating systems, but the developers should be free to alter that base as they feel necessary to carry out their work. Given a deployment environment on a UNIX platform, the ideal development environment would be a PC running Linux. This would prevent the accidental introduction of any Windows-centric development, as well as allowing the developers to improve their UNIX skills. As an alternative that would be easier to manage with the current corporate IT infrastructure, emulation software (such as VMWare) that allows a full UNIX environment to be run on a Windows machine could be used. Regardless of operating system, a developer machine should have the following minimum system requirements: 2.0 GHz processor, 1 GB memory, and 60 GB hard drive. Databases can be provided on a central server, but it is probably easier to allow the developers to manage the database on their local machine, as long as they have a reference copy of the database with some sample data to revert to if necessary. ATG comes with its own "solid" database that is easy to use for development and testing purposes. ATG also provides an HTTP server "serverina" that is easy to use and test with ATG software.

Note that developers working on the same project are expected to thoroughly test their code in development. Generally this means the creation of unit tests that validate behavior to the extent possible, relying on stub implementations of integrations as necessary. The unit test should be reviewed by the project manager prior and tech lead to integration testing.

The following tasks are recommended:

- Verify the resource availability of personnel and equipment for the installation. An ATG Deployment Specialist should be scheduled to assist with the installation.

- Review the ATG Installation Requirements. Note: these may also be IBM or BEA requirements if the IBM or BEA Application Servers are to be used.
- Review the installation documentation to schedule the sequence and approach for the installation of each product.
- Perform the installation according to the installation documentation with the assistance of ATG.
- Verify the installation of all products completed successfully.
- Document your installation and client configurations for future reference.

Several source control systems are available to fulfill ATG customer needs. The simplest to implement is CVS, a free versioning system that has existed for decades. A disadvantage of CVS is that it lacks some of the management features of more sophisticated commercial offerings, but its basic GUI tools such as WinCVS and Eclipse integration may still be superior to pother options. Another alternative is Perforce, a relatively inexpensive commercial solution that is used internally by both ATG. Regardless of the final choice, one or more experts that are capable of making the source control system conform to the stated objectives will be necessary.

The staging environment is set up exactly like production, with full access to all services and identical data.

The performance environment needs to be a scale replica of production. Ideally, one quarter scale is best if that can be accommodated. It is critical that all pieces scale as closely as possible, including firewalls, third party services such as search, and most importantly the database machines. Experts in the appropriate areas should be consulted on the best way to achieve scaling for that system, since the objective is to produce a load equivalent to production on the reduced hardware scale.

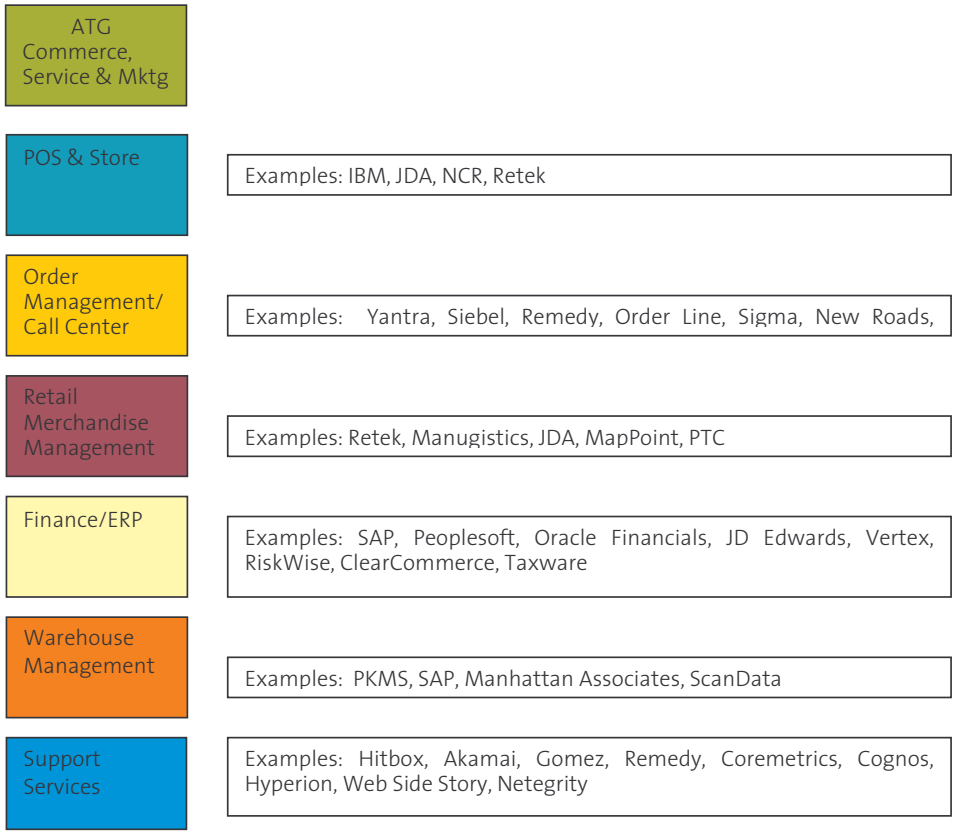
Dedicated hardware will also be necessary to generate the load for these tests, as it will be important for the test sources to not influence the results. The exact hardware specifications will depend on the testing tool chosen and the desired maximum load. Alternatively, if a regular release cycle can be established, performance testing can be contracted out, with the added benefit of having the testing occur over the internet, giving a more realistic picture of the end user effects.

This is a lot of hardware to leave idle much of the time. When considering the cost, remember that this is the environment intended to catch the unintended consequences of a development effort. Sometimes work will be done that doesn't hold up well under production traffic levels. Much of that problem can be avoided with proper project architecture, but this environment is designed to catch the parts that slip through and can cause serious production problems. If necessary, one way to reduce the hardware overhead is to share some of the QA hardware. However, in that case QA must be shut down during any performance tests, which should last a full day.

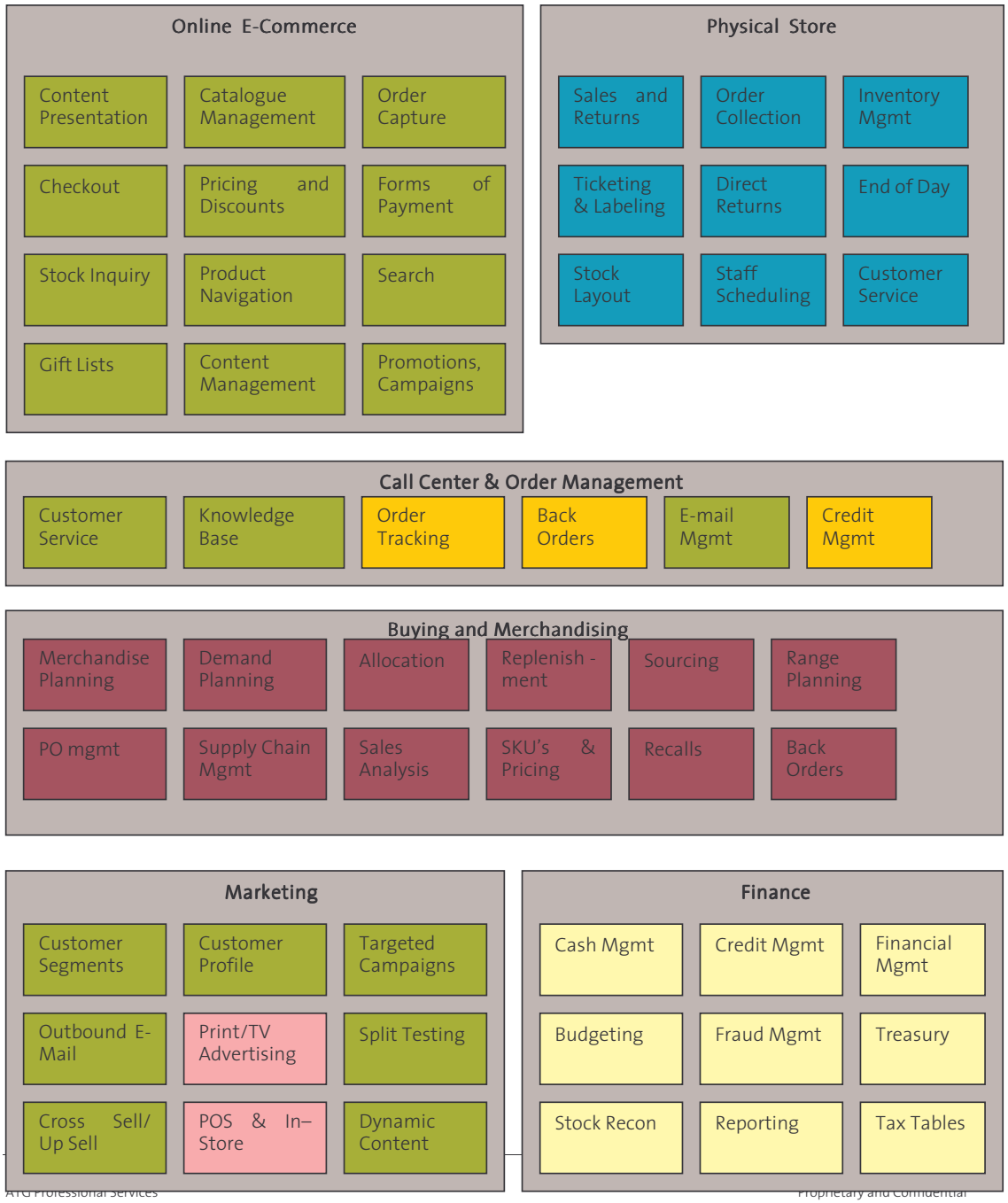
3.2 Phase I – Analysis Phase Overview

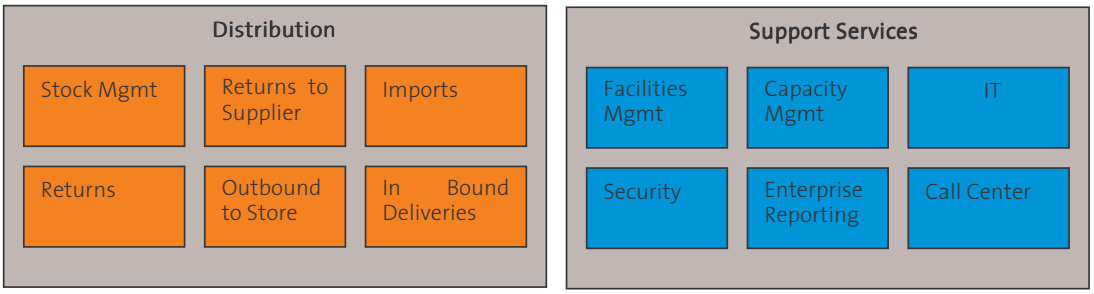
During the Analysis, you should gain an understanding of your current business practices and identify areas for improvements. The objective is to develop a comprehensive list of requirements for the implementation. During this phase, project team members gain a thorough understanding of the delivered software application. They gain this understanding by attending formal classes conducted by ATG and participating in extensive hands-on practice sessions. With the knowledge gained in training and the requirements defined, the project team prepares the Business Requirements & Solutions Document. The Business Requirements & Solutions Document maps the functionality of the software to the requirements of your organization.

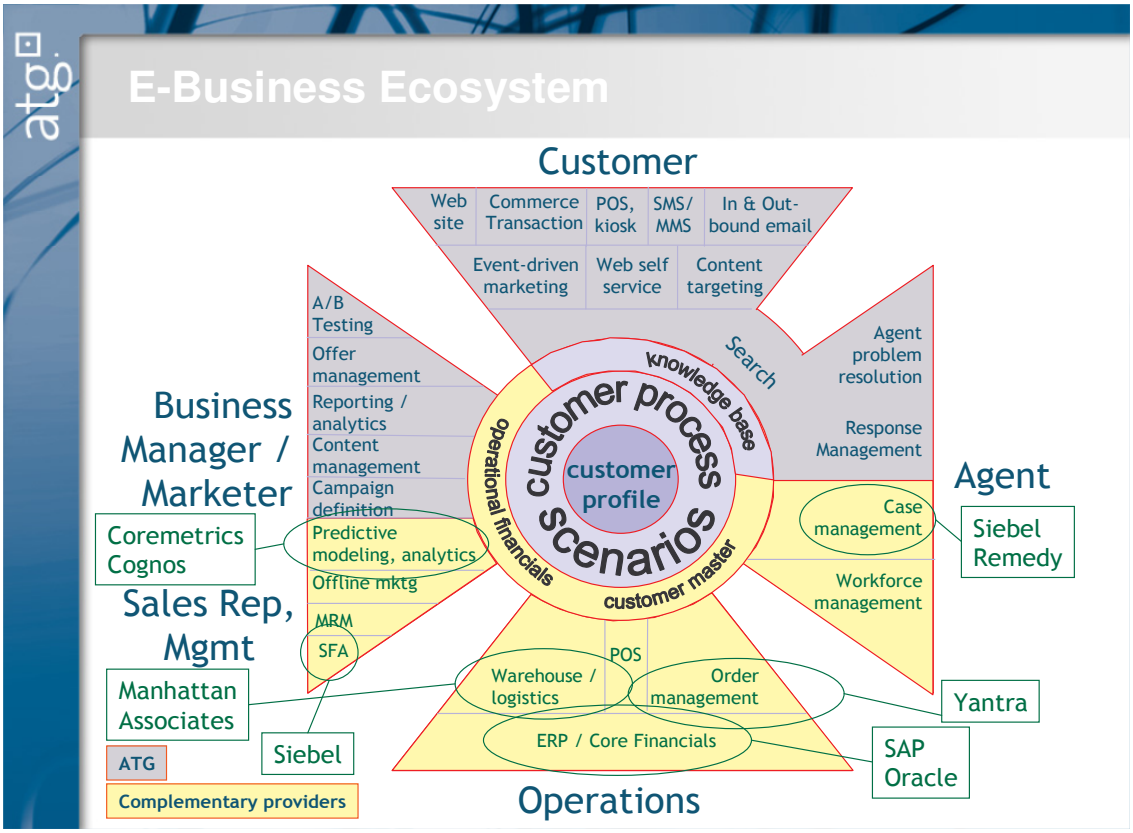
The project team members develop plans for conducting prototype sessions for each product and/or combination of products to be implemented. It is important to understand the full scale of the project including complimentary solutions that ATG will integrate with. The following is a high level business overview of a large scale retail cross channel implementation:



Multi Channel Retailer – Reference Business Architecture




Distribution
Stock Mgmt
Returns to Supplier
Imports
Returns
Outbound to Store
In Bound Deliveries
Support Services
Facilities Mgmt
Capacity Mgmt
IT
Security
Enterprise Reporting
Call Center


E-Business Ecosystem
Customer
Web site
Commerce Transaction
POS, kiosk
SMS/ MMS
In & Out-bound email
Event-driven marketing
Web self service
Content targeting
A/B Testing
Offer management
Reporting / analytics
Content management
Campaign definition
Predictive modeling, analytics
Offline mktg
MRM
SFA
Search
knowledge base
customer process scenarios
customer profile
operational financials
customer master
Agent problem resolution
Response Management
Case management
Workforce management
POS
Warehouse / logistics
Order management
ERP / Core Financials
Business Manager / Marketer
Coremetrics Cognos
Sales Rep, Mgmt
Manhattan Associates
Siebel
Agent
Siebel Remedy
Yantra
SAP Oracle
Operations
ATG
Complementary providers

The following is a sample software/server matrix that should be created by the customer:

Type	Name	Version	OS	Comment
Web Server	Apache	1.3.27	Solaris 9	
Application Server	Dynamo Application Server	6.3p2	Solaris 9	Includes Connection Module, Load Manager
Scenario Personalization	Dynamo Personalization Server, Dynamo Scenario Server	7.0p1	Solaris 9	
E-Commerce	Dynamo Commerce Server (B2C)	7.0p1	Solaris 9	
Fulfillment Integration	Dynamo Fulfillment module	7.0p1	Solaris 9	
Customized Dynamo Software	Dynamo Web Application	1.0	Solaris 9	This is the custom application
Catalog Administration	ATG Control Center	7.0	NT	
Basic Content Management	ATG Publishing	7.0p1	Solaris 9	
Tax Integration	Taxware	3.4.5	Solaris 9	
SQL Database	Oracle	9.2.0.6i	Solaris 9	
Messaging	Websphere MQ	5.3	Solaris 9	
Order Fulfillment	DMMS (NewRoads)	N/A	AS/400	

3.2.1 Document Preparation

Members of the project team create and present the completed flowcharts to the key people in the organization to determine the accuracy and completeness of the Operations Review. Revise the flowcharts to reflect any inconsistencies or suggested enhancements as a result of the walk-through. Distribute the final Operations Review document to the general user community for final approval. A Business Requirements & Solutions Document is a formal document, which identifies in detail the objectives of the project - what is to be achieved, and what functionality is required. It is important to include "wish list" items - functionality that is not currently available but is desirable. Again, conduct interviews with key application users and compile their requirements into a formal document. It is quite likely that the requirements have been previously defined in the Request for Proposal during the software evaluation phase. Distribute the completed Requirements Definition to interviewees and project team members for written approval of the content.

As part of the software selection process, some of this may have already been completed. It may be useful to obtain or review the original software requirement document created during the software evaluation process.

3.2.2 Requirements Definition

The objective of the Business Requirements Definition is to gain an understanding of the existing flow of information within a business process. This is achieved by interviewing key people in the organization and observing routine activities and exception processes. Develop flowcharts to diagrammatically represent the activities within your business processes. Obtain a thorough understanding of the existing environment, both automated and manual; in order to determine the potential impact and benefits of converting to the new software.

The most common frustration in online shopping, cited by 25% of respondents as among the 10 biggest problems, is being forced to register in order to make a purchase. Other problems and the percentage of respondents citing them as the 10 worst problems:

- Lack of ability to compare items with market research, 20%
- Inadequate or missing consumer ratings or reviews, 17%
- Difficulty in navigating category links to find particular items, 14%
- Inadequate expert ratings or reviews, 14%
- Too many site search results, 13%
- Lack of suggestions to improve site search, 13%
- Difficulty to compare items through site search, 13%
- Products not listed where expected, 13%
- Too few results in site search, 12%.

Topics covered during the Business Requirements include:

- Requirements Definition
- Finalization of goals and objectives
- Key system setup activities
- Key decisions required and assignment of ownership for resolution (Examples: chart of accounts structure, payroll processing cycles, issue resolution, etc.)
- Conversion concepts and interfaces to/from other systems
- Education plan for the project team

The major outcome of the analysis phase is a revised and more detailed product project plan. The project team identifies all known tasks, establishes milestones, estimates duration's of tasks, defines dependencies, and assigns resources.

Sample high level requirements for a Business to Consumer project are summarized below:

Repositories

Catalog Repository

Order Repository

Pricing & Promotions Repository

Profile Repository

Inventory Repository

Database

Database configuration

Populate Catalog and Price Data

Integration

Order Management

- Catalog import
- Web Methods
- Inventory
- Product customization
- Tandem
- Coremetrics
- Akamai
- Store Locator
- Search
- Custom Orders
- Rewards Program

UI Elements

- Site Header
- Site Footer
- General Navigation
- Home Page
- Category Detail Page
- Product Detail Pages
- Static Content Integration (help, contact us, etc..)

Core Commerce Functionality

- Catalog Navigation
- Catalog Search
- Design and Create Objects like Order, items, Shipping Groups, Payment Groups, etc
- Shopping Cart functionality i.e. Add items, remove items, etc
- Checkout Process
- Inventory Functionality

Pricing

- Fulfillment
- Payment
- Shipping
- Billing
- Tax
- Gift cards
- Wish list
- Order Status

Personalization

- Emails
- Registration/Login
- Promotion
- My Account

Other

- Kiosk
- Micro sites

It is important to remember that the plan is not "etched in stone." A well managed plan changes as the project progresses and tasks are added as they are identified.

The most common sought after commerce features retailers would most like to add or improve (source: e-tailing group):

1. Personalization
2. Improving search
3. Cross-sells/up-sells
4. Gifting features
5. Dynamic technology
6. Interactive tools
7. Multi-channel issues
8. Product enhancement (images, zoom, rotate, etc.)
9. Analytics
10. Live chat

It is of no surprise that there are many initiatives that could be completed for online retailers. In an effort to help prioritize based on impact vs. effort, creating a matrix of objectives may help your business to prioritize (see example below). Keep in mind, the suggestions may all be viable initiatives, the decision is more one of degree of impact measured against degree of effort.

Category	Activity	Impact	Effort
Increase conversions	Checkout Improvements	High	Medium
Increase conversions	Abandoned Order Tools	High – need more metrics on who, when & why	Medium
Increase conversions	Business Process Tracking	High – need to determine where customers disengage	Medium
Increase conversions	Campaign Optimizer	High – need to know more about which promotions are most	Medium

		effective	
Increase conversions	Redeem Gift cards, Points program Online	Medium – review the impact on margins	Medium
Increase conversions	Accept checks online	Medium	Medium
Ease of use/Increase conversions	Search Improvements	High	High
Ease of use/Increase conversions	Navigation and UI Improvements	High	Medium – performed incrementally
Industry Expertise/ Increase conversions	Contextual Selling	Medium	Medium – performed incrementally
Industry Expertise/ Increase conversions	Implement Customer Ratings & Reviews	Medium	High
Industry Expertise/ Increase conversions	Provide Top 10 Sellers	Low	Low (assuming data is available)
Marketing & Promotions	Define Target Markets and Customer Segmentation	High	Low – High depending on data available and programs to be deployed
Marketing & Promotions	Cross Sell Improvements	Medium	Medium – use in conjunction with Campaign Optimizer
Marketing & Promotions	Promotional Upselling	Medium	Medium – use in conjunction with Campaign Optimizer
Increase profit	Change Shipping Charges Policy	High – review the impact on conversions	Low
Multi- channel integration	Integration of abandoned carts to POS	High	High
Multi-channel integration	Text chat - Integration of web site to phone support	High	High
Multi-channel integration	Create a customer information hub – see all channel touch points	High	Medium – use in conjunction with knowledge management
Technical Stability & Operational	Implement a staging Environment	High	Medium

Improvement			
Technical Stability & Operational Improvement	Implement current version of the JDK	Medium	Medium
Technical Stability & Operational Improvement	Eliminate Downtime for catalogue refreshes	Medium	Low – implement catalogue swapping

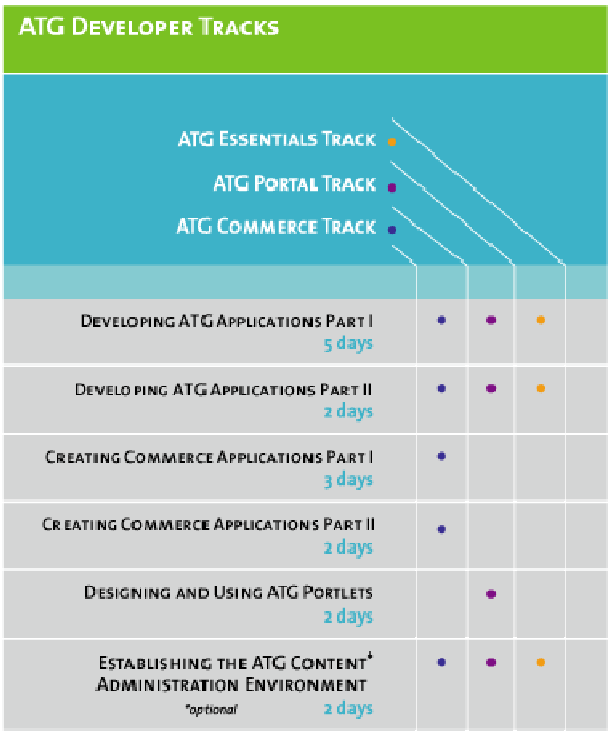
3.2.3 Project Team Training

atg Best Practice

This is the execution of the finalized education plan from Phase I of the implementation methodology. The intent is to gain an understanding of the functionality of the standard software. Core project team members only should be trained by this time. These will be individuals within your organization who have committed their time to the decision-making and setup of the software.

A complete listing of ATG classes can be found at www.atg.com. Our professional services and education team can discuss with you the proper training program that would be appropriate for your unique situation.

Below is a sample of ATG's developer track. Simpler tracks exist for Managers, System Administrators, and End Users.



Developing ATG Applications Part I provides you with the essential skills and experience you need to take advantage of the powerful features of ATG, including the value-added services that the Dynamo Application Framework (DAF) provides on top of the application server. You will learn how to access and manipulate visitor data and Web site content using ATG Data Anywhere Architecture, and utilize the customer relationship management features of ATG Scenario Personalization.

Developing ATG Applications Part II completes the education started with Part I, providing in-depth exploration of the data access, user tracking and management, and scenario customization features you need to start building applications using the ATG Adaptive Scenario Engine.

Creating Commerce Applications Part I explores the core skills that you need to create consumer and business commerce applications.

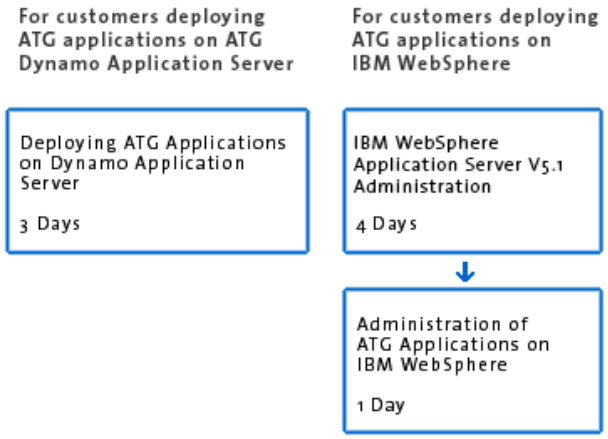
Creating Commerce Applications Part II is for developers looking to expand on their consumer and business commerce application development skills. You will learn about multiple catalogs, price lists, pricing models, invoicing, and more.

Exploring ATG Technology is for business managers seeking to learn how to maximize ATG technology to meet their business goals. It provides a business-focused overview of ATG products and technology and uses case studies to explore how ATG can be used to solve several common business problems.

Using ATG Scenarios is for business managers seeking to learn how to use scenarios in their ATG applications. Business managers will receive hands-on instruction on using ATG Scenario Personalization to target the right content and promotions to the right audience. The course teaches users how to create successful campaigns that will strengthen their relationships with customers, partners, and suppliers.

Modeling and Designing ATG Scenarios teaches business and technical managers how to facilitate the adoption of ATG scenarios by carefully articulating the business requirements and the design details in a way that leads to successful implementations. Students will walk away with a well-defined process for effectively translating business requirements into a detailed scenario model and an implementation-ready scenario design. Case studies for modeling and designing scenarios are used throughout.

ATG system administrator courses are designed to meet the needs of customers deploying ATG applications on various application servers. ATG offers two different course tracks based on the type of application server you are using.



Deploying ATG Applications on Dynamo Application Server introduces system administrators to the best methods for deploying, testing, and tuning an application built on top of the ATG Dynamo Application Server (DAS) for optimal performance, and how to maintain that performance

IBM WebSphere Application Server V5.1 Administration teaches administrators to install, configure, and maintain IBM WebSphere Application Server (WAS) Network Deployment V5 and to deploy enterprise Java applications in a single machine or clustered configuration.

Administration of ATG Applications on IBM WebSphere guides system administrators in the configuration, tuning and deployment of web applications built using ATG Adaptive Scenario Engine technology on the IBM WebSphere application server.

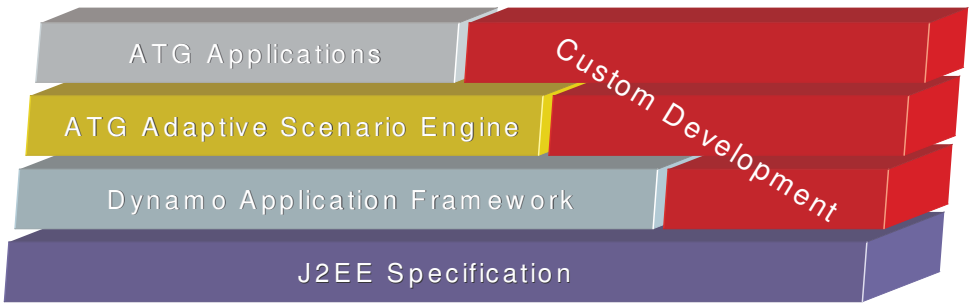
Project Team Training can take place at your location or at an ATG Regional Training Center. This is a critical step; the project team must be focused during training. If it is likely that your project team members will be interrupted by day to day activities during the training, we recommend off site training at a regional ATG training facility. If interruptions can be eliminated, on-site training provides the opportunity for setup discussions that are unique to your particular situation.

At a minimum, a primary and secondary person should be scheduled for each training class.

Three key ATG technology components that any ATG customer must become familiar with in order to properly understand and take advantage of the ATG platform are: the Nucleus, Form Handlers and the Dynamo Application Architecture (DAA).

- **Form handlers**, which you can use to evaluate the validity of form data before it is submitted, write data to and read data from a database or repository, and direct the user to different pages, depending on the results of the form submission.
- The **ATG Nucleus** is the Dynamo Application Framework's (DAF) component model for building applications from JavaBeans. DAF is a Java-based application development framework and programming model, primarily consisting of a component model, a data-access model, a message-based event-model, and a UI/form development model. Nucleus allows you to assemble applications through simple configuration files that specify what components are used by the application, what parameters are used to initialize those components, and how those components hook up to each other. The ATG Control Center Components window provides a handy way to create, modify, and manage Nucleus components.
- The **ATG Data Anywhere Architecture (DAA)** provides a unified view of content and data across a business for organizations and their customers. The core of the ATG Data Anywhere Architecture is the Repository API. Through the Repository API, you can employ a single approach to accessing disparate data types, including SQL databases, LDAP directories, content management systems, and file systems.

There is no substitute for properly training staff. Not taking the time to properly train staff on the core functionality of the product is a short cut mistake that no one company can afford to make.



3.3 Phase II - Design

3.3.1 Functional Specification

The objective of the Functional Specification is to map the features of the software to the requirements specified by the project team and the users. Where the software does not perform the requirement in the exact manner that the users may wish, ATG may suggest other means either through software customization or extension, interfaces, conversions or manual procedures to accomplish the requirement. Although the ATG products have been designed for general use in a variety of business settings, there may sometimes be a need for a work-around, or delay the request to a future phase, to allow the project to proceed on schedule. Identification and containment of "scope creep" is very important during this phase, and is often the first opportunity to address how to handle scope creep early in the project. The Functional Specification documents the recommended use of the software and identifies potential areas requiring customization or extension. Preparation of the Functional Specification highlights changes to be incorporated in the original detailed project plan. The Functional Specification document should be created at a level of detail that provides as much clarity regarding the specifics of the site to be built as possible. Appendix C of this document includes a sample table of contents for a functional specification of an ATG Commerce site. This TOC can be used by your organization to ensure you have considered all of the factors involved with creating an ATG Commerce web site.

The project team and the users review the Functional Specification to determine completeness and accuracy.

3.3.2 Catalog Design

The catalog is at the center of a commerce website and must be designed appropriately. The catalog implementation is designed with caching in mind and tries to minimize the amount of database I/O. With this in mind, designers of the catalog should understand:

- In most cases, the catalog should be read only. A read only design ensures that caching is used effectively and will minimize database I/O.
- The hierarchy is extremely flexible, but hierarchies and item relationships must be designed intelligently.
- The ACC may be used as a tool for catalog administration on staging and Q/A environments, but using it against a production catalog should be avoided due to the power of updating a production site directly.

Catalog Maintenance should use a combination of cycling the ATG instance to stop accepting new sessions, hot-caching catalog items, and reinitializing the instance.

ATG provides a sample catalog as a set of sample JSPs that constitute a small scale, but functional commerce site. As you develop your site, you can refer to the sample catalog for simple code examples that illustrate common ATG Commerce features, such as the following:

- Using dynamic pricing and inventory
- Navigating the product catalog
- Searching the product catalog
- Adding items to a shopping cart or gift list
- Checking out orders with a single shipping group and payment group
- Managing multiple shopping carts within one person's session

Additionally, you can use the sample catalog JSPs as a starting point for your own JSP templates.

The Catalog is the data representation of the collection of goods that can be purchased from the application. We provide a sample catalog schema that can be used, but this is determined by the customer: usually they modify what we give them, but sometimes the changes are very significant. ATG Commerce is flexible enough to accommodate most common catalog structures. The default schema has the notion of hierarchical categories, and specific items (corresponding to SKUs) in those categories.

The generic idea that is represented here is a collection of things with metadata that describe them. The metadata generally include product attributes (size, color, etc., though these things can also be represented as individual items). As a result, you can not only represent different kinds of hierarchies in the catalog, the individual items could represent other kinds of goods, whether digital goods or, perhaps, services.

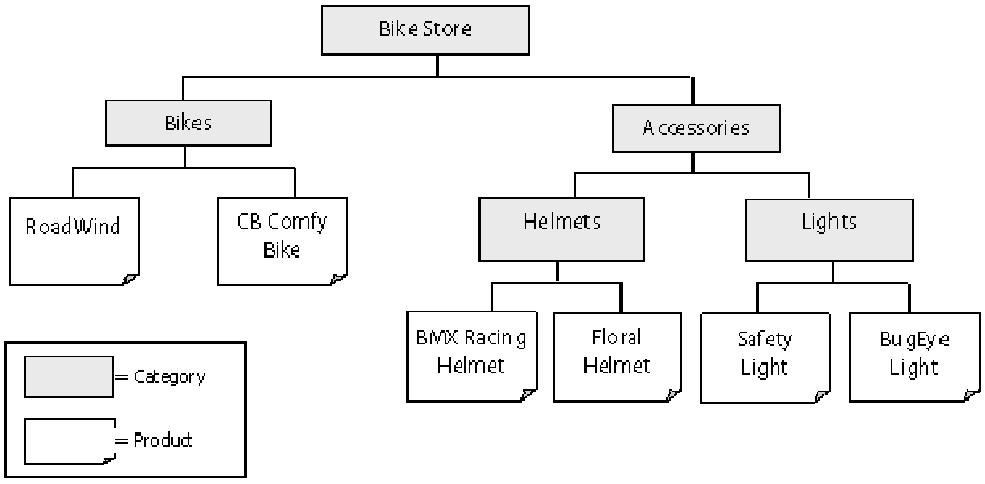
You can run the sample catalog with either ATG Consumer Commerce or ATG Business Commerce. If you're running ATG Consumer Commerce, the sample pages use a B2C-style user profile and a standard catalog. If you're running ATG Business Commerce, the sample pages use a B2B-style user profile and custom catalogs. This is achieved through the use of a ComponentExists servlet bean that determines which version of ATG Commerce is running and then renders the appropriate open parameter.

To run the sample catalog, when you start up ATG Business Commerce or ATG Consumer Commerce, simply append the DCSSampleCatalog module to the list of modules specified after the -m switch, as follows:

- For ATG Consumer Commerce: -m PioneerCyclingJSP DCSSampleCatalog
- For ATG Business Commerce: -m MotorpriseJSP DCSSampleCatalog

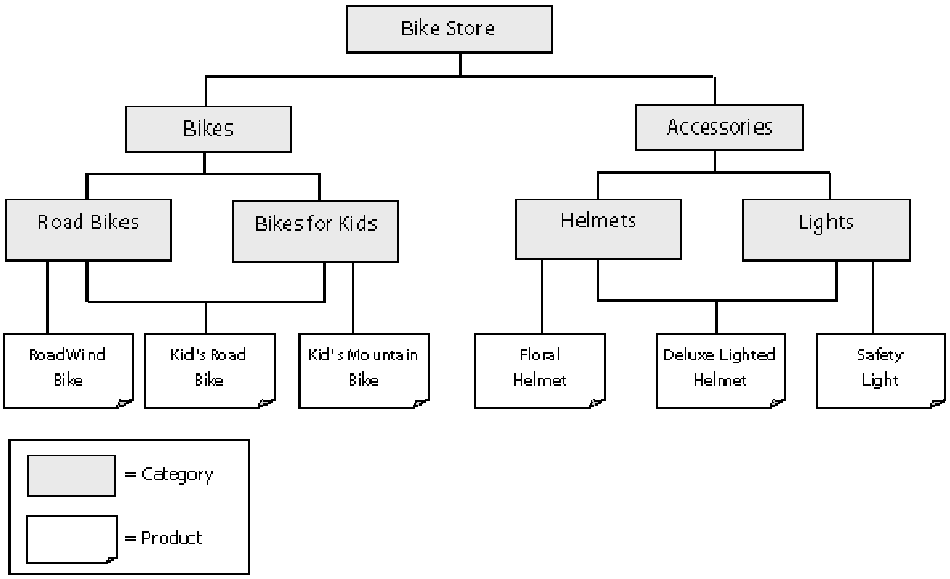
There are three organizational models you can use for your product catalog in ATG Commerce: single-path, multi-path, and custom catalog.

A single-path catalog contains items arranged in a tree-like structure. Each category contains products or other categories as its children, and each category or product has one parent category. The following diagram illustrates a product catalog in which each product and category has only one parent category.



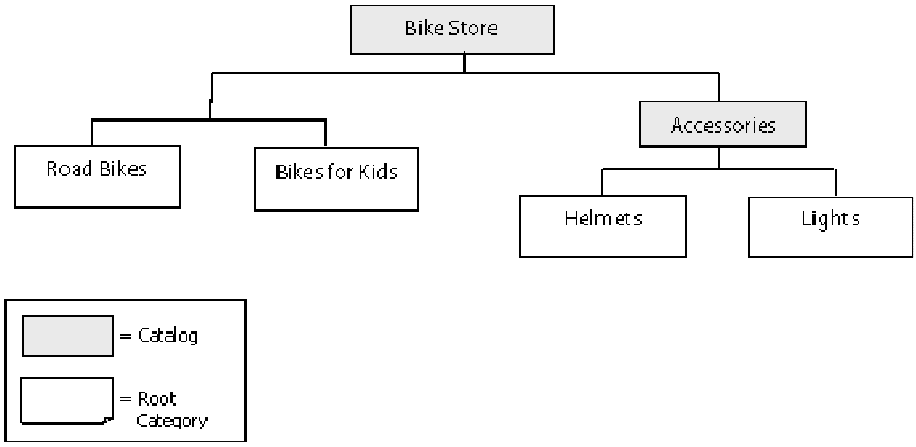
Note that there is a single navigational path to any given product. For example, to get to the BMX Racing Helmet product, the customer must select **Accessories**, then **Helmets**, and then **BMX Racing Helmet**.

A multi-path catalog offers a more flexible approach, and allows the customer multiple navigation paths to reach a given product. The following diagram illustrates a product catalog in which products have multiple parent categories.



In this example, the Kid's Road Bike and the Deluxe Lighted Helmet are accessible through multiple categories.

Custom catalogs are still more complex, in that entirely different catalogs can be presented to different users. If you have ATG Business Commerce installed, you can create custom catalogs immediately. If you have ATG Consumer Commerce installed and want to use custom catalogs, see Converting from Standard to Custom Catalogs in the ATG Commerce Administration and Development Guide. .



When a user with access to the Bike Store catalog views the root categories, he or she sees all of the root categories in the Accessories sub catalog as well as the root categories of the Bike Store catalog itself. A user who only has access to the Accessories catalog, however, sees only the Helmets and Lights root categories.

The Commerce product catalog supports all of these structures. Your catalog can have a rigid hierarchy in which each category or product has only one parent, a flexible structure in which a category or product can have multiple parents, or a still more flexible design in which different users see specially tailored catalogs.

3.3.3 Promotions and the Adaptive Scenario Engine (ASE)

atg® Best Practice

The tasks that a business user performs with ATG cover a range of planning and management activities related to setting up and maintaining your web site. You perform all these activities in the ATG Control Center. However, the tasks you carry out depend on the parts of the ATG product suite that you have installed.

Below is an overview of the steps you carry out to set up a fully personalized site with the Adaptive Scenario Engine (ASE).

1. **Set up visitor profiles**

At the foundation of ATG's personalization system are visitor profiles, a collection of properties that you use to store a range of data about everyone who visits your Web site. You use this data to tailor the content of your site so that, on subsequent visits, each person sees information that matches his or her interests and requirements.

The ATG system creates visitor profiles automatically, but there are some setup and maintenance tasks that you may need to perform to prepare them for the next step in the personalization process.

2. **Create profile groups**

The next step is to put your visitor profiles into groups that represent your Web site's target audiences. For example, you could set up groups based on income range, geographic location, or buying style. You use the profile groups to determine what content you show to whom - for example, you might advertise a specific product to one group and a different product to another.

3. **Set up content items**

Web site content is the text, images, and other items (for example, magazine articles or product descriptions) that your visitors see on your site's pages. Content is usually created by a site designer or page developer, and it is stored in content repositories. Like a visitor profile, each content item has a range of properties that defines it. You use these properties when you set up content groups, targeting rules, and tracking sensors for your site, and preparing these properties is one of the tasks you perform as part of setting up a personalized site.

4. **Group content items together**

In the same way that you group visitor profiles together, you must also identify similar content items and put them into groups. For example, if you based your profile groups on geographic location (one for U.S. visitors and another for European visitors), you would then create content groups that defined the content that was appropriate for each profile (articles on the New York stock exchange for the US group, articles on European markets for the European group).

5. **Match content with your target audiences**

Now that you have set up visitor profile groups and content groups, you can create business rules (called content targeters) that match the two together - in other words, business rules define the content that you show to each profile group. For example, you could write a rule telling the site to display items from the content group that includes articles about the US stock market to anyone in the profile group for site visitors from the US. In addition to varying content delivery according to a visitor's profile, you can also change it according to the date, the time of day, and other conditions. If you use the Scenarios module, you can create and use scenarios instead of targeting rules to control the content that you display to each profile group.

6. Track visitor behavior

An important part of the process of personalizing a Web site is updating visitor profiles with fresh data. This step allows you to consistently show visitors the content that is most relevant for them. For example, you probably want to track the site pages that each visitor displays so that you can determine the content that interests him or her the most. To do this, you set up tracking sensors that monitor your visitors' Web site behavior and activities and update their profiles accordingly. If your site use the Scenarios module, you can use scenarios instead of sensors to track visitor behavior and update their profiles.

7. Preview your Web site

The ATG Control Center provides preview features that you use to check how your Web site will look to visitors with certain profile properties under various time conditions. For example, you can see how a particular page in the Quincy Funds demo will appear to visitors who describe their investment strategy as "conservative".

8. Analyze data

Creating business charts allows you to capture and analyze a wide range of data related to your site visitors and their activities at your Web site. By analyzing this data, you can improve your visitors' experiences at your Web site, predicting their needs and responding by providing more relevant content. You can also use data from charts to fine-tune your scenarios and make changes to your Web site design.

You can set up promotions in the Pricing > Promotions area of the ATG Control Center. Promotions specify options that define the circumstances under which site visitors will be offered promotions. For example, you can specify the type of discount calculation to use for each promotion (fixed amount off the regular price or a percentage off the regular price) and the period of time for which it applies. You can also specify other options such as the number of times a visitor can use the same discount.

After setting up a promotion, you must set up a scenario that determines the visitors who qualify for the promotion. (For more information, see the *Creating Scenarios* chapter in the ATG Personalization Guide for Business Users.) The scenario tells the system how to determine the people that qualify, and then it marks their visitor profiles accordingly by adding the promotion to their activePromotions profile attribute. You can also set up global promotions that you give automatically to all customers - for these you do not need to set up a scenario.

When a customer visits a page that contains a product and its associated price, or performs some other action that involves requesting a price from the system, Dynamo checks his or her visitor profile and looks at the activePromotions attribute to see whether the customer currently qualifies to receive any of the promotions you have set up. It also checks to see whether you have set up any global promotions for all customers. Dynamo then uses those discounts to calculate the price of the product for the customer, and it adjusts the price accordingly.

You can inform site visitors about promotions in several ways. For example, you could set up a discount for a product without advertising it in any special way; the visitor simply sees the adjusted price when he or she displays the checkout page. Alternatively, you could include some text that describes the offer on, for example, the "Welcome!" page and make that text a link to the regular catalog page for the product. Or you could send an e-mail that describes the promotion, perhaps including a discount coupon code in the message.

The ATG Adaptive Scenario Engine (ASE), provides the foundation you need to create a more relevant and compelling customer experience - resulting in higher revenue, greater customer loyalty, and reduced operating costs. By helping you learn about, adapt to, and respond to your customers' ever changing needs and preferences, ASE automates the process of leading your customers toward a desired outcome (such as making a purchase, fulfilling a service need, or accessing information). A collection of technologies to enable web-based applications to learn and store user behavior, and in return for this user insight, use the knowledge to present personalized content and a compelling and relevant web experience.

ASE helps you:

- Improve marketing effectiveness, increase sales, and improve service across the Web, e-mail, mobile devices, and contact center
- Reduce operating and sales costs
- Improve customer satisfaction and loyalty

ASE includes an extensive palette of pre-defined building blocks that choreograph your customer interactions. To make your customer experience more relevant and compelling, ASE can gather, organize, and leverage existing information in your enterprise, using key capabilities:

Profile management and segmentation
ASE creates customer profiles based on a customizable and extendable set of attributes. It can update customer profiles continuously, based on every interaction or event that the user executes online, with a call center, or at a point of service. ASE also allows you to group customers into market segments and automatically move them into and out of different segments as their preferences or behavior change. The process of defining "personas" or customer segments has become very popular within the ATG community and has allowed ATG customers to more effectively target market customer segments.

The Adaptive Scenario Engine™ allows a business to specify the definitions of their customer segments using a set of rules. Within ASE, customer segments are known as 'Profile Groups', and the Profile Group editor provides a user interface for business managers to define the rules that determine these groups.

Segments can be defined based upon a number of constructs:

- **Profile Attributes** – Any attribute that is available in the Profile Repository can be used as an operand in the Profile Group editor, and a wide selection of operators are available to complete an expression, for example, "Gender = Male" or "Expiry Date <= 01/01/2006".

- **Roles** – A person's roles (or lack of role) within an organization can be used to as the basis of Profile Group inclusion.

- **Profile Group Membership** – Inclusion to a Profile Group can be referenced in the rules for building a new Profile Group. Thus, a hierarchy of Profile Groups can be created, where membership is inherited from high level Profile Groups.

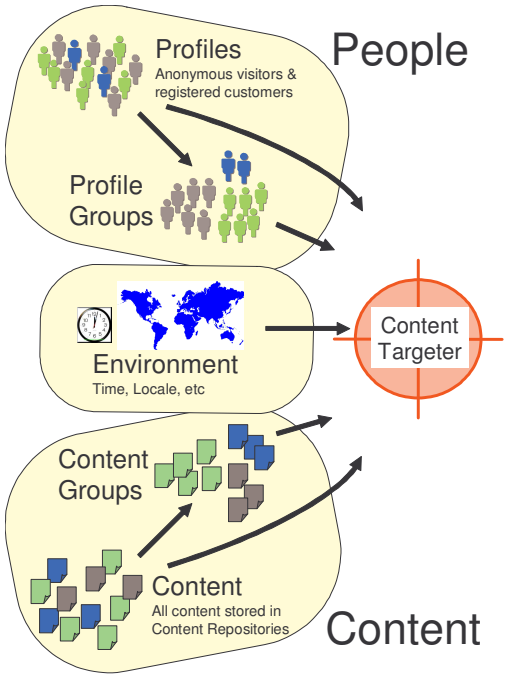
Having specified the various customer segments (aka Profile Groups) and the rules that define them, these segments can be referenced elsewhere in the ASE for the purposes of determining which segments see which web content. Additionally, application developers have access to the segment definition and can query the membership within their own application.

An important aspect of Profile Groups is that they are dynamic entities, meaning that the profiles that conform to the segment definitions are not determined on a one-time basis. Every time that ASE or an application queries a Profile Group for its membership, or queries a customer profile to see what segments they are in, the rules are executed to give the most up-to-date information regarding how customers relate to the defined segments. This means that segment 'lists' do not need to be maintained; ASE is doing the maintenance for you.

Content Targeters

Many view personalization as the mere ability of presenting a "Welcome back Sara!" as personalization, whereas others, perhaps in a portal context, believe that personalization is an ability to select and layout ones own portal page content. While there is some truth in these definitions, neither gets at the essence of what personalization is all about. Fundamentally, personalization should be a two-way street. A personalized web experience should certainly present a customer with a set of content and an experience that is relevant, interesting and timely to them, but personalization is also about what a business can learn about their customers as a result of doing so; it is an ongoing process of learning and refinement. Importantly, personalization should mostly happen in real-time, where software and rules are making the decisions on how to personalize the experience.

The ASE provides many tools to explicitly and implicitly 'learn' about customers. Explicit learning is characterized as the information that customer's willing let you know, such as their name, email address, billing address, etc. Implicit learning is characterized as the information that can be inferred through watching customer behavior (for example, the kinds of products they like to look at), or analyzing their transaction history (for example, the average size of their order). Both sets of information all help enrich the customer profile to the extent that, for many business, their web customer profile is the best source of 'hard' and 'soft' facts about their customers than any of their other systems.



Content Targeting is the practice of using the rich customer information and the categorization of content, and building rules that marry the two together in a way that something relevant and interesting will be seen by the customer.

Within the ASE, **Content Targeters** draw upon the customer profile information, the content (and meta data) and from environmental factors (such as the time of the year, the language setting in the customer's browser, etc). The execution of the targeting rule determines piece of content is placed into the Targeter, and then ultimately presented on a Web page.

For example, a 'Todays Interesting News Articles' Targeter may draw information from a Content Repository that contains news articles, and based upon the categorization of the news articles and the interests and preferences of the Web visitor, an appropriate set of new articles can be displayed.

Likewise, a 'Product Recommendation' Targeter may present a customer with the most compelling product of the month based upon the available products and the customer's past order history.

Content Targeting rules enable web designers to define the circumstances in which certain content will (or will not) be displayed to certain people. These rules are defined through an intuitive, easy-to-understand interface.

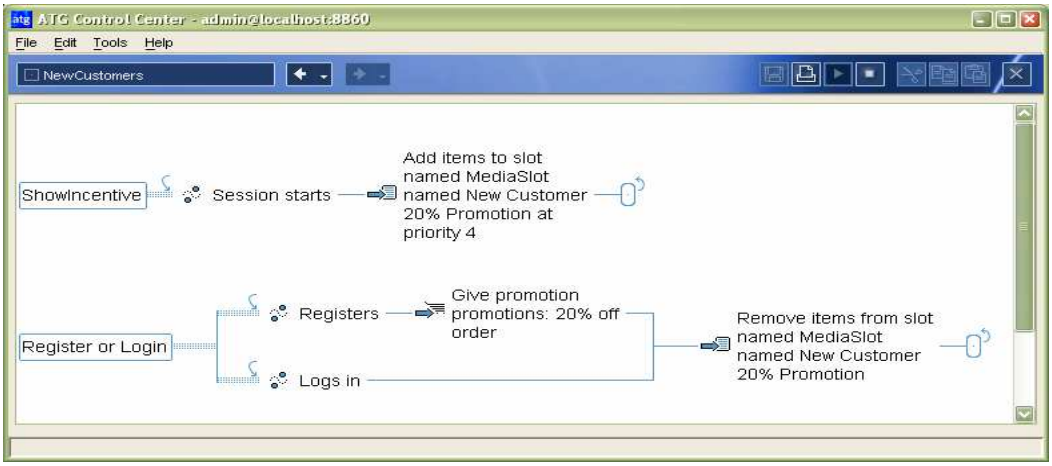
A Content Targeter can store more than one piece of content, since they are analogous to a deck of cards. The Web developer who uses the Content Targeter within their web page can choose to pick from the top of the deck, display the whole deck, a range within the deck, or randomly pick a card from the deck. The dynamic nature of Content Targeters means that as the rules may change which determines the contents of the Targeter, no code changes are required for the new content to take effect.

Content Targeters are an important concept in how the behavior of a Web site and the customer experience can be de-coupled from the page and web application code that is executing. They provide the business with declarative control over their site, rather than having to draw upon costly development resources every time a page or site needs refining. They enable a Web site to become considerably more 'data-driven', and within the context of their Web visitors and the content available to the site.

Scenario Personalization

In the same way that any journey has things that happen on the way, decisions to be made, and corresponding actions to take, Scenarios enable business to map out the likely journeys (and possible detours) that a customer may take to achieve their objective in dealing with a business. Scenarios are customer processes that are triggered by a significant event (an order placed, a bill paid, a click-through from an email), and then continue to layout the sequence of subsequent actions, decision points and subsequent events to watch for in order to guide a customer towards their objective, whilst ensuring the business is receiving value as well. The execution of the Scenario may be directly obvious to the customer, or could be invisible, yet is making some change or notification within the business, such altering an aspect of the customer profile or triggering some internal business process.

For example, consider this 'New Customers' scenario. It has the basic task of tempting new customer's to register with a site by offering them a 20% promotional discount. The first part of the Scenario ensures that as every new Web session starts, the 20% offer is appropriately displayed on the page via a 'Slot' (see the UI section of this guide). The second part of the Scenario branches, and waits on one of two things to happen; either the customer 'registers' or they 'log in'. If the visitor registers, the scenario grants the visitor with the promotional offer, whereas if they log in, the offer is not granted (since they are clearly already a customer). Finally, the promotion is removed from the page so that the visitor no longer sees it.



This basic example demonstrates many of the interesting features of Scenario Personalization. Firstly, it provides the sense of 'waiting for things to happen' and then 'doing something'. Secondly, it provides a minute glimpse at the numerous out-of-the-box 'events' and 'actions' that are provided with ASE. But lastly, and most importantly, the Scenario has graphically implemented an important piece of functionality on the Web site without a line of code being written!

Scenario Personalization lets you easily drive pre-designed dialogues that automatically adapt as customers' circumstances change. For example, a retail banking customer may update his or her online profile to reflect being newly married or having had a baby. That new information can trigger tailored promotions for joint banking services, life insurance options, and home-improvement loans.

Scenarios, available exclusively from ATG, personalize every single process or user dialog over the Web, e-mail, or mobile devices according to a user's interests, segment, market, and history. An easy-to-use interface gives your business and technical staff the ability to work together to create and maintain a personalized user experience, easing communications and reducing development time.

Scenario Personalization supports Web Services so you can leverage scenario technology from any remote system. You can integrate scenarios across both online and offline channels and leverage existing personalization rules across channels to provide personalized content and slots. Scenario Personalization allows you to provide consistent, personalized service to users in every situation, resulting in measurably increased usage and sales.

What has not been mentioned is that there is, in fact, no limit to the events, actions and conditions that may be created. This is because at its core, Scenarios are simply tapping-in to the extensive use of the JMS-based messaging system through which different ATG system components communicate. As a result of this, any JMS message that is "listened for" within the system can be surfaced in the Scenario UI as an 'event', and any JMS messages generated by the Scenario Engine can be surfaced in the UI as an 'action'. As such, a business can easily extend the set of Scenario events and actions to suit their own needs.

The underlying JMS infrastructure means that these custom events and actions are not limited to messages that are created and within the ASE. Any external application that is able to generate or receive a JMS message can interact with the Scenario Engine. Furthermore, since the Dynamo Application Framework makes it very easy for a JMS message to enter the system through SOAP, the requirements for any external system that want to interact with the Scenario Engine is merely a need to support Web Services.

How are scenarios different than targeters?

Several features in the Scenarios module are also available in a different form in the Personalization module. For example, you can set up targeters in the Personalization module that respond to user actions by displaying personalized content on a page or by sending e-mail.

The Scenarios module, however, introduces an **element of time** to the way you interact with visitors and personalize your site's content. Scenarios can contain elements that tell the system to wait for a specified period before continuing with the next step; for example, you can have the system identify new members, and then wait three months before sending out a follow-up message. It is primarily this feature that turns a collection of unconnected personalization and targeting activities into a campaign and allows you to choreograph (and therefore better control) the relationship between your visitors and your Web site.

The following guidelines below suggest when to use scenarios and when to use targeters.

- Scenarios are triggered by events. If you want to personalize content as a result of a site visitor's doing something specific (for example, logging in), use a scenario. Targeters are not event based.
- The business rules you can set up through targeters are more flexible in some cases than scenarios. For this reason, if you need to set up highly complex rules to match content to visitors, use targeters.
- Scenarios use a feature called slots to display dynamic content. Slots provide some powerful advantages over targeters (for example, you can set up empty slots that generate their own requests for content).
- The reporting features of the Scenarios module rely on scenarios to supply data. If you want to use reports to track and analyze business information, you must set up appropriate scenarios.
- As described above, scenarios are time aware. If you want to set up a long-term approach to content and visitor management, use scenarios.

Scenarios and targeters are, however, complementary features. You can use a combination of targeters and scenarios to achieve the results you want.

3.3.4 Shopping Cart

Order processing begins when the user places a product in the shopping cart, and an ATG order is created in the ATG repository. As soon as the user progresses in the checkout process, the system places several calls to the Order Management system to obtain the additional information required to complete the order. All the information returned from the Order Management system can be overridden by the ATG front end. At the end of the order creation process, there will be an ATG order in the ATG repository and if there was no contingency in the Order Management system.

If, for some reason, the Order Management system cannot be reached or the call times out, the order is marked as "contingent" in ATG, and should be synchronized with the Order Management system via a scheduled batch process.

In ATG Commerce, a shopping cart is an order in an incomplete state. The shopping cart stores the information about the items a given customer wants to order and their associated quantities and prices. In addition, it stores the shipping and payment information for the order. The ShoppingCart component stores a user's current and saved shopping carts. You can use the ShoppingCart component's properties and handle methods to create a new shopping cart or retrieve one of the user's saved shopping carts and make it the user's current shopping cart. The ShoppingCart.savedEmpty property is checked to determine whether the current user has any saved shopping carts. If the user doesn't have any saved shopping carts, the user is given the option to create one. If the user has saved shopping carts, the user is given the option to select one of the saved shopping carts to either delete or make the current shopping cart, to delete all of the saved shopping carts, or to create a new shopping cart.

As mentioned, the Shopping Cart is nothing more than an order being assembled for processing, and, in fact, within the Commerce application that is exactly what it is called: the Order. The Order is simply the collection that is used to aggregate the basket of goods that might be purchased, on which the application may perform useful operations, such as checking inventory, pricing, calculating shipping, or identifying cross-sell and up-sell opportunities. In fact, other collections of goods, such as gift lists and recurring orders are essentially saved orders that might be run through the Purchase Process at a later date.

The generic idea is simply the collection of goods to be processed -- where processing could in the long run be scheduling with a services scheduler, or the enablement of entitlements on a soft-goods server.

You can create pages that allow users to add multiple items to the current shopping cart in a single form submission. The items can refer to different products, different SKUs, and have different quantities. The CartModifierFormHandler contains an items property that allows you to set per-item property values

By default, ATG Commerce supports three types of commerce items. One corresponds to regular SKUs. The other two correspond to configurable SKUs and their subproperties (see Creating a Configurable SKU in the *ATG Commerce Catalog Administration* chapter). Your site may support additional custom commerce item types (see *Extending the Purchase Process* in the *Customizing the Purchase Process Externals* chapter in the ATG 7 Commerce Programming Guide).

Removing Items from an unusable cart for a returning user is based on criteria defined by business rules. Many Commerce sites retrieve persisted carts when anonymous or registered user revisits the site. If the user is returning after a long period then the cart might be unusable for various reasons, for example the products from the cart are discontinued, the site has a new catalog every season, the cart is older then x days etc.

ATG Commerce executes various commerce pipelines based on the operation being performed, for example loadOrder, saveOrder, commitOrder. The pipeline consists of pipeline processors, each performing a subset of functionality of the entire chain.

In order to cleanup an unusable cart, add a custom processor RemoveItemsFromUnusableCartProcessor, which is added to the loadOrder chain towards the beginning of the chain as defined by business rules.

This processor inspects the order object against business rules to determine if the cart needs to be cleaned up and cleans up corresponding CommerceItem and the relationship objects CommerceItemShippingItemRelationship, CommerceItemPaymentGroupRelationship) from the Order object.

Use the OrderLookup servlet bean to retrieve a user's incomplete orders (that is, shopping carts). OrderLookup enables you to retrieve a single order, all orders assigned to a particular cost center (ATG Business Commerce only), all orders placed by a particular user, or all orders placed by a particular user that are in specific state, such as incomplete.

3.3.5 Checkout

The checkout process collects information such as shipping information and payment information. Many sites implement two different types of checkout processes: basic and advanced.

The basic process allows a user to perform the following functions:

- Enter a single address that all purchased goods can be sent to
- Enter a single credit card number as a payment method

The advanced checkout process adds to this functionality by allowing:

- The ability to ship to multiple locations
- The ability to pay with multiple payment methods

When customers enter credit card information in the checkout process or on the My Profile page, they must be transferred from a non-secure server to a secure server. In order to configure this, you must change the properties of the ProtocolChange component. You can edit these properties in the Pages and Components> Components by path section of the ACC. The component pathname is /atg/dynamo/droplet/ProtocolChange.

Set the values of the secure properties appropriate to your server. Note that the securePort (the port setup for secure server) need not equal 443.

Set the enable property to true.

The ATG Order Manager is a collection of "pipelines" that are used to process Orders(e.g., manipulating, loading or saving Orders) -- they specify the actions that should be taken on an Order in, for example, a checkout process. This might involve pricing, calculating shipping, etc. The Order Managers in ATG are all configured in ways useful to hard goods, other kinds of goods might require other kinds of managers.

The Checkout process varies depending on what type of user is placing the order:

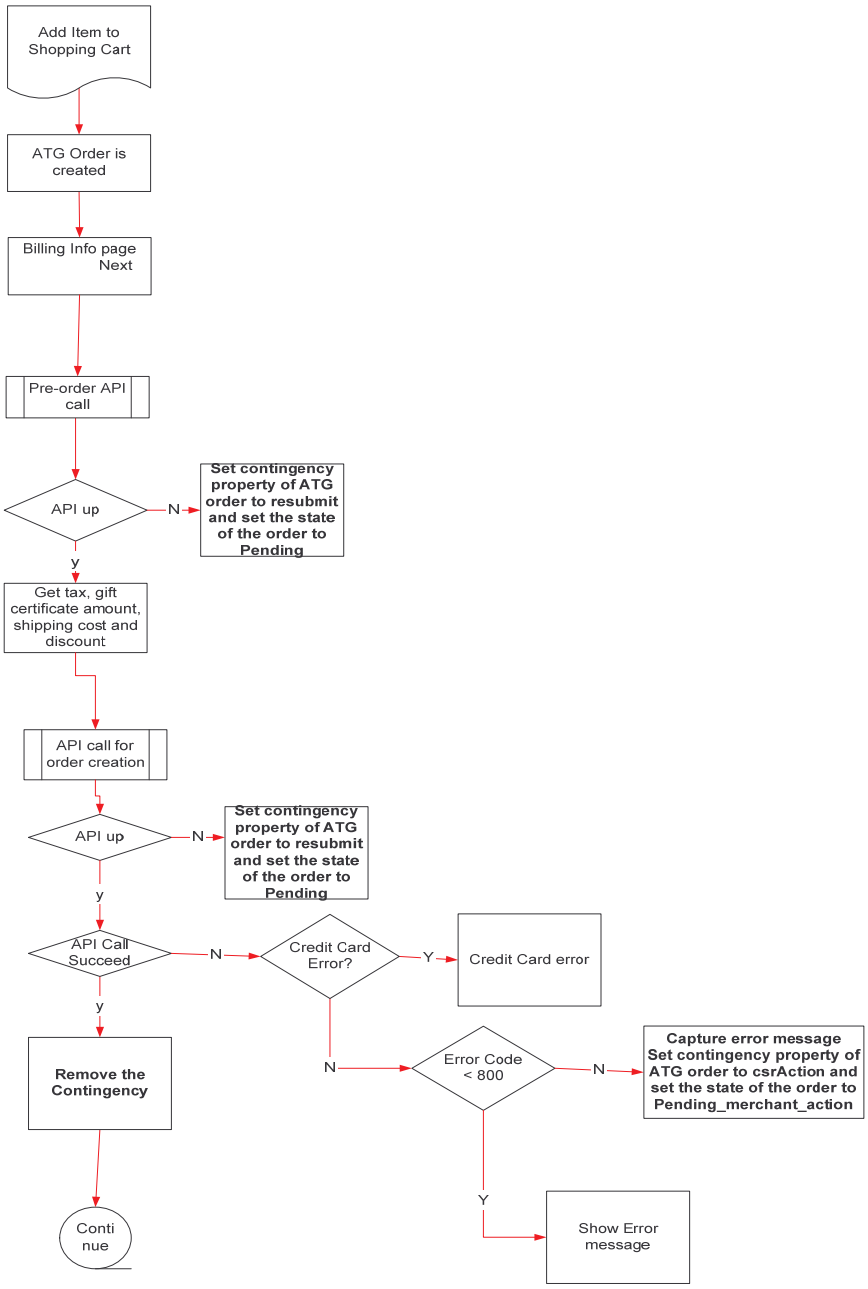
- Registered users who have credit card information on file are taken directly to the Order Confirmation page with all the required order information already populated with the information stored in their user profile.
- Registered users who do not have credit card information stored are required to enter this information in the Check Out page.
- Anonymous users are required to enter shipping information, billing information, and credit card information on the Check Out page.

The Order Confirmation page allows the user to validate all the pertinent order information, including coupons or promotions, gift bags, etc. and ultimately submit the order for processing.

An important feature of the Checkout process is to perform on-demand inventory checks during the process: 1.) the start of Checkout, 2.) when shipping charges and tax are applied to the order, and 3.) at the very end of the order. This will ensure that inventory has not been re-allocated during the Checkout process.

The check process performance needs to be fast. This is critical since during this process the system is most likely interacting with back end systems. Lack of acceptable system performance (response time) during check out means lost customers!

During the checkout process whenever the API is down to the order management system the order should be marked as a contingency order and is subject to resubmit. Contingencies should be a property in the order repository and the ATG orders that are created through the checkout flow will have the state of Pending and contingency property that will show the nature of contingency. The flow diagram below defines the process flow for order management contingency processing:


Add Item to Shopping Cart
ATG Order is created
Billing Info page Next
Pre-order API call
API up
N
Set contingency property of ATG order to resubmit and set the state of the order to Pending
y
Get tax, gift certificate amount, shipping cost and discount
API call for order creation
API up
N
Set contingency property of ATG order to resubmit and set the state of the order to Pending
y
API Call Succeed
N
Credit Card Error?
Y
Credit Card error
y
Remove the Contingency
N
Error Code < 800
N
Capture error message Set contingency property of ATG order to csrAction and set the state of the order to Pending_merchant_action
Y
Show Error message
Conti nue

ATG Commerce offers the following two alternative component sets for the implementation of the Checkout functionality.

- Multiple request scoped checkout formhandler
- Single session scoped shopping cart formhandler

This section describes how to make a selection between the two sets.

ATG Commerce offers duplicate functionality components for implementing checkout functionality as listed below:

- Session scoped form handler.
  - **atg.commerce.order.FullShoppingCartFormHandler**
- Request scoped multiple form handlers.
  - **atg.commerce.order.purchase. CartModifierFormHandler**
  - **atg.commerce.order.purchase.ShippingGroupFormHandler**
  - **atg.commerce.order.purchase.BillingGroupFormHandler**
  - **atg.commerce.order.purchase.CommitOrderFormHandler**

We have identified and listed various evaluation criteria that will aid your development team and help make an informed decision.

Alternatives/ Trade-off/ Recommended Solution:

Early Commerce site implementations offered limited options to end-users during the checkout process, e.g. create single shipping address and create single payment method. To provide basic checkout process functionality ATG Commerce designed a single session scoped component called ShoppingCartFormHandler. This component provided shipping, payment, order review and order submission functionality.

As Commerce sites evolved, more complex business rules were defined and more options were offered to end-users, e.g. multiple shipping address, address selection or creation, multiple payment methods and payment method selection or creation. To enable advanced check out process requirements ATG Commerce product evolved to offer two alternatives:

1. Added additional functionality to ShoppingCartFormHandler
2. Created smaller, request-scoped form handlers to deal with each aspect of the checkout process for e.g. Shipping, Billing , Order review and Order submission.

The additional functionality resulted in bloated ShoppingCartFormHandler implementation. Some of the subsequent issues noticed in sites using the multi-purpose and monolithic ShoppingCartFormHandler are listed below:

- Session-scoped form handlers have proven to be ineffective at various commerce implementation sites.

  Session-scoped form handlers store state based on user's interaction in current session. Typically during a "happy navigation scenario", session-scoped form handlers become handy as less development effort is required to maintain information across pages. Checkout being a complex aspect of the site, and many users go through "unhappy navigation scenario", which leaves partial state and error messages stored on form handlers. In no time, it becomes a development challenge to clear partial state and error messages for all possible "unhappy navigation scenarios".

- Large Memory footprint per user

  Session-scoped form handlers store state based on the user's interaction in current session. ShoppingCartFormHandler stores data about multiple shipping locations, address book from profile, credit card profile from profile, multiple payment methods and etc. For a complex commerce site with many concurrent users on the site, memory requirements per DRP increase considerably with session-scoped form handler. To keep memory footprints low , request-scope form handlers stores data on to the order object or as hidden fields across pages.

- Development and Maintenance complexity due to Monolithic ShoppingCartFormHandler

Since ShoppingCartFormHandler provides multi-purpose functionality, many concurrent development and maintenance tasks would require developers to modify a single component. This makes concurrent development and maintenance difficult during project life cycle.

- ShoppingCartFormHandler is going to be depreciated in future versions of ATG commerce.

  ATG Commerce introduced small and easy to use request-scope form handlers as replacement alternatives to monolithic and complex single form handler. Current versions of ATG Commerce supports both alternatives, but there have been a thoughts of deprecating ShoppingCartFormHandler over time.

For a complex Commerce implementation with a sufficient development timeline, ATG recommends the use of multiple request-scoped form handlers that are available in ATG commerce.order.purchase package.

For a simple Commerce implementation with tight deadlines, ATG recommends simple development using a session-scoped single form handler that should be replaced with multiple form handlers when project timelines permits.

3.3.6 Pricing

The pricing engine is extremely flexible and highly customizable. Some of the best practices in regards to pricing are detailed here:

- Pricing information need not be shown during casual catalog browsing (or just a default price used if desired). On the item detail page or when a user shows interest in an item then real-time pricing information should be calculated.
- Use price lists appropriately instead of over saturating the use of promotions.
- Design price lists in a hierarchical fashion that makes best use of the existing architecture.
- Use the existing architecture and extend it as necessary. It is not desirable to replace the pricing architecture. This will ensure that promotions and other out of the box features work seamlessly with your data.

CartModifierFormHandler automatically reprices a shopping cart when it is used to add items to the cart. (Note that it also reprices the cart when it is used to *remove* items from the cart.) However, you'll need to reprice shopping carts via some other mechanism if customers can make changes that affect order price through other form handlers that do not reprice shopping carts (for example, by making shipping changes via the form handlers that create and manage shipping groups), or if the shopping carts are modified through some other means in ways that affect order price, such as the delivery of a promotion via a scenario.

If your site has any pages where you need to reprice a shopping cart, but you cannot do so through a form action and corresponding handle method, use the RepriceOrderDroplet servlet bean. In fact, you can use the RepriceOrderDroplet servlet bean to reprice a customer's shopping cart every time the customer accesses a shopping cart page. This ensures that the customer always views accurate pricing information as he or she makes changes to cart.

3.3.7 Shipping

ATG provides flexible methods to calculate an estimated shipping cost from available shipping methods.

In order to aid a user to compare available shipping options, many Commerce sites inform users of the available shipping methods, estimated delivery duration and estimated shipping cost. This information needs to be refreshed when the user is interacting with the UI to decide which item should go to what location.

Commerce sites shipping cost calculations are complex in nature and usually have multiple dimensions such as:

- Multiple shipping methods
- Shipping method based on address
- Domestic, APO/FPO, International shipping

- Multiple ship-to per order
- Split single commerce item to multiple ship-to
- Special shipping charge for xyz product
- Shipping promotions needs to be applied

Shipping Address of the user is captured in the Checkout. There should be appropriate sub-navigation bars available for checkout process. Shipping restriction validations need to be completed that compare shipping address and items in the shopping basket to ensure they are acceptable.

Adding shipping information to shopping carts involves the following sub processes:

- Creating a list of shipping groups for potential use in the current order. The user can select from among these shipping groups when checking out the order.
- Specifying the shipping groups to use with the current order.
- Selecting the shipping methods, such as Ground or Next Day, for the order's shipping groups.

Use the atg/commerce/pricing/AvailableShippingMethods servlet bean to provide the user with a list of available shipping methods (such as Ground or NextDay) for a particular shipping group. Given a shipping group, AvailableShippingMethods queries the ShippingPricingEngine and returns a list of the shipping methods available for the type of the given shipping group.

Most of the time shipping cost calculations need to be performed at the back-end due to complex business rule and requiring database access for shipping charges.

Most shipping cost calculators have exhibited following limitations:

- Association between the Commerce Item and Shipping Address should have been established, which is done after the shipping screen.
- The calculator can calculate the cost of a selected shipping method but not all the available options.

The following extention can be applied to handle such situations:

- Add a shipping cost calculation utility class to encapsulate complex shipping calculation logic.
- Add an estimated shipping price droplet which shares the shipping cost calculation utility with OOTB Shipping calculators.

The following data fields are required for shipping:

Data Element	Description
Type of Address	Refers to the Country/ Region – whether US/ CA/ International or military
First Name	First Name of the person the order is shipped to
Last Name	Last Name of the person the order is shipped to
Street 1	Ship to Street 1address
Street 2	Ship to Street 2 address
City	Ship to City

State/ Province	Ship to State
Zip/ Postal Code	Ship to Zip/ Postal Code
Country	Ship to Country
Wish List Address List	Shipping Address of the person from whose wish list items have been bought
Address Book List	Stored Shipping Addresses in the Address Book by Registered Users

3.3.8 Order Management Integration

Many enterprises prefer to leverage their investment in their Order Management System (OMS) and use ATG commerce platform for capturing orders from the web. As part of the order placement process, the order needs to be submitted to the OMS and the order number and tracking information needs to be displayed to the user in a synchronous call.

Many enterprises have made significant financial, technical and infrastructure investments in their OMS. Order Management offers features like real-time inventory check, order submission, order fulfillment, order tracking, order cancellation and etc. Most order management systems do not accept orders directly from the web, and that is where customers use the ATG commerce platform for capturing the order and submitting it to order management as the last step.

The Order Life Cycle business process flow represents all business processes involved in the Order flow, from the time a Web User adds a Catalog Item to the Shopping Cart, until the Order is sent to Fulfillment. The Order Life Cycle incorporates the following System processes:

- Inventory availability checks (ATG)
- Web Inventory allocation (ATG)
- Unplanned Backorder allocation (ATG)
- Inventory Updates (Fulfillment)
- Billing Address Verification (Taxware)
- Shipping Address Verification (Taxware)
- Tax Calculation (Taxware)
- Payment Processing (store cards and credit card), including Split Tender Processing
- Fraud Checks
- Email notifications

The following table shows a sample Order System Status associated with the different steps of the Order Flow.

System Status	Description
Incomplete	Order was initiated but not completed (the User started the CheckOut process but never completed it by clicking "Place Order") - This is the pre-CheckOut state of the Order.
OrderPlaced	User has submitted the Order after successfully completing the Check Out process – User clicked on the "Place Order" button, placed the Order and received a Thank You page or Order Confirmation message.

In_Fraud	Order was flagged as potentially fraudulent. The System will provide additional information about the fraud flag that triggered the state. Order can be flagged as In_Fraud either before or after Payment Authorization.
In_Suspense	Order was placed in suspense for one of the following reasons: • Multiple gift cards need to be merged • Incomplete Order – missing fields • Address Issues • Invalid or Expired Promo or Money Card • Store Referral Problem • Unable to calculate tax • Duplicate Orders • Missing SKU
Auth_Fail	Payment authorization on the Credit Card failed or there was a problem with the Store Card, Balance Check or Debit on one of the Payment Groups.
Submitted	Order was submitted to the OMS and waiting response.
In_Process	Order was submitted and accepted by the OMS.
Failed	Order was submitted to OMS and was not accepted by OMS (bad XML, invalid SKU).
Backorder	Entire order has been placed in ATG BackOrder queue.
Partial_Ship	One or more order lines have shipped.
Shipped	Entire order has shipped by OMS, but payment has not yet been has not been captured.
Completed	Order Closed. No more Order or Order Line activity is pending.
Cancelled	Entire Order was cancelled.

Since ATG and OMS are transactional in nature, design order placement as an ATG transaction. The information needed on thankYou.jsp is generated by the ATG application for information such as the order number or the shipment tracking information. ATG can send out JMS messages asynchronously to OMS and then the OMS takes over the control for the order.

Many times OMS offers order tracking for its orders and are integrated with shipping systems to provide shipping tracking functionality to end users. This requires a synchronous call from ATG to the OMS as part of the last step. This information is subsequently displayed by ATG by the thankYou.jsp.

ATG Commerce executes various commerce pipelines based on the operation being performed, for e.g. loadOrder, saveOrder, commitOrder. The pipeline consists of pipeline processors, each performing subset of functionality of entire chain.

In order to place an order in OMS, add a custom processor PlaceOrderInOMSProcessor, which is added to the committOrder chain towards the end of the chain. This processor maps the ATG Order information to the OMS order information and places the order in the OMS. The processor retrieves information generated by OMS and updates the ATG Order. Place the order in the OMS as late as possible because that will reduce reasons for the ATG transaction to rollback after placing the order. This approach has proven to minimize duplicate orders in the OMS.

Inventory

Effective Inventory Management is a tradeoff between the level of functionality that will satisfy the business requirements yet still provide the best possible performance. Two key practices in terms of inventory practices are mentioned here:

- Use the out of the box Inventory Manager component to cache and manage inventory data, or design a similar approach.
- The casual browse for items should not retrieve real time inventory information. Real time inventory information can be retrieved when a user shows interest in an item on an item detail page.

Fulfillment

The fulfillment framework provided within ATG is scalable and can be used as is or customized appropriately. Making appropriate use of this framework involves keeping the following things in mind:

- Use an appropriate caching strategy for the Order repository. Keep in mind that fulfillment may be reading and locking records when they are still being accessed by the rest of the system.
- The fulfillment system should make use of the existing pipeline with custom components.
- The fulfillment system should use a JMS based approach, taking into consideration whether fulfillment processing is real-time or batch.
- Fulfillment should take place on a global ATG instance, with the out of the box backup systems in place as necessary.

Email notifications should be delivered to the customer for each step of the order process:

Email Name	Description
Fraud Email – Generic Deny	Sent to Customers when their Order is denied in the Fraud Application for any reason other than being known as a Fraud perpetrator
Fraud Email – Known Fraud	Sent to a Customer who is a known Fraud perpetrator
Fraud Hold Email	Sent to Customers from the Fraud application when they are required to contact you for further verification of their Order
Order Confirmation	Sent to Customers upon completion of their Order
Cancelled Item(s)	Sent to Customers when the System or CSR cancels their Order for any reason (inventory issues, authorization failure)
Order Status Update	Sent to Customers once all Items in their Order have reached a Shipped, Cancelled or Backordered status. Sent every time the status of one Line Item changes and is a complete snapshot of all items in the Order at that point.
Order Return	Sent to customers once an order has been returned to the warehouse and it has been entered into the system.

3.3.9 Customer Service

The Customer Service Application is important to consider when detailing overall best practices for commerce because it often touches orders and other parts of the system. These practices are generally recommended when customizing or building the CSR application:

- Use an intelligent caching scheme that takes into account the locking and changing of data.

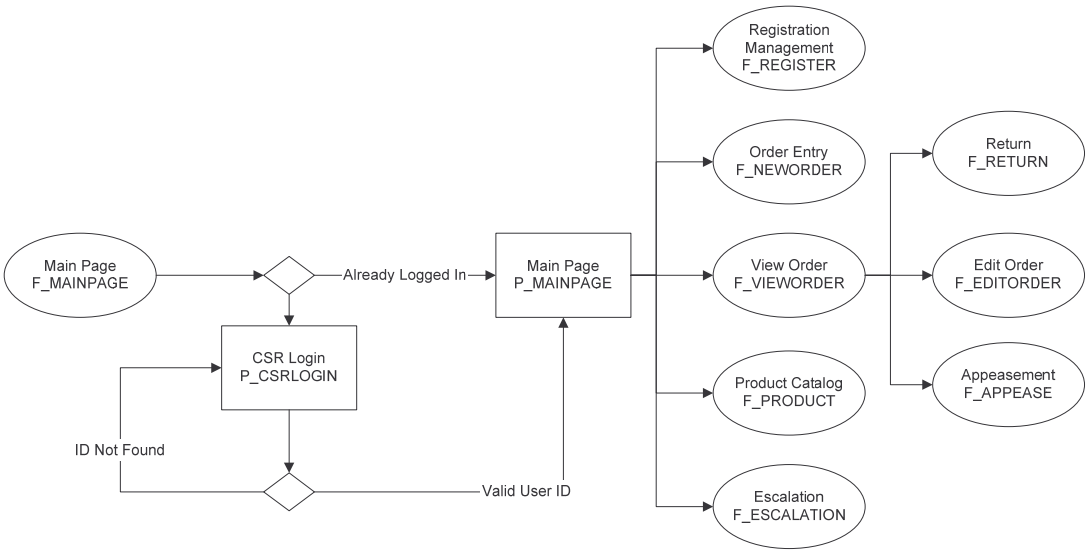
- Design the application and any custom code in order to minimize the possibility for two records to be used at once. E.g. only use read locks on the data until absolutely ready to commit changes. Handle errors in this process intelligently.

The ATG CSR Reference Application is a customizable and deployable customer service application that is provided with ATG Commerce. Built using preconfigured ATG Commerce components, out-of-the-box it enables a Customer Service Representative (CSR) to perform the following general tasks:

- Create and manage organization profiles (ATG Business Commerce only)
- Create and manage customer profiles
- Create and manage orders
- Issue exchanges and refunds
- Process returned items
- Research customer activity

The ATG CSR Reference Application (also called Customer Service Module) is intended to be used as a template for your own customer service application. The CSR Reference Application offers online order management and customer profile management capabilities. A Customer Service Representative (CSR) can use the CSR Reference Application to perform a variety of tasks, such as editing customer profiles, processing orders, and issuing exchanges and refunds.

The CSR application has the flexibility to create your own custom CSR application if you so desire. The diagram below represents a high level process flow of a CSR application using ATG CSR application. In this example, from the CSR home page, the customer service representative can perform functions such as registration management, order entry, view orders, edit orders and process returns. CSR applications built on ATG ASE can provide the same look and feel to the CSR as the web site including promotions, cross sells, up sells, etc.



The CSR Reference Application user interface is divided into "task areas" that group related customer service tasks that CSRs can perform. The CSR Reference Application includes an Organizations task area *if you are running the application with ATG Business Commerce*.

In the Organizations task area, a CSR can create and manage organization profiles; these are the profiles of business customers that buy products and services on your commerce site. An organization profile includes the

buying organization's shipping and billing addresses, cost centers, authorized payment methods, and order approval requirements. Additionally, the organization profile stores information about the members of the organization who interact with your company's site as buyers, approvers, and administrators.

In the Customers task area, a CSR can create and manage the customer profiles of users who do business on your site. A customer profile is a summary of a customer's login, contact, and address information. It also includes information about the customer's activity on your site, such as the customer's order history.

If you are running the CSR Reference Application with ATG Business Commerce, customers are associated with buying organizations. Consequently, the CSR Reference Application stores additional organization-related information about each customer. For example, each user (or customer) is identified by one or more of the following roles: buyer, approver, administrator. A given user's role indicates what function the user performs as a member of the buying organization. **Note:** A user's organization-related information is entered and modified via the Organizations task area.

In the Orders task area, a CSR can create and manage the orders placed by customers. The CSR can modify orders by adding and removing items, overriding item prices and shipping costs, editing payment or shipping information, adding comments, and canceling orders. Additionally, if you are running the CSR Reference Application with ATG Business Commerce, a CSR can also specify an existing order's cost centers and approve orders.

In the Product Catalog task area, a CSR can search for the following:

- organization profiles
- customer profiles
- existing orders
- items in the product catalog
- returns

If a customer has contacted your company before returning any merchandise, usually the return will have an associated Return Authorization Number. In the Returns task area, a CSR can process returned items or "returns." The CSR can search for the return by its Return Authorization Number, mark one or more items associated with the return as received, and specify the action to be taken with each returned item, such as returning the item to stock or sending the item for refurbishment.

3.3.10 UI and Prototype Development

HTML provides the basic elements of a UI that can be interpreted and presented through a browser, but DAF provides a number of tools that let UI developers embed dynamic elements in their pages, access and use Nucleus components, all of which ultimately result in HTML at the browser. Two fundamental building blocks are DSP Tag Libraries and Form Handlers. As well as these constructs, DAF provides facilities for creating multi-lingual Web sites that can serve multiple audiences from a single Web infrastructure.

DSP Tag Libraries allow a page developer to embed dynamic content and access Nucleus components from within a JavaServer page. The existence of DSP tag libraries pre-date JSTL, so DAF provides two versions of the DSP tag library; one uses the JSTL expression language, and the other uses an ATG expression style.

Both libraries give the page developer a set of functions for manipulating dynamic content into their pages. These include tags for iteration, data access, XML manipulation, transaction demarcation, and access to personalized content.

One important DSP tag is the 'droplet' tag. In the early days of dynamic page compilation, ATG realized that it was vitally important to separate the page layout code from the application code. Servlets made it possible to pass parameters, but all the resulting HTML mark-up was embedded in the Servlet's Java code. This meant that any page development required a Java programmer. To prevent this, ATG developed a concept called a 'droplet.'

A droplet is a Nucleus component, so it can directly access any other components in Nucleus. Because droplets are Nucleus components they are also configured using property files. Most importantly, droplets can have multiple output blocks that are rendered as the droplet requires. This enables a droplet to conditionally render output without having to force the page designer to figure out the logic.

This 'droplet' DSP tag is a bridge between the JSP page and the full power of the Nucleus component model, especially remembering that both the Data Anywhere Architecture and the Patch Bay Messaging system are themselves instances of Nucleus components.

Using the DSP Tag Libraries, page developers have an easy way to bring any part of their application into their page design.

Form handlers are also an important UI construct. DAF provides DSP tags for combining HTML forms with properties of Nucleus components, but sometimes, more sophisticated processing is required for complex forms. Form handlers are the means for providing this extra degree of processing in dealing with forms. Form handlers provide the page developer with tools to evaluate the validity of form input data, detect missing form/field data, direct users to alternative pages based upon the input data, read/write form input data from/to a Repository and perform data type conversions and formatting.

Form handlers promote good programming principals. It is all too easy for application developers to become accustomed to embedding too much Java directly into JavaServer pages. Form handlers help developers split the form processing logic into components that improve maintenance and readability of the individual pages.

Internationalization – Public Web sites are, by their nature, open for all to see. Increasingly, businesses are centralizing their Web infrastructure, but serving multiple countries, languages and currencies.

DAF uses locales to determine which language/content should be seen by which visitors. The locale setting can be taken from the browser request or a user profile.

To separate the language-specific elements of an application, and especially the UI, DAF uses ResourceBundles, which contain all of the message and UI elements that are used by an application, rather than messages being written into the code. Translating an application into a new language then becomes a matter of translating and creating a new ResourceBundle, instead of having to touch application code.

Together, the DSP Tag Libraries and Form Handlers provide convenient mechanisms for developers to tie their UI designs to the application code. They promote good programming practice for code separation so that the resulting application can be maintained and enhanced over its lifetime. These components use the "Model-View-Controller" design pattern.

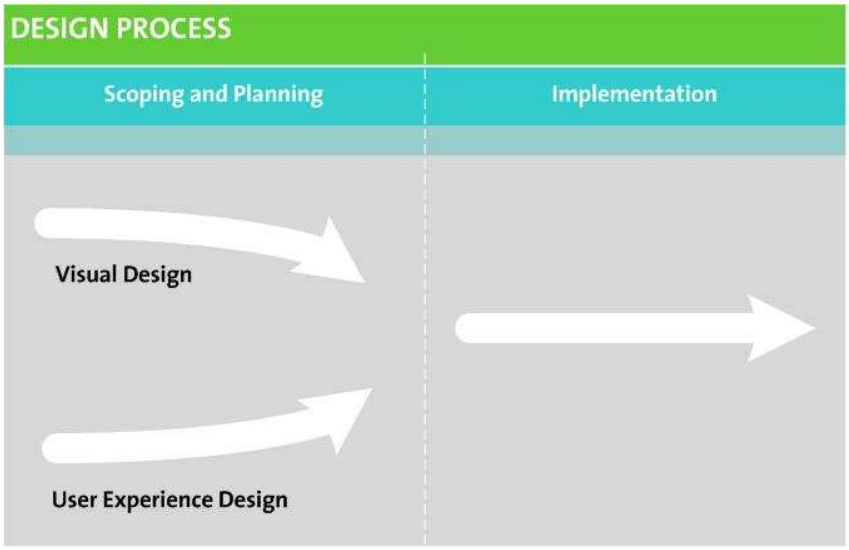
The information architecture focuses on the interaction between the user and the system. Effective site development allows for rapid testing and refinement of interactions using tangible communication tools such as:

- Site maps
- Paper mockups
- Wire frames
- Static HTML prototypes

These tools allow for an emphasis on creation of prototypes which users and system owners can touch, feel and comment on. This allows for feedback on each successive iteration. Use cases are not good at capturing detailed layout and design requirements. HTML prototypes are the foundation of the construction phase. Web

developers literally "tear up" the prototype and "stitch it" to the underlying object model. For ATG applications, 80% of the underlying model is already built for you, so puts you very close to completion.

Customer Respect Group, an independent research company, rates web sites on how well they serve online customers. The companies scoring favorably on the survey shared consistent, high ratings on each of the individual categories: simplicity (ease of navigation), responsiveness (quick and thorough responses to inquiries), privacy (respects customer privacy), attitude (customer-focus of site), transparency (open and honest policies) and principles (values and respects customer data). The strongest category for the 100 companies was simplicity, with an average rating of 7.1. Companies with a strong showing in this category usually have a site map that outlines the entire site. Make it easy to use in terms of one to two clicks get you to anywhere you want to go on the site.

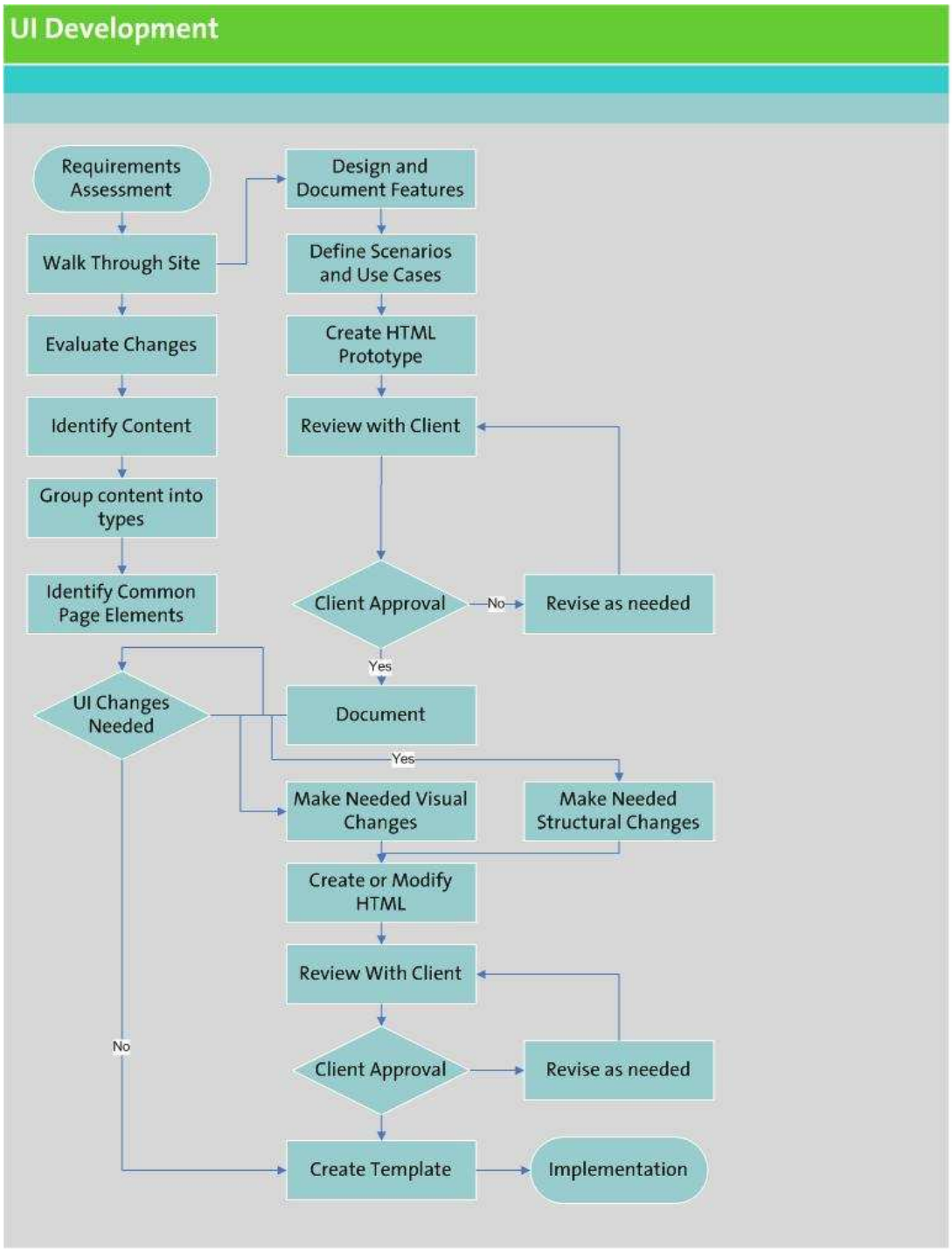


Knowledge of the existing system, capabilities of the new system, and requirements documented in the Functional Specification are used to create use cases. These use cases are used as detailed agendas for the prototype (pilot) session. The use cases include the goals and objectives for the prototype and exactly which functions and features require testing. The prototype is where we test your business model and requirements against the page flow and navigation. The team creates a schedule, which includes the participants in the prototype(s), when and where the prototype takes place, and the equipment requirements. The project team must also define test data. Test data can be manually entered to understand the basic software concepts and not cloud the prototype with data conversion issues. A more elaborate prototype would include a review of some converted data and interfaces. Consult with your ATG architect regarding the use of converted data during the prototype phase. Converted data can sometimes distort the business value of the prototype by introducing incomplete or bad data, or old methods of doing business. Most of the conversion and interface testing will take place in the Implementation phase. Until all pilots are complete (it can be an iterative process, if necessary) the full conversion or interface requirements may not be known.

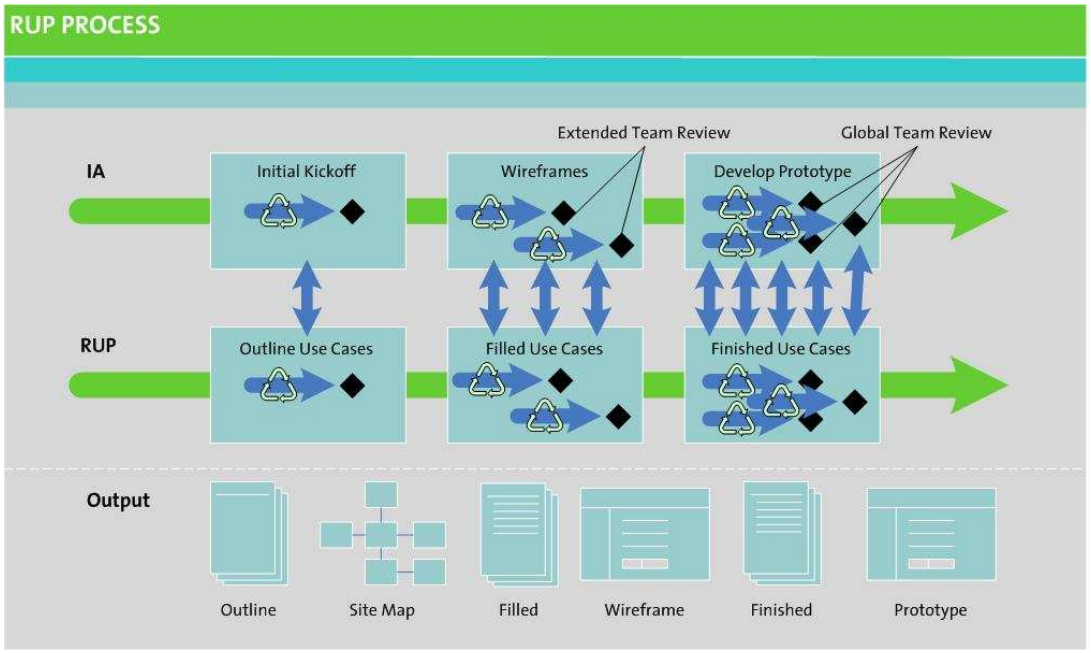
The Prototype is a process used to determine how the software will be used to perform the business activities. The project team meets in a conference room setting and follows the detailed scripts, which simulate the processes of the business, from data input to expected outputs. The prototype tests various alternatives to satisfy the business requirements. The goal is to determine the best solution for each requirement. A team member acts as scribe and records all decisions and action items identified during these sessions. Below is a diagram that describes the iterative process of building the user interface and using use cases in order to prototype the website.

Slots are containers that you can use to display and manage dynamic items on your Web site. You use targeting servlet beans to include slots in site pages, and you use scenarios to fill them with content.

Note that, as you could with previous versions of ATG personalization products, you can use targeting servlet beans without slots to display dynamic content. However, slots provide more power and flexibility than targeters. Slots have better caching capabilities, which can make displaying content faster. In addition, because you use scenarios to display your slots, you can take advantage of scenario features to help you manage the delivery of dynamic content. For example, scenarios allow you to set up an empty slot that generates its own request for content when a site visitor displays the page. With slots, you can also display content other than repository items. You can set up slots as components that you register with Nucleus, or you can send them as properties of JMS messages.


UI Development
Requirements Assessment
Walk Through Site
Evaluate Changes
Identify Content
Group content into types
Identify Common Page Elements
Design and Document Features
Define Scenarios and Use Cases
Create HTML Prototype
Review with Client
Client Approval
No
Revise as needed
Yes
UI Changes Needed
Document
Yes
Make Needed Visual Changes
Make Needed Structural Changes
Create or Modify HTML
Review With Client
Client Approval
Revise as needed
No
Create Template
Implementation

ATG recommends that the project team compile the results of the prototype into a summarized report for presentation to the Executive Steering Committee. The following diagram explains the relations between an IA (Information Architecture) and RUP (Rational Unified Process) based process (use cases and wireframes) and how to bring these approaches together to present completed use cases in conjunction with a completed prototype.



3.3.11 Database Design

Effective database design needs to be completed early in the process to understand the entity relationships. The SOLID database included with ATG is provided so that customers can sample ATG functionality. You will need to install a production-ready database before you can begin building your Web application.

When creating a database, you will require a data model for the target database to define and create the database schema. When complete, the target database may then be populated with customer or sample data to create the required database. In this case, the input work products necessary to create a sample database are as follows:

1. The data model
2. The database schema
3. The sample data

The output work product of the database task is a sample database ready to be used by the project team during the implementation.

One of the key features of ATG is the Repository. It acts as an abstract layer between the application and the database. In essence, data from potentially different data sources are mapped into objects; these objects are in turn accessed by the application. The mapping is defined via an XML file.

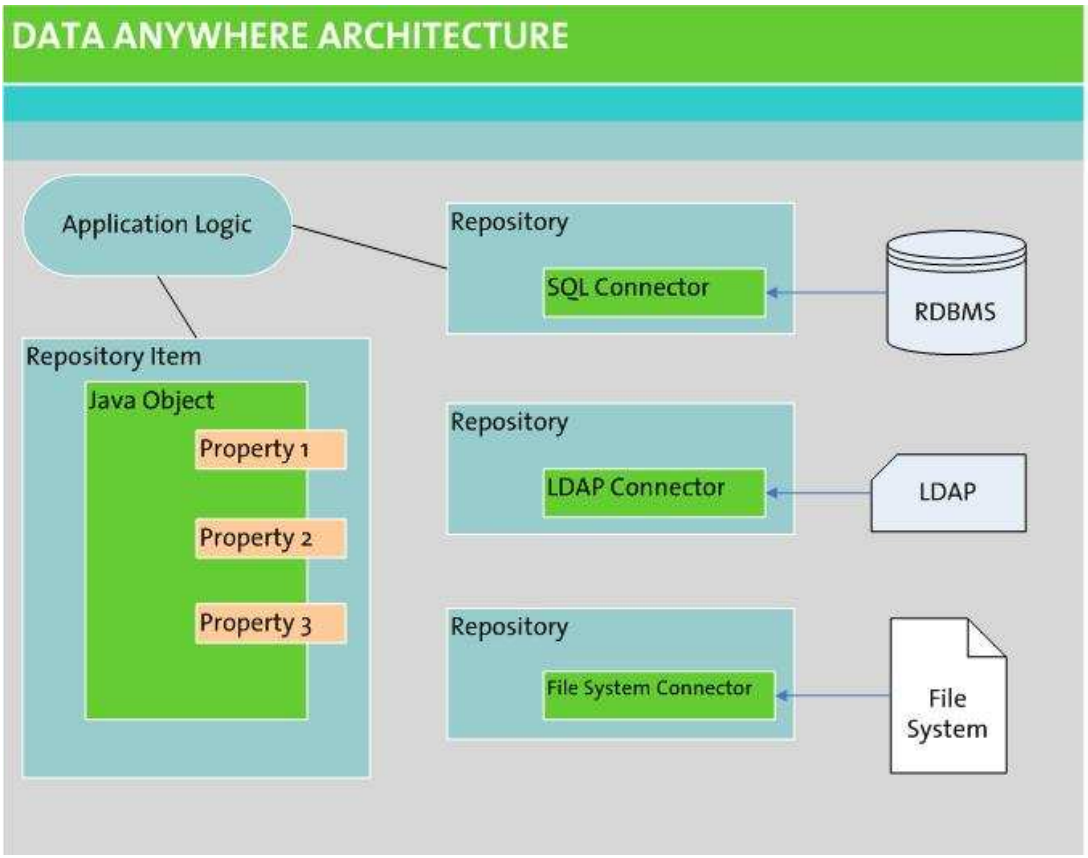
Data access is a large part of most Internet applications. The ATG Data Anywhere Architecture provides a unified view of content and data across a business for organizations and their customers. The core of the ATG

Data Anywhere Architecture is the Repository API. Through the Repository API, you can employ a single approach to accessing disparate data types, including SQL databases, LDAP directories, content management systems, and file systems. The ATG Data Anywhere Architecture offers several advantages over the standard data access methods such as Java Data Objects (JDO), Enterprise JavaBeans (EJB), and Java Database Connectivity (JDBC). Among the differences:

- Data source independence - The ATG Data Anywhere Architecture provides access to relational database management systems, LDAP directories, and file systems using the same interfaces. This insulates application developers from schema changes and also storage mechanism. Data can even move from a relational database to an LDAP directory without requiring re-coding. Java Data Objects support data source independence, but it is up to vendors to provide an LDAP implementation.
- Fewer lines of Java code - Less code leads to faster time-to-market and reduced maintenance cost. Persistent data types created using ATG Data Anywhere are described in an XML file, with no Java code required.
- Unified view of all customer interactions - A unified view of customer data (gathered using web applications, call center applications, and ERP systems) can be provided without copying data into a central data source. This unified view of customer data leads to a coherent and consistent customer experience.
- Maximum performance - Our intelligent caching of data objects ensures excellent performance and timely, accurate results. The JDO and EJB standards rely on a vendor implementation of caching which may or may not be available.
- Simplified transactional control - The key to overall system performance is minimizing the impact of transactions while maintaining the integrity of your data. In addition to full Java Transaction API (JTA) support, ATG Data Anywhere allows both page developers and software engineers to control the scope of transactions using the same transactional modes (required, supports, never, etc.) used by EJB deployment engineers.
- Fine-grained access control - You can control who has access to which data at the data type, data object, even down to the individual property using Access Control Lists (ACLs).
- Integration with ATG product suites - Our personalization, scenarios, commerce, portal, and content administration applications all make use of repositories for data access. A development team is free to use EJBs along side of ATG technology, but the easiest way to leverage investment in ATG technology is to follow the example set by our solution sets. The ATG solution sets satisfy all of their data access needs using repositories.

With the ATG Data Anywhere, the application logic created by developers uses the same approach to interact with data regardless of the source of that data. One of the most powerful aspects of this architecture is that the source of the data is hidden behind the ATG Repository abstraction. It would be easy to change from a relational data source to an LDAP directory since none of the application logic would need to change. Once data is retrieved from a data source it is transformed into an object-oriented representation. Manipulation of the data can then be done using simple getPropertyValue and setPropertyValue methods. The Repository API ties in closely with ATG's targeting APIs, so you can retrieve items from the repository based on a variety of targeting rules, as well as retrieving specific identified items.

The figure below provides a high-level overview of the ATG Data Anywhere Architecture.



ATG is a layered product. Each layer extends functionality and often requires additional properties or data elements.

Rather than adding columns to existing tables, ATG creates new tables to hold the data and makes use of Repository mapping to extend existing object or create new ones as appropriate.

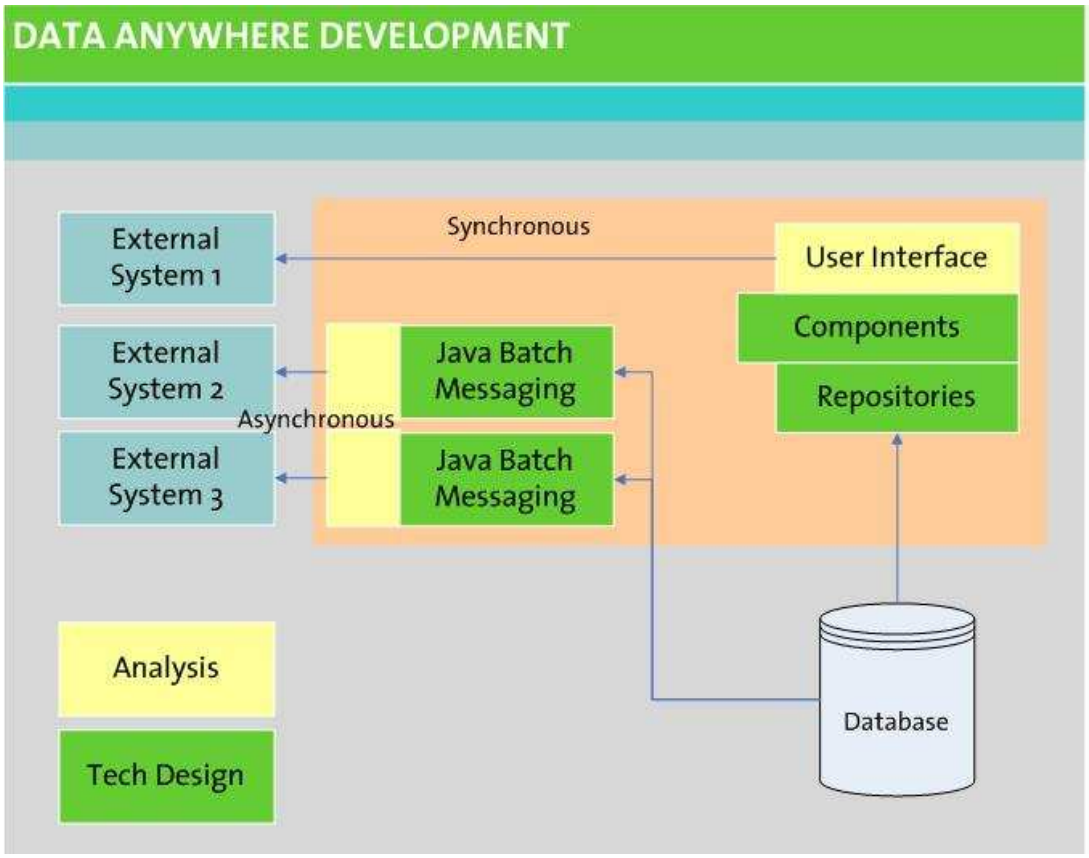
Application specific data is grouped and stored in distinct tables bearing a prefix identifying the application. This can be done in any manner the customer chooses, assuming good database design standards are being followed for performance and efficiency. This approach constitutes the Best Practice vis-à-vis the structuring of the data base schema.

ATG tables should always be extended and not modified directly. In addition, ATG tables should not be renamed. This will help with the upgradability and maintainability of the site for future releases. Each new release of ATG software comes with DB scripts that allows for ease of uploading previous releases of ATG software tables to the new release table schema. Modifying tables directly and renaming tables will require manual intervention to upgrade these tables to the current release.

3.3.12 Conversion, Interface & Enhancement Design

A detailed conversion plan helps ensure a smooth and orderly transition to the new system. The plan must assign responsibilities to specific individuals, assess levels of required effort and identify the time frame for the performance of each work step. By methodically addressing all conversion requirements, the project team can identify potential problems and allow for additional resources, procedures and personnel.

Define and document the data conversion, interface and enhancement requirements into a data conversion plan. The conversion procedures are the main tools used to perform the work steps in the conversion plan. Most of the procedures should deal with the creation of conversion files or tables, including the maintenance of new files or tables until the system is fully converted. Keep in mind the affect of your conversion on the systems outside the scope of your project.



Integrations are important to consider because they often affect the performance of a ATG system. This section only deals with real-time processing integrations such as Cybersource or another verification software vendor.

- Any integrations should use existing pipelines and extend these in an intelligent manner where appropriate
- In certain situations integrations can be a potential bottleneck. To avoid these, out of the box functionality, for example, caching, and resource pools, should be effectively along with any necessary resource monitoring, and is, intelligent error handling. Where possible, the existing pipelines should be used to allow for a more flexible extension to ATG's out of the box mechanisms.

Common integrations into ATG applications include:

- Cybersource - for credit card authorization, settlement, and crediting (an ATG API is included)
- Cybersource – for credit card authorization, settlement, crediting, tax calculations, and address verification (an ATG API is included)
- Taxware – for tax calculations (an ATG API is included)

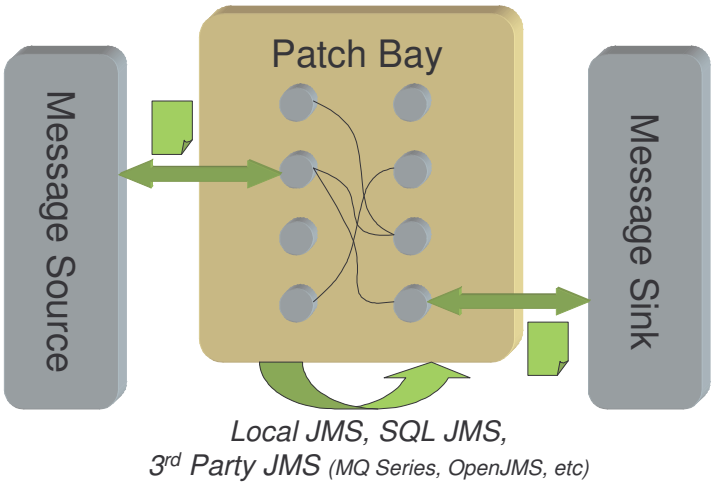
- Security – Netegrity
- Fraud Management – Clear Commerce
- Inventory – ATG provides a set of APIs for the rest of the application to know how to check or update inventory on an item or do things like send backorder or preorder notifications. This is probably useful only in hard goods sales for inventory check and out of stock conditions (an ATG API is included – access to SAP, Peoplesoft, Siebel, etc.)
- Supply Chain/Fulfillment – This is the embodiment of post-checkout processing on an Order. Once the application determines that a checkout has been successful, the fulfillment Process determines which fulfillers should be responsible for fulfilling which items and parcels them out appropriately. In addition, the Fulfillment Process allows for different categories of fulfillers, which allows for differentiating hard goods from soft goods, for example, or one fulfiller from another for picking and shipping to the customer (ATG persists the order to the database for access to backend systems i.e. SAP, Manhatten Associates, Retek, Peoplesoft, PKMS, Yantra, etc.)
- Content Management – due to the repository API data abstraction layer ATG provides to content repositories, no special requirements are needed to integrate ATG with a content management system.
- Search – ATG integrates with all the common search engines. No special requirements are need to search an ATG website. Common data tagging applies.
- Analytics - Site Analytics is a fairly straightforward installation and configuration. It requires little maintenance and performs quite well. ATG has a relationship with Coremetrics that also includes an auto tagging feature.
- Akamai – for local/distributed data cache management – no special requirements needed for ATG solutions, but should be considered as part of the site design.
- Other custom or in house solutions

The Java Messaging Service (JMS) defines a standard way for different elements of a J2EE application to communicate with each other. The JMS API defines the interfaces for sending and receiving messages. It also defines the semantics for message delivery and acknowledgement, and how message delivery should behave in a transactional environment. The API is flexible enough to allow applications to work with existing enterprise messaging products.

The Dynamo Application Framework™ provides a set of tools for working with the Java Messaging System (JMS), which is part of the J2EE specification. These tools sit above two JMS 'providers' that are delivered with DAF:

- **Local JMS** is an in-memory JMS provider designed for high-speed and low-latency synchronous messaging between software components running in the same application server process. It combines the open-ness and flexibility of JMS with the high-performance requirements of an enterprise Web application. The ATG Adaptive Scenario Engine and ATG's applications use Local JMS for most of their page and session level messaging needs.
- **SQL JMS** is a distributed JMS 'provider.' It uses a SQL database to handle communication between components either within the same application server process or running in different processes. It also provides a simple mechanism to integrate with external systems. SQL JMS provides a unique extension that allows the sender to specify a message's delivery time, providing a resilient timer mechanism that ATG scenarios use extensively.

To help simplify the configuration of JMS and facilitate the integration of 3rd party JMS implementations, DAF provides a messaging management facility called Patch Bay. Patch Bay simplifies JMS application creation. It separates an application developer's 'business' code from lower-level JMS code. It does this declaratively by using configuration files to define the sources, sinks, and topology of a JMS-based application, so a developer need not create this programmatically. Since Patch Bay is represented by a Nucleus component, the sources and sinks of the messaging application are automatically created at application startup, so an administrator doesn't have to remember to perform the task.



The DAF messaging architecture is a fundamental element of ATG's applications. The ATG Commerce application uses the messaging architecture to communicate significant events from one part of the application to another regarding orders, inventory, promotions, etc. The messaging architecture makes ATG Commerce very extensible, since additional functionality can be written as discrete components that subscribe to existing events (message queues), rather than spaghetti code that would have to be added to the right part of an application.

The messaging architecture also forms the underpinning for what is arguably ATG's most unique feature: Scenario Personalization.

The ATG RPC API is intended to be as generic as possible so implementations can support a wide variety of transports and remote systems. However, this flexibility also puts a lot of burden on the implementer Other significant conversion procedures relate to control, backup, and contingencies. Contingencies outline the actions necessary if the system cannot be converted by the expected date. For more information review the Web Services and Integration Framework Guide.

The database Copy and Switch features assist you in moving your product catalog and price lists data from one environment to another, for example, from a staging environment to a "live" production environment. The database Copy feature enables you to copy product catalog and price lists data from one database to another. The database Switch feature enables you to switch the product catalog and price lists on your Web site to use a different data source. Both database Copy and database Switch are ATG Application Framework (DAF) features that can be used with any database. However, ATG Commerce provides a user interface for performing a database copy or switch. This user interface requires initial configuration before you perform each type of update for the first time. (See the configuration sections in this chapter for more information on the configuration processes.)

The project team gathers the data required for conversion from existing manual and automated files or tables. Some of the data may be validated and corrected within the existing system. Other data may be captured, validated, corrected and maintained until conversion.

The project team may use special ATG programs or utilities during this period to help analyze the conversion results. Follow detailed procedures to ensure the data is accurately transcribed from its original format.

The data conversion design includes the following tasks:

- Establish Cutoff Date
- Define Conversion Requirements
- Develop Conversion Plan
- Design Conversions/Data Mapping
- User Verification and Sign Off of the Conversion design

For integration development, define and document the integration needs to implement your new applications. The integrations developed may be for long term use or “throw-Away" integrations to serve as bridges for one time use. Keep in mind that integration decisions can impact systems outside the scope of your project.

Integration Development includes the following tasks:

- Defining Integration Requirements
- Design Integration/Data Mapping
- User Verification and Sign Off on the Integration design
- Test and debug

3.3.13 Reporting and Analytics Design

Through a relationship with Coremetrics, the leading provider of hosted analytic solutions, ATG integrates a robust, easy-to-use auto-tagging feature with Coremetrics' best-of-breed data collection and reporting offering.

With auto-tagging, only available from ATG, the laborious process of manually tagging content is a thing of the past. Whether publishing catalogs, launching a new marketing campaign, or updating service content, IT staff can reduce tagging time by up to 75 percent. As a result, business users will have speedier access to the data flow and site traffic analysis they need to improve customer satisfaction and revenue generation.

By combining ATG's unmatched customer experience technology with the leading hosted analytics solution, you gain a more cost-effective, less resource-intensive way to deploy, assess, and maintain all of your Web site content – from catalogs to marketing campaigns, special promotions, and service offerings.

In addition, all repositories have the ability to execute queries. A query is a request to find all of the items of a particular item type that fits a certain set of criteria. Those criteria are specified in terms of the item type's properties. For example:

- Find all Person items whose age property is greater than 30

The Repository API can express a wide variety of queries, including queries that match patterns in text, query through collections, or even query through complex values. Queries can also specify the order in which results should be returned, and can specify which results from a large result set should be returned. For example, a query can express something like:

- Find all Person items whose last Name starts with 'A', whose interests includes 'biking', and whose addresses property contains an Address with a zip Code of '02139'. Order the results by last Name, and return only items 10-20. Queries can be built and executed using the Repository API. For complex queries, this can be a tedious task. In these cases, or in cases where the Repository API should not be used directly, queries can be represented in a textual form called Repository Query Language (RQL). See the Repository Query Language section in the ATG Repository Guide for details of this language and other information about repository queries. In most cases, however, repository queries can be

constructed easily using targeting UI components in the ATG Control Center. Define the non-standard reporting requirements based upon the mapping of your existing reports against the standard system reports. Evaluate the best approach for the development of the report, which may include report writers, interactive views, or custom programs. Also, evaluate the need for additional reports that provide added business value.

Custom Report Design includes the following tasks:

- Defining Custom Reporting Requirements
- Identification of the proper report writer tool for each report
- Design Reports
- User Verification and Sign off on the designed reports

Keep in mind Desktop Tailoring may encompass desktop views and filters, which may minimize the need for reports. In addition, testing, analytics, and reporting tools are critical to reducing shopping cart abandonment, enhancing revenues, and cutting down on content deployment times. ATG Commerce also offers:

Enhanced shopping cart abandonment – Most shopping cart abandonment reports simply calculate abandonment rates based on non-purchase in a single visit. But ATG tracks a customer's Web site activity throughout the purchase or non-purchase of items – from the time a customer leaves a shopping cart to the time she returns to it, if she ever does. With this level of understanding, you can quickly identify areas of improvement, helping you turn more shoppers into buyers.

Business process tracking – ATG enables business managers and IT staff to define and track various stages in the purchasing cycle, generate reports that evaluate site performance and buyer behavior, and pinpoint opportunities for improvement.

This feature allows ATG customers to define paths through their applications that correspond to business processes, name the various stages of the path, and get reports on the progress of end-users as they move through the processes. The default example in the product is the "Shopping Process", which consists of the various stages of shopping from browsing to selecting to purchase, but this is a pretty generic feature as it stands; you can name your own stages and place "markers" on your pages that cause the stage counters to be updated. For example, this could be applied to customers looking at mutual funds as well as people buying clothes.

Business Process Tracking is ATG's version of "pipeline dropoff reporting" or "funnel reporting" in the web analytics world.

The idea is that business users can (with development help) define one or more processes that they want to track -- e.g. "New Customer Acquisition", "Purchase Process", "Email Subscription Sign Up", etc. Each process can have multiple stages, so the Purchase Process can have stages like BrowseCatalog, CartItem, BeginCheckout, ViewShippingCost, ConfirmOrder, etc.

ATG provides a set of page components, scenario actions, and API's that an ATG customer can use to specify that a user has reached a particular stage in a particular business process, or that an order has progressed to a particular stage for processes that are order-oriented like checkout.

ATG also provides page components and scenario conditions that allow you to test whether the user or order has reached a particular stage so you can take appropriate action -- e.g. when a user logs in with a shopping cart that is in "ViewShippingCost" stage of the checkout process, you might offer to take them directly to the next step in checkout rather than making them go to the shopping cart and check out all over again. Or you might target different content to people who are in stage 1 of some process vs. stage 5 of that same process.

Finally, ATG provides simple reporting to show how many customers or orders have reached each stage of a process, and how the number drops off as you go from stage to stage. A simple report might provide the following data

PURCHASE PROCESS DROP OFF – May 1 – May 7, 2005

Stage Name	Visitors
001. Browse Catalog	10,000
002. Add Item To Cart	1,000
003. Begin Checkout	900
004. View Shipping Cost	900
005. Enter Shipping Address	90
006. Confirm Order	90
007. Place Order	87

In this example, you discover that you're losing 99.1% of your potential customers between the time they come to your site and the time they check out, that you're 90% of them between browsing the cart and adding an item, and you're losing 90% of the survivors after they see your shipping costs, so maybe you should offer some shipping promotions or rethink how you charge for shipping (or at least how you present the info).

3.3.14 Security and Fraud Definition

This activity includes an assessment of the impact of the new system(s) on the operations and user staff. Determine resource requirements and establish new procedures for security setup, backup and recovery procedures, and performance considerations. Any additional clients or network servers should be installed at this time.

As part of the operational setup, enter user profile definitions and menu designs on the system. Review the equipment setup (PCs, printers, cabling, and communications) to determine if the system is operating efficiently. In addition, any other client machines that need to be configured for your production environment can be installed at this point.

Develop an operations maintenance plan to determine the ongoing effect of security changes, (new hires, promotions, etc.), menu setup, and installation of new ATG releases.

A Fraud Management Application may need to be designed to allow Fraud Reps to process all the Orders that are flagged as fraudulent by the Order Management System. Such a design will include Fraud Management Interface screens.

High Level requirements for such an application include:

- Orders should be reviewed and reworked within the ACC Order Management module
- Fraud tables need to be easily modifiable (add fields, change constraints, etc.)
- The System should be secure, requiring encrypted password-based login of CSR's (and the system should prompt users to change their passwords periodically
- The System will provide a timeout capability during a period of inactivity
- All credit card information must be encrypted and only transmitted via SSL or some other secure protocol

Example Fraud Screening Business Rules:

1. The Fraud logic must be configurable by the Fraud Manager (must have ability to turn the fraud screening process on and off real-time and to turn off particular checks/sections of the fraud logic (e.g., turn of the International check without turning off the entire screening process).
2. The Known Fraud Table must be fully scalable and configurable with regard to the number of fields, data types, relationships, etc. to allow for future changes in fraud trends.
3. The System must capture IP information and Country of the IP for all Orders.
4. Fraud Rules: must have the ability to develop rules on an as-needed basis. Rules must be able to be combined based on criteria of the Order. Some example fraud rules are as follows:

- Email Address (suspected email addresses, not confirmed fraudulent email addresses)
- International Billing Address
- International Shipping Address
- Suspect Countries (based on IP address)
- Email Velocity (Same email address used x times in y days)
- Canada Billing Address
- Canada Shipping Address
- Repeat SKU (more than x number of the same item)
- Shipping Address Velocity (Same shipping address used x times in y days)
- Order Total (greater than x)
- Shipping Address (suspected shipping addresses – numeric portion and zip code)
- Shipping Address Zip Code only
- Number of total items (more than x number of items)
- Number of gift cards (more than x number of gift cards, any denomination)
- Shipping Name
- Billing Name
- Suspected BIN range of credit cards
- Bad AVS Score
- Ship to is different from Bill to
- Failed Authorization
- Product-specific trends
- Suspect registered user account
- Repeat orders in the same session
- Time stamp of the browser is x hours difference from the ship or bill to country
- Known fraudulent registered user account

5. Fraud validation process will be based on weights assigned to each fraud rule in place. Every Order will go through every rule that it qualifies for, and a total fraud score will be calculated. For example, if an order is placed from a suspect country (score of 20), is using a credit card number from a suspect BIN range (score of 10), and is going to a suspect Zip Code (fraud score of 5), the over all Fraud Score will be 35. Permanently record all flags that were tripped, the scores for each flag at the time, and the overall fraud score for each order.
6. All rule weights must be able to be assigned dynamically and take affect real-time. Additionally, the fraud threshold score must also be dynamic and real-time.
7. In addition to saving the overall fraud score, each check that was triggered must be saved so that the Fraud Manager can see exactly which rules were triggered. So, for the above example, the following would be saved for the order: COUNTRY, BIN, SHIP ZIP).
8. A database flag will determine when Bill To and Ship To addresses are identical.
9. In the Order Process flow, the System will need to keep track of Authorization-related and AVS-related data: Order Number, Timestamp, each AVS attempt and score, and each Authorization attempt and result.
10. A complete audit trail must be saved for any modifications made to the fraud application: turning the app off or on, changing a threshold (e.g. from $100 to $250), adding or removing a fraud check, etc. The value before the change, the value after the change, and the person who made the change must be recorded.

3.4 Phase III – Implementation

3.4.1 Phase Overview

Completion of the design activities enables the project team to initiate the implementation phase. During this phase, the project team codes the site elements, or components, and when done, it unit tests each prior to integration the components into builds. The recommended development strategy a project team should pursue is the continuous and incremental development and code integration testing strategy. This strategy advocates use cases and then testing and integrating code components into builds, which, in turn, must be tested prior to development. The build process should define:

- A build lead who is responsible for managing the process
- A build schedule outlining when builds will be made available
- The estimated roles and responsibilities for team members who create and use builds

In addition to the build process, the project team must also define and implement an integration test strategy. This will enable the team to test and, as needed, to correct code components of builds as they are integrated into the appropriate ATG Commerce subsystems.

The Implementation phase consists of the preparation of the new system and users for going into production with the new software. This phase includes the actual writing of programs to convert data from existing systems to the new system, control file or business rules definition, and report writing, as well as planning for the actual cut over to the new system. Also, included in this phase is the user training for both application and technical personnel.

Phase Activities

- Final Application Configuration
- Conversion Development
- Interface Development
- Custom Enhancement Development
- Report & Inquiry View Development
- System Testing
- User Training
- User Policies and Procedures

3.4.2 ATG Commerce and Eclipse Development Tools

ATG supports IBM's open source Eclipse integrated development environment. The Eclipse Platform is designed for building integrated development environments (IDEs). It can be used to create diverse end-to-end computing solutions for multiple execution environments. The idea of its Eclipse Project is to create an "Apache for developer tools" -- an open source framework that provides many of the underlying services software developers need. This would be a "toolkit for designing toolkits." Not just a set of APIs, the framework will consist of real code designed to do real work.

The Eclipse Platform is the foundation for constructing and running integrated end-to-end software development tools. The platform consists of open source software components that tool vendors use to construct solutions that plugin to integrated software workbenches. The Eclipse Platform incorporates technology that is expressed through a well-defined design and implementation framework. When selecting an architecture, tool builders need three assurances:

- A level playing field and full disclosure with no hidden APIs and no hidden tool-to-tool interfaces. Eclipse delivers by shipping open platform source. Published APIs are tested by the cross-industry consortium to ensure code quality portability and performance.
- The freedom to enhance the platform to address new opportunities. Eclipse delivers the ability to create derivative works, including redistribution of the platform. The Eclipse Platform allows tools developers to focus on their core competence and new models for development technology.
- Timely response to their requests for changes and enhancements in a controlled and organized way. Through www.eclipse.org, developers can make a difference by collaborating and contributing to the platform.

Open source is the only way to deliver an open platform for tool integration. But open source also has several other advantages.

Why rebuild something, when it exists in a working format already? By using the open Eclipse Platform, tool builders can focus on domain specific expertise and functions (thus focusing on what makes them successful), while providing the tooling infrastructure for building Integrated Development Environments. But reusing someone else's code takes trust.

Trust for any new architecture or platform is slowly earned. For instance, it is hard to earn the trust of developers for tooling that contains proprietary interfaces that limit the application to a single (for example, Windows) operating system. It is also hard to earn the trust of tool builders when different levels of APIs are shipped with different level of tool offerings (for example, Community APIs differ from Enterprise APIs).

The Eclipse Platform builds confidence and trust by providing the source code for the platform.

Open source provides all of the APIs, without internal, proprietary, or hidden interfaces. Developers, whose trust is earned slowly, can look at the source and learn. Trust the source, then launch innovation.

Open source consortiums also can produce a higher quality of code. When code review is collaborative, people put extra effort into it. The source that they contribute becomes a reflection of the work that they do, establishing both individual and corporate reputation. Trust the source, then launch market position.

Open source based upon clear specifications can deliver code that is easier to understand. An Interface describes (in black box terms) the promise of component behavior. By directly inspecting the source, developers can examine, line by line, how the code works. It is hard to trust someone else's interface. Trust the source, then inspect technology.

Open source can be easier to debug. Late at night, when encountering a bug, source code can speed identification of the root cause. It could be your fault, or the fault of the platform and environment. With access to the source, guesswork is eliminated. With access to a collaborative discussion forum, it's even possible to compare notes with someone else familiar with the environment or problem. If the problem appears to be in shared open source, it's easy to patch it and attempt a workaround. Trust the source, then verify the base.

Flexibility is a fundamental value of Eclipse. With the open Eclipse Platform, an unsatisfactory component can be modified to suit your needs. For example, if the editor is inadequate, create your own, or plug in a popular one created in the open component market established by the Eclipse Platform. Want to tie a new deployment platform (for example, a set top box) into existing end-to-end support? Trust the source, and then create a plugin.

The Eclipse Platform has been released for and tested on Windows NT, Windows XP (Beta), Windows 2000, Windows 98, Windows ME, and Red Hat Linux Version 7.1. Eclipse technology is written in the Java language, making it easier to use on a wide variety of developer workstation platforms.

The design principles that eclipse was written to are as follows:

- Facilitate seamless integration of tools within and across different content types and tool providers.
- Support the construction of a variety of tools.
- Support an unrestricted set of tool providers, including independent software vendors (ISVs) such as ATG.
- Support tools to manipulate arbitrary content types (including HTML, Java, C, JSP, EJB, XML, and GIF).
- Support both GUI and non GUI-based application development environments.
- Run on a wide range of operating systems, including Windows and Linux.
- Capitalize on the popularity of the Java language to write tools.

How does Eclipse differ from .Net? .NET is designed only for use with Microsoft platforms, through a proprietary interface that is dictated by Microsoft. As .NET changes, developers must react. In a world characterized by so much more than "wintel" technology, .NET is at a considerable disadvantage. End-to-end computing projects that need to integrate servers, workstations, embedded devices and handheld PDAs run on many other mature and cutting edge execution environments. This involves many powerful CPU architectures and operating platforms like OS/390, Linux and QNX.

Since Eclipse is available through the open-source common public license, with all APIs and extension points clearly documented, it allows tool developers to support any number of runtime environments including those from Microsoft.

The idea of its Eclipse Project is to create an "Apache for developer tools" -- an open-source framework that provides many of the underlying services software developers need. This would be a "toolkit for designing toolkits." Not just a set of APIs, the framework will consist of real code designed to do real work."

"In Eclipse everything is a plugin. The Java IDE doesn't have a special status and is just another set of plugins demonstrating the seamless extensibility of the platform. Turning the Eclipse Platform over to an open source initiative enables tools builders to do the same and to not only contribute new plugins but to also help improving the existing platform plugins," said Erich Gamma. "As a result, large enterprises and enterprising individuals have a level playing field for tool integration."

3.4.3 Using Web Services

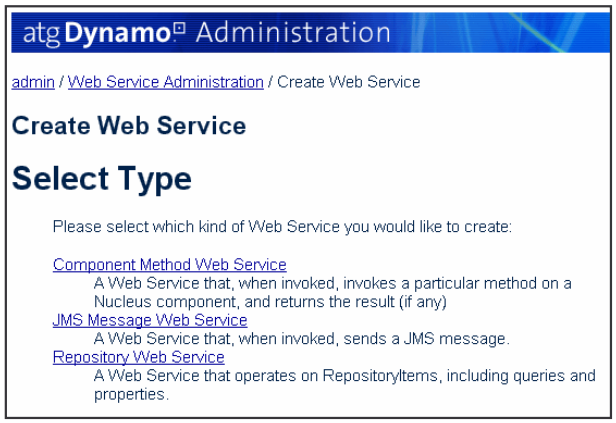
Web services use established Internet standards like HTTP, XML, and URLs as the building blocks for making application integration easier than ever before. They allow the functions of one enterprise application to invoke another, and they help the enterprise accelerate application development and ensure interoperability. The use of established Internet technologies makes Web services easy to deploy and manage so that a standard Web browser can access enterprise applications and data -- and the enterprise can apply its existing IT skills to set up and maintain a Web services infrastructure.

DAF provides a very simple mechanism to create Web Services from three types of Nucleus component:

- **Component method Web Service** – The component method wizard allows an application developer to select a specific method on a Nucleus component, and DAF will take all of the necessary steps to expose the method as a Web Service.

- **JMS message Web Service** – ATG applications make extensive use of JMS as the primary communication between software components within the system. The JMS message wizard allows an application developer to select a specific JMS message type that is used within the system, and create a Web Service that sends the appropriate message type. As a result, external applications can communicate with an ATG application by invoking events, through Web Services, and the ATG application will not differentiate between whether the event came from within or outside of the application.

- **Repository Web Service** – As already discussed, Repositories are ATG's logical view of data which can be used by an application. The Repository Web Service wizard allows an application developer to choose an already defined Repository, and quickly present a set of Web Services to allow access and manipulation of data in the Repository. Hence, all Repository data sources defined within the Data Anywhere Architecture are available, through Web Services, by external systems



In addition to the wizards available to create Web Services from new or existing component, message, and repository assets, ATG provides a number of pre-defined 'application level' Web Services. Many of these expose elements of the ATG Adaptive Scenario Engine, such as allowing a 'login' or a 'customer profile creation' through a Web Services interface. In addition, the ATG Commerce application uses the same architecture to expose Commerce processing to external systems, such as the placement of an 'order' from an external procurement system.

With this rich set of functionality available to an application developer, it is very easy for an application written on DAF to expose aspects of the application to third party systems using Web Services.

It could be legitimately claimed that the Dynamo Application Framework™ was the first SOA-enabled application framework available. As many enterprise application vendors now struggle to make their monolithic applications function in a SOA architecture, ATG's applications benefit from being written on a single framework that is itself SOA enabled. The applications immediately enjoy the resulting benefits.

ATG offers extensive Web services support for commerce applications to help you strengthen and deepen relationships with customers and partners. ATG eliminates the need to deploy or configure middleware, letting you use existing Web services standards like SOAP and WSDL to reduce total costs of ownership. Web services solutions from ATG allow you to:

- Enhance customer relationships across channels
- Increase business agility and flexibility
- Improve business efficiency and decision-making
- Reduce integration cost and complexity

You can quickly apply the industry leading capabilities of ATG Commerce and the ATG Adaptive Scenario Engine across all sales channels, while eliminating much of the complexity and expense associated with traditional enterprise integration technologies. ATG Scenario Personalization, the backbone of ATG technology, gives you unmatched awareness of and control over the user experience. Now you can use scenarios from any remote system across your organization. ATG also helps you integrate business applications for creating personalized interactions in .NET environments.

Web services support in the ATG Adaptive Scenario Engine:

- Allows access to profile information and Scenario Personalization so you can create highly personalized content across touch points, integrate online and offline channels and deploy best practices for accessing Web services from within a scenario.
- Works alongside repository integration, which presents a unified, global view of all data and provides

unprecedented control and access to data sources across the enterprise.

Web Services support in ATG Commerce:

- Lets you assemble orders for review by your customers, and allows you to use the rich promotion and pricing engines provided by ATG.
- Supports inventory management and lets you open all catalogs and product information across channels.

3.4.4 Repository APIs

Providing a great online experience to customers requires access to vast amounts of information. You need to be able to access all of your company's knowledge about your customers to provide relevant information. This involves accessing content from many different data sources to provide a breadth of information. To ensure consistent service throughout all customer touch points you also need the ability to reuse existing data across multiple channels. And you need to access all this data without bringing your site to its knees.

ATG's Data Anywhere Architecture (DAA) meets all of these challenges. DAA gives developers a single API, called the Repository API, for using data resources in their applications. Behind the Repository API, DAA insulates and abstracts application developers from the specifics of the data source, so much so that the nature of the underlying data source may completely change without major impact. For example, customer data may reside in a SQL/JDBC database today, but will move to an LDAP directory in the future. DAA could handle this without having to touch any code within the application.

The fundamental construct in the DAA is a 'Repository'. A Repository is a logical view of a data resource (or resources), and to a developer, manifests itself as a set of JavaBeans to be used within their application. Like everything else in an ATG application, Repositories are represented as Nucleus components.

ATG's unique repository integration provides an efficient way of delivering enterprise-wide content to hundreds of thousands of users concurrently. A key component of the ATG Adaptive Scenario Engine, repository integration presents a unified, global view of all data across the enterprise.

Repository integration enables you to access the customer data that you need quickly and economically without moving or replicating files. It accomplishes this by mapping to your existing databases, removing the need to modify existing applications.

Repository integration provides a single method of accessing data stored in disparate systems and data types, including HTML, XML, LDAP, and SQL. Data source independence insulates IT from changes to schemas and storage mechanisms. By reducing the amount of code that developers must write, repository integration reduces time-to-market of new Web applications and reduces overall IT costs.

The Repository is described in a Repository Definition XML file, which holds all appropriate information about the data's physical location and how it is mapped to the logical view. The DAA consists of three primary Repository types for data access and manipulation.

- **SQL repository** – A SQL Repository presents a logical view of data stored in a relational database, accessed through JDBC. The Repository definition file defines how the databases, tables, rows, and columns of a relational database are presented through the Repository API. It also defines the item caching strategy to be employed to optimize database read/write performance.
- **LDAP repository** – An LDAP Repository presents a logical view of any data source that has an LDAP interface, typically used to store user data. The Repository definition file defines how the hierarchical structure and contents of an LDAP directory are presented through the Repository API.
- **Integration repository** – In some cases, data sources may not be directly accessible, or information may be returned by an application rather than directly from a database or directory service. The Integration

Repository presents a Repository API in front of some remote application processing. For example, an Integration Repository may be implemented to facilitate integration with SAP's BAPI interface, or to retrieve information through the execution of a Web Service SOAP call. The Integration Repository is an open architecture into which these specific integration technologies can be plugged, while still presenting the same Repository API to the application developer. It also gives developer sophisticated data access and caching strategies to protect the application from remote latency and downtime.

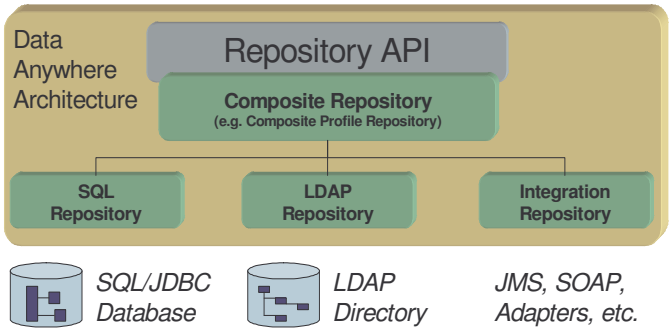
In addition to the primary types of Repository mentioned so far, there are two types of 'overlay' repository types that can be used.

- **Secure repository** – A Secure Repository introduces application level security and access control to the data being accessed and manipulated. Working with ATG's Security Management Framework, varying levels of security can be defined on the Repository contents, all the way down to individual data properties. Access Control Lists (ACLs) are written to describe the different levels of access that are provided to ATG's User Model, which itself provides a rich structure to model user, organizational hierarchies and roles.

- **Versioned repository** – A Versioned Repository introduces a versioning mechanism to one of the other primary Repository types. It provides all of the required tools to maintain, version and roll-back versions of a Repositories contents. Any existing SQL Repository may be turned into a Versioned Repository through additional configuration files. The Versioned Repository architecture is heavily used by ATG's Content Administration product, but the versioning features are open for any other type of application usage that may be customer specific. Versioned Repositories integrate closely with ATG's workflow capabilities that reside in the ATG Adaptive Scenario Engine.

A Composite Repository is the final construct that can be especially useful for building applications requiring access to data in multiple data sources and formats.

- **Composite repository** – A Composite Repository represents an aggregate view over other repository types, or over other composite Repositories (although one should not create too many layers of Composite Repository). The most common use of a Composite Repository is where a businesses customer data is distributed over multiple SQL databases and an LDAP directory, but a Web application wants a 'single view of the customer' to reduce application complexity. A Composite Repository provides some welcome simplicity.

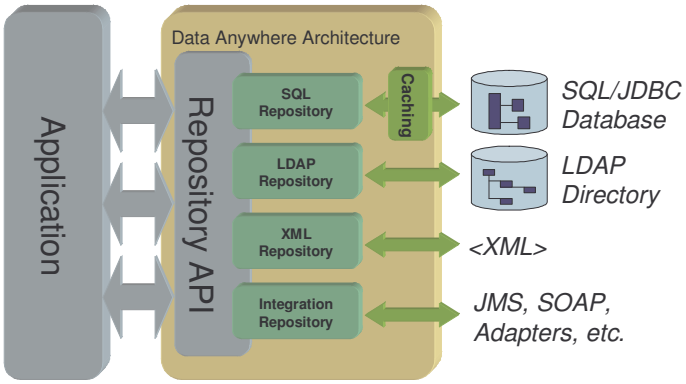


To ensure scalability of Web site usage of SQL database, the DAA provides sophisticated caching and cache invalidation mechanisms for SQL Repositories. DAA provides all of the necessary tools to manage and flush caches at the repository item level. There are also mechanisms for managing distributed caches and cache invalidation via JMS or TCP/IP.

All in all, ATG's Data Anywhere Architecture provides a rich, robust, and highly scalable set of tools to facilitate the use of enterprise data resources, while providing a loose coupling approach between data source and application.

Repository integration works alongside web services so you can expose functionality and enable features such as remote creation, modification, removal, and querying. Repository integration allows you to access and manage data from distinct sources to gain unprecedented control and access to data sources throughout the enterprise.

In addition, repository Integration enables you to achieve maximum performance capabilities through the use of advanced, proprietary caching mechanisms specifically pioneered to handle the traffic loads of top Web sites.



3.4.5 Conversion Development

From this point you begin the actual coding, testing and documenting of the programs for the conversions. Reconcile the conversion against expected results to assess accuracy. The technical lead verifies and signs off on all conversion programs.

Capturing and transcribing manually recorded data is an arduous task which if often error prone. If a large amount of data needs to be transcribed, there should be controls and validations to ensure that all data is correctly captured. Completeness and accuracy are fundamental in this task.

Data that is maintained in a present system can be reviewed and corrected in the current system or "scrubbed" as part of the conversion. Special reports or audit programs may be written to ensure data conversion accuracy. Conversion programs translate the old format into the new format at conversion time.

The conversion development includes the following tasks:

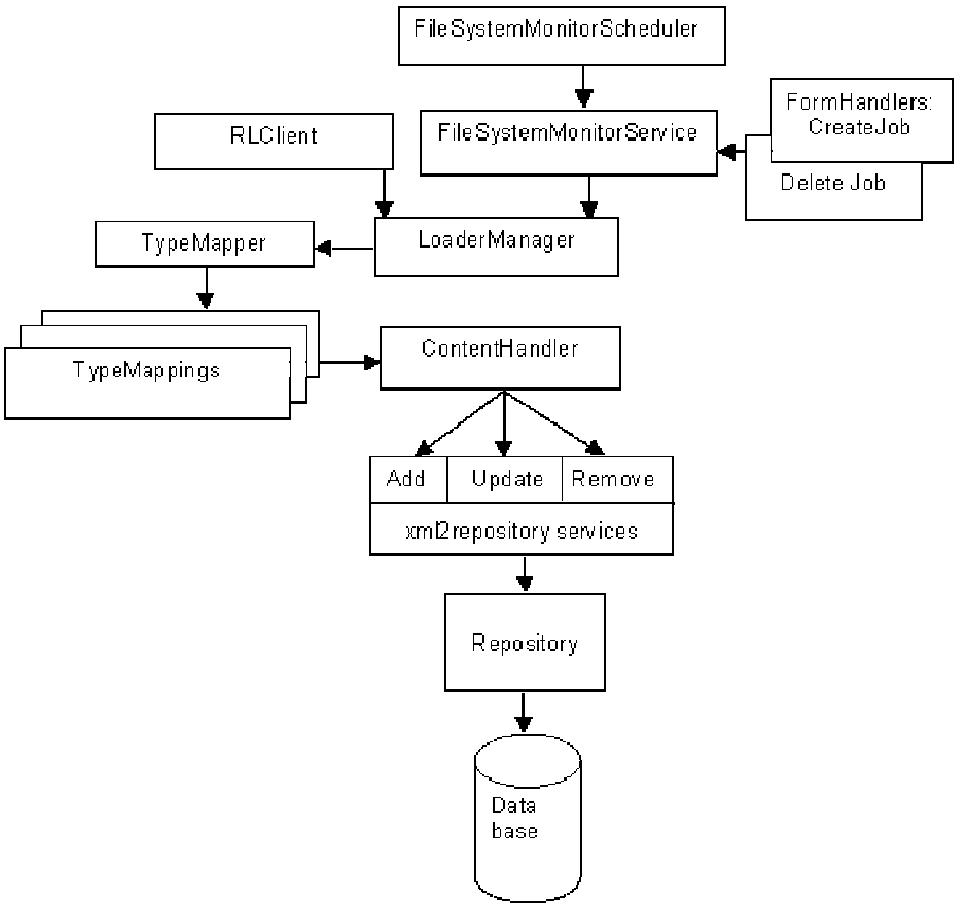
- Design/Code/Test Conversions
- Initial Execution of Conversions
- Testing and Reconciliation of the Conversion Output
- User Verification and Sign Off of the Converted Data
- Development of a Migration Plan from Test to Production

The ATG Repository Loader can take a set of files maintained in a file system, convert them to repository items, and make them available in a repository. The principal elements of the Repository Loader are components of the following classes or interfaces:

- FileSystemMonitorService

- FileSystemMonitorScheduler
- LoaderManager
- Type Mapper and TypeMappings
- Content Handlers and Back Ends
- User Interface Elements

A fully-configured Repository Loader setup might look like this:



3.4.6 Interface Development

From this point you begin the actual coding, testing and documenting of the interfaces. Reconcile the interface against expected results to assess accuracy. The technical lead verifies and signs off on all interface programs.

The interface development includes the following tasks:

- Code/Test Interfaces
- Initial Execution of Interfaces
- Testing and Reconciliation of the Interfaces
- User Verification and Sign Off of the Interface
- Development of a Migration Plan from Test to Production

3.4.7 Custom Enhancement Development

Features of the ATG applications allow you to make changes to your application through the use of framework tools (API's, business rules, etc). These enhancements can help you tailor your ATG environment to match the way your company needs to do business.

If modifications to core functions (ATG supplied code) are required, you should thoroughly define and document the proposed specifications and obtain management approval before modification work begins. Prioritization of the modifications and enhancements ensures that critical changes are completed within the project timeline.

ATG is available to perform a technical feasibility review of your proposed modifications. If a technical review is not required, we request that you inform your ATG Consultant of all planned software modifications. This information will enable us to better support your implementation and to provide more effective post-production support.

After receiving the appropriate authorization, you can begin the development and testing of the custom enhancements. Remember to develop technical documentation for all custom modifications and enhancements. This is very important when it comes time to upgrade to the current release of ATG Software.

The project plan is reviewed and revised periodically to incorporate any changes due to enhancements or modifications. The project manager monitors and approves all changes to the project plan.

3.4.8 Reporting & Analytics Development

Reporting and analytics development is often one of the most under estimated aspects of any systems implementation. This task can start as early as the beginning of the Analysis phase for database queries and continue through the Production Rollout phase. Site statistics reports can often be produced after Launch.

From this point you begin the actual development, testing and documenting of the reports. Have the functional user sign off on the delivered report prior to setup in the production environment.

3.5 Phase IV – Testing

3.5.1 Overview

Best Practice

The QA team is responsible for executing the Test Plan and the Functional Test Plan developed during the Design phase and modified as necessary during the Implementation phase. System Testing begins once the build is deemed feature-complete and unit-testing has passed. The objective of this set of activities is for the QA team to identify functional and UI deficiencies.

This QA environment has two sub-sections: one for the current project, and one for production support. There should only ever be one project in QA at a time, as only one project is scheduled for release at any one time. QA is the environment where fully automated testing is carried out against a full production data set. Additional manual testing can also be performed if some piece of functionality can't be validated in an automated way, but the vast majority of testing must be automated to achieve the rapid turn around that will be necessary. The entire test suite must be run against every build that makes it to QA. Some errors will still slip through, but a full regression test of every build prior to production will significantly reduce this number.

Much like integration Testing, QA is about functionality, not performance, though it has a larger data set to work with. Any set of dedicated hardware powerful enough to functionally simulate production will be sufficient. As a suggestion, two application server machines and a dedicated database and other integration hardware should suffice for both sub-sections.

The objective of the Integration Test is to gain agreement that the completed system meets their requirements as defined and agreed upon at the beginning of the project. It is imperative that the users assist with the Integration Test facilitated by the project team.

Typically, the Integration Test simulates an entire transaction. The project team defines test scripts which detail the critical features and functions to be tested by each functional area including external interface testing. In defining the scripts, pay particular attention to the documented procedures. The prototype scripts (if available) provide an excellent foundation for the building of the integration test scripts. Develop Integration Test Procedures, which consist of procedures for conducting the Integration Test and a formal set of pass/fail criteria within a structured plan.

The integration testing environment actually consists of multiple isolated environments dedicated to each project as well as production support. The primary purpose of this environment is to provide a place for validating third party integrations, which means that an isolated copy of those integrations will be necessary for each sub-section of integration. Since the goal is functional testing, full data is not necessary, and so the simplest way to handle this requirement is to provide a single HP-UX machine for each sub-section. Furthermore, the staging and QA environments should simulate the production environment.

The secondary objective of the integration environment is to provide a stable enough platform for the development and initial run through of automated smoke testing for QA. With this in mind, projects should only move to integration when all possible development testing has been passed.

The final step for a new project prior to its introduction into production is a full scale performance test. At this point the project has passed all functional testing, so it should be robust enough to run some combination of the QA smoke testing scripts (or even better, a playback of real user behavior where that's possible) in a large volume. Note that success in this environment is somewhat arbitrary. It should however be measured against results from previous releases. The primary purpose is to provide a definite benchmark on expected performance changes with the release of the project. Some projects will be expected to improve performance, while some will be expected to enhance functionality at the expense of performance. The only reason to fail a project at this point is if the performance loss is not worth the functionality gain and the original objectives need to be reassessed.

ATG's Performance Monitor component provides a tool you can use to monitor the performance of regions of your code. The Performance Monitor can run in different modes; it causes negligible overhead when run in normal (default) mode, but allows you to globally turn on one or more monitoring options which give

more diagnostic information. These monitoring options would typically be used during load testing but are not suitable for running on a live site under heavy load.

A release going into staging has passed all technical criteria to its deployment and needs a final business authorization. Staging provides a location to validate all changes against production configuration and datasets without impacting production. That means that the changes can be made and validated as part of a regular business day, without needing special late night events or taking the production site off line. This in turn allows unlimited time for production specific issue resolution. A project that has been passed in the staging environment can be deployed to production with minimal effort, in full confidence that any production specific errors have already been caught.

3.5.2 Critical Issues

- A good set of data and test plan is needed for testing.
- Technical lead needs to be available to the QA lead for answering questions based on testing results. A dedicated go to person needs to be assigned for this.
- The ATG and customer Project Manager needs to be involved in the triage process.
- Engineers need to attempt to reproduce the found bugs on the QA environment as well as their own machines prior to resolving.
- No one other than the QA team is allowed on the QA environment without explicit permission from the QA Lead.
- Unless another tracking system is specified by the client, the ATG team should use Bugzilla for all bug tracking. – is this true? Often, the client will want to use their own system to identify and track bugs found during their own testing. In these situations, it is imperative that we be able to map the client's bug # to our own in Bugzilla. In this case, you should maintain two separate lists of bugs: one that tracks bugs found by the client (and any found by ATG that you want to share with the client), and another internal-only list that contains all bugs, including those that we do not want to share with the client.
- No bugs should be deleted from the system under any circumstances.
- Don't let bugs fester - make sure that they are addressed. Track bugs that are not closed by the Estimated Close Date.
- Keep your ratings logical - not everything is stop-ship!

3.5.3 Activities

- Smoke Testing: A quick functional overview of the entire site. Sometimes may be accomplished through an automated script. For example the login or checkout function may be the scripted smoke test.
- Usability: The application matches the Use Cases defined previously in the Design phase.
- UI testing: The basic testing done to a site to flush out the possible spelling errors, typos, broken image links, mismatched images, widows and orphans, color issues, html/jhtml/jsp errors etc. These are generally not highest priority, though this is obviously dependent on error.
- Triage a periodic assessment of open bugs. This eliminates the engineers wasting time on bugs that are not properly prioritized. Also, bugs that are open may be postponed to a later release.
- Levels of testing: Because of the length of the testing process for the Solutions Client Projects, the testing is primarily manual. There are cases of automation for the checkout procedure, dependant on when the actual code is written and implemented.
- Level 0: the most basic, minimal level of functionality testing of an application or a unit of an application. Breakage at this level would be grounds for postponing a release.

- Level 1: testing the breadth of an application's functionality. It is the broadest of the four levels, and can also be thought of as an "extended" level 0. Level 1 testing includes all options accessible to users; sequences likely to be included in a typical usage pattern covers all non-advanced topics mentioned in documentation.
- Level 2: testing the depth of an application's functionality. Level 2 testing includes usage patterns of users who specialize in that area of software (artist's use of graphics, lisp developer's use of lisp, etc.); covers all advanced topics mentioned in documentation; exercises more cross functionality.
- Level 3: testing the robustness of an application's functionality. It covers the most detailed of usage scenarios which users may attempt. Level 3 is the most in depth level and includes sequences only performed by master users and QA engineers; covers scenarios unlikely to happen intentionally but which are potentially problematic, i.e. negative testing; heaviest interaction between related subsystems.
- API/Integration testing: Testing the Application Programming Interface (API) and testing for bugs in the relationships and interfaces between software integration. Normally this requires confirming the proper connections between an application and its database(s).
- Security testing: The security checklist is a basic checklist of the points that cause common problems. Depending on the type of secure server there are various related quick checks. (iPlanet, Stronghold, HTTPS).
- Regression: When some previously correct operation no longer has the expected and correct result.
- Load and Performance: This requires the use of an automated test tool to gauge the performance of an application under load. Performance testing is normally performed on a staging environment to confirm an applications ability to handle multiple users prior to going live. Performance testing usually measures a site's ability to serve pages to multiple users within a given time frame. Benchmarks are established and based on the type of content and number of ATG instances the site is running.
- Logging Bugs: a bug is a defect that is detected because the application does not perform to some specifications. QA procedure is: "When in doubt, open a bug." (The person logging the bug should conduct a search to see if the bug has already been logged.) Standard of bug priority within ATG are as follows:
- QA-block when an install is defective or the web server errors are fatal. These bugs must be fixed before QA can test the product
- Stop-ship the build may be installed but there are far too many failures to ship, and no workarounds exist. For example, the inability to checkout, logon, etc. All stop-ship bugs must be fixed prior to the release of a project. Only stop-ship bugs get fixed after a code freeze.
- High. There is functional activity, but all or most tests in a use case fail (but there is a workaround). Fix first, this is holding up other work.
- Medium. Fix in this release
- Low. Possibly an enhancement, task or wish list item. Possibly this bug may be ignored for this release.
- Severity levels are as follows:
- Critical: a bug that prevents a function from operating, and for which there is no workaround.
- Serious: a bug that prevents a major function from operating, but for which there is a workaround.
- Non-critical: a bug that might prevent a minor function from operating, but for which there is a relatively simple workaround, or a very obvious typographical error (as in a heading or sub-heading).
- Minor: a minor error (such as a typographical error in standard font) or aesthetic issue or objection (no error and does not fail to meet spec; e.g. distracting icon layout or font that is hard to read

Prior to conducting the Integration Test, the project team should review the Integration Test Procedures with the users in each of the functional areas. Select test data that closely represents actual "live" data, and actual "live" volumes. Define the expected results and pass/fail criteria prior to conducting the tests as benchmarks for evaluating the results. In the event that test results do not meet the predefined pass/fail criteria, ensure that you have a well-defined contingency plan.

During Integration Testing you define various cycles with related functions grouped together. Each cycle has specific objectives and may be dependent upon successful completion of previously run cycles. Therefore, the order of the cycles is extremely important. You must have available back-ups of the database at the beginning and end of each cycle. You may have to restore to the beginning of a cycle already completed to retest functions that may have failed previously.

You must thoroughly document the results of the Integration Test. Cross-reference the output generated as a result of the Integration Testing to the summarized test results. This cross-reference creates an excellent audit trail and facilitates the project team's ability to respond to questions at a later date. Present the summarized results to top management during an Executive Steering Committee Meeting. The project team, the users, and top management must be in agreement that the system as a whole is functioning as it should be and that the risk associated with going live is minimized.

3.5.4 Documentation

As a result of the implementation of new application software, custom code and interfaces must be documented, process written regarding the timing and the updating of updates, data loads, etc. In addition, maintenance policies and procedures may be affected.

The project team must present all changes to management for review and approval. Once approved, the new policies and job descriptions are communicated to the entire organization. The following is a sample table of contents for an ATG Site Administration Guide:

- Introduction
- About this Document
- Related Documents
- Project Description
- Product Requirements
- Server Software
- Client Software
- Supported Browsers
- Names and Passwords
- Nightly Data Migration
- Data Architecture
- Naming Conventions
- Behind the Scenes
- Before Data Migration
- During Data Migration
- After Data Migration
- Collecting Data
- Setting up the Staging Environment

- Setting up the Production Environment
- Troubleshooting Data Migration
- Soft Failure
- Hard Failure
- Installing the web server
- Installing ATG
- Module Directory Structure
- Build Structure
- Working with the content management system
- Installing a ATG Build on a Staging or Production Server
- Installing and Configuring a Connection Module
- Running the website
- Starting and Stopping ATG and the website
- Configuring the website
- Maintaining Servers
- Load Management

3.6 Phase V – Deployment

3.6.1 Phase Overview

During the Production Rollout phase ownership of the new system passes from the project team to the users.

After live, this phase provides you with a formal project closeout and review to ensure that the system operates efficiently and continues to meet the requirements of the users. Changes in your business environment and changes in ATG Software's products are monitored and evaluated on a regular basis to maximize the return on your investment in ATG products.

3.6.2 Production Cut Over Planning

Schedule a meeting to discuss the planned live date by application. If applicable, schedule the dates for parallel processing and communicate those dates to all those involved with the implementation. Be careful to avoid launch dates prior to holidays. Should the launch be delayed this would put the launch schedule into the holiday. Also be realistic about the timing of the launch. Having project team members work nights and weekends in anticipation of the launch only to postpone the launch and expect the project team to continue to work nights and weekends is poor project planning that creates a tired, error-prone project team at a time when you will rely on them the most.

Establish preliminary cut over responsibilities. Schedule participants for the cut over activities in the parallel (if applicable). Verify the output data produced during the conversion, keeping in mind that this task is very resource intensive.

Conduct a thorough review of the project plan including a realistic assessment of the possibility of meeting the cut over dates. If necessary, review and revise the contingency plan after assessment of the feasibility of the live date.

Schedule a visit with an ATG Deployment Specialist. The ATG Deployment Specialist will help you configure the site for production including the following activities:

- Enable liveconfig settings
- Disable checking for changed properties files
- Setup web server error pages
- Shut off serverina and HTTPServer
- Disable the performance Monitor
- Adjust the pagecheckseconds property
- Adjust the filecache size
- Enable the repository cache lock managers
- Adjust the session manager properties
- Set logging levels
- Disable screen log
- Change the default cookie hash key
- Limit initial services for quicker restarts
- Configure repository database verification for quicker restarts
- Configure document and component indexing
- Turn off document and component indexing

- Prepopulate caches on startup
- Configure repository cache modes
- Enable the protocol change servlet bean
- Prepare for visits by robots
- Fine tune performance of the JDK with Hotspot
- Adjust Solaris settings

3.6.3 Acceptance Testing

An ATG expert works with your project team to perform an acceptance test and pre-implementation readiness audit to ensure that all outstanding issues have been identified and addressed and that the system as a whole is functioning as expected.

ATG also recommends that you perform a qualitative review of the project plan to determine that all required tasks have been completed and that the plan accurately reflects the resources required for the remainder of the project. Be certain that the following tasks have been completed:

- User Training
- Policies and Procedures Documentation
- Help Desk Process Review
- User and Help Desk Personnel Certification
- Conversion, Interface, Report, and Enhancement Testing
- Resolution of System Test Exceptions
- Control File/Rules Review

3.6.4 Production Cut Over Support

Upon successful completion of System Testing, review and finalize the plan to cutover from the existing system and to support the new ATG system. Until the cut off, continue to process transactions on the old system and convert the data to the new system as historical transaction data or standing data. As of the cut off date, enter transactions and changes directly into the ATG system as production data.

Be certain to clearly define the cut over to avoid duplicate entry or non-entry of production data. Clearly identify roles and responsibilities to ensure no duplication or non-entry of transactions. Communication is the key to a successful cut over. Therefore, you must present the plan for cut over support to all users, all members of the project team, and applicable members of top management. You may need to schedule the users, members of the project team, and the operations staff for weekend or evening coverage during final conversion. Additionally, we suggest that you notify the ATG Customer Support Center of your "go live" plan to ensure appropriate coverage from ATG's support organization. Ensure that everyone involved understands the implications of a clean cut over.

3.6.5 Final Conversion

Prior to the final conversion review the project plan and the issues log to ensure that all tasks have been completed and that the members of the project team and the users understand the requirements for executing the final conversion.

The final conversion task refers to the conversion of existing data to the new ATG system. This is typically performed utilizing conversion programs which redefine the existing data to a format recognized by the new system. The volume of conversion data varies from implementation to implementation.

The build process is the mechanism for creating a unified code driver that has all the necessary code and non-code pieces to allow site testing. The project team should create a build process for the project. The build process should be finalized at the end of the design phase, and the project manager should identify a build person who will:

- Be in charge of creating regular builds
- Agree to a build schedule with the team
- Identify all elements and the contents of the builds
- Establish roles and responsibilities of team members in creating and using builds
- Create the build deployment process
- Establish actions that must be taken when a build failure occurs

The build environment must be capable of producing builds for production support and each project independently. A build environment that contains a build area for each desired target project. These areas would contain their own ATG installs to build against, as well as their own copies of third party tools used as part of the build. This can be easily implemented using a single machine.

The build environment is the single source for all builds in environments outside of development. It is also the logical place to handle code merges from production support. Projects currently in development need production support changes as soon as possible, but they also need them in a predictable way to avoid disruption. Weekly merges of production support changes into all relevant project branches is a good solution. A build master role has the necessary access to perform the merges, as well as the ability to produce a test build immediately after the merge so that developers know what needs to be fixed.

Verification of the conversion data is an extremely important component of the Production Cut Over phase. It can be extremely time consuming and is best performed by the users with the assistance of the project team. To facilitate the reconciliation process, format reconciliation reports to closely resemble existing reports produced by the old system. Encourage the users to accept responsibility for reconciling the conversion data and obtain sign-off from the users that the conversion has been performed successfully.

The last step prior to cutting over is to maintain and update the post conversion activity. Post conversion activity includes all transactions that occur between the conversion date and the production cut over date. Post conversion activity is typically entered manually and can impact either the control files or the transaction files.

3.6.6 Launch

By executing the steps of your Cutover Plan, you bring your organization live on its new applications. Each step of the cutover plan needs to be monitored to make sure that the results are in compliance with the conditions established. Your ATG Professional Services team should be scheduled on-site during the go live step to help trouble shoot any unforeseen problems.

The final step for any project is to reach production. Production deployments should be simple and straightforward. Unless a system is undergoing a major upgrade that prevents backwards compatibility, there should be no need for an outage. Following a successful deployment to production, the project branch either becomes the new mainline code base or is merged into it, creating the baseline for any projects that start after that point. From an infrastructure standpoint, all other environments should be as close to production as possible.

3.6.7 Post Implementation

Members of the project team should continue to monitor user satisfaction and system performance for the few months following production cut over. As the system begins to "settle down," project team members should be available to address areas of frustration or difficulty experienced by the users. Keep in mind that the learning curve is quite steep and that frustration is typically the result of a lack of understanding. Continue to work with the users and offer suggestions on how to maximize system usage.

3.6.8 A/B Split Testing

Companies that truly want to connect with customers and understand their buying habits take the time to test. ATG Campaign Optimizer lets you test numerous Web site variables, e.g., promotions, products, creative, and Web functionality, in your required timeframe. It's the most powerful online business asset you can own – a direct, always accessible, infinitely flexible pipeline to your customers' preferences.

Convert browsers to buyers knowledgeably.

As competition intensifies, so do time-to-market pressures. Your company must find ways to quickly test Web site effectiveness and ensure that browsers convert to buyers. ATG Campaign Optimizer provides a better alternative to other Web-based testing methods due to its ease of use, and flexibility of the tests that can be run. ATG Campaign Optimizer is the only multi-dimensional testing capability that enables you to:

- Run multiple tests concurrently (test A/B and A/B/C/D, etc.)
- Combine multiple types of tests
- Conduct simultaneous testing
- Apply segmentation to show different campaigns to different audiences
- Run tests for any length of time
- Change tests on the fly
- Provide easy to use control by business users with an intuitive user interface

A business tool for business people.

In the past, marketing staff relied on IT to create and conduct Web-based testing. ATG Campaign Optimizer enables your marketing team to create campaigns with ease - and without creating drag on your Web site. The Wizards-based interface makes it easy to set up and change even the most sophisticated tests. And the out-of-the-box report package provides easy to read results at the click of a mouse.

Optimize relationships and resources.

As market conditions change or priorities shift, use ATG Campaign Optimizer in new and different ways to deepen customer relationships or hone your internal resources. Here are just some of the ways you can use ATG Campaign Optimizer to serve the bottom line:

- Increase sales by delivering the right product and offer to the right customer
- Save on time-to-market for new products, promotions, and creative
- Reduce the costs associated with creating and running promotions
- Build customer loyalty by providing a more satisfying online experience
- Test product concepts before incurring development or manufacturing costs
- Build better relationships with dealers or suppliers by providing better research data on customer response to new products

Test the message and the messenger – your Web site.

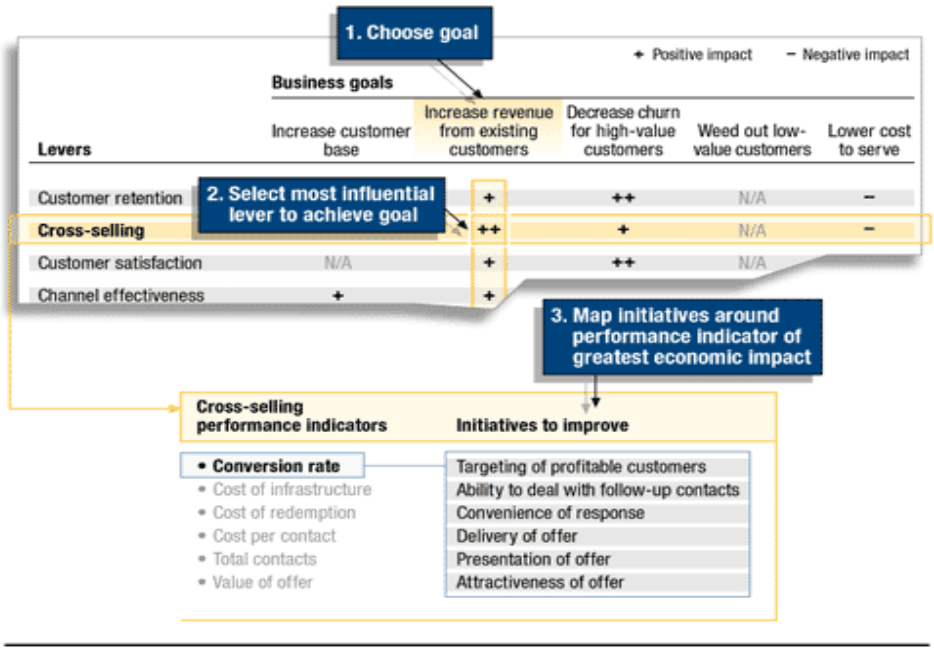
One of the most important capabilities of ATG Campaign Optimizer is the ability to test the effectiveness of your Web site itself. The placement of different variables on a page can have a significant impact on their effectiveness, as can ease of navigation, click-through, search, check-out and numerous other factors. ATG Campaign Optimizer helps you create the most compelling shopping experience by testing any and all of these variables alone or in combination with specific campaigns.

ATG Campaign Optimizer was developed to optimize your existing e-business environment. The software is tightly integrated with ATG Adaptive Scenario Engine, so no time-consuming integration is required. And you can reduce dependence on IT staff without fear of compromising Web site integrity.

According to a recent McKinsey Quarterly article, the basic process for setting and troubleshooting goals is the same for any company. The exhibit below illustrates some possible business goals and ways of achieving them. To capture more revenue from existing customers, for instance, your company might select four levers. From experience your company knows that the best way to get more money from its existing customers is to cross-sell products, such as accessories or extended service plans. Assuming this is a priority for your organization, the business and IT team would specify the performance indicators of successful cross-selling campaigns and work out their cost and potential effect. To simplify the campaign, the team would then pick the indicator with the greatest economic impact—in this case, increasing the conversion rate: the number of offers to customers that your company can turn into an additional accessory sale or extended warranty purchase. Economic analysis might show, for example, that a 10 percent increase in the response rate to a marketing effort could raise revenue by as much as 60 percent, whereas a 10 percent increase in the number of customer contacts would raise it by only 10 percent. Improving the conversion rate itself would entail several supporting initiatives: more precise targeting of prospective buyers, more attractive terms, or a better follow-up system.

EXHIBIT 2

Pick your goal, and the lever to achieve it



Perform marketing and site design experimentation, using a combination of ATG's Campaign Optimizer tool and web site analytics. Campaign Optimizer offers the ability to split test (or A/B test) two or more promotions, products, advertisements, or pieces of content simultaneously to measure there effectiveness and track user navigation habits. With A/B testing, business users would be able to compare results of different promotions being exposed to different audiences at the same time, thereby eliminating confounding temporal elements such as seasonality, traffic fluctuations and current events.

Using this campaign selection process, ATG's Campaign Optimizer can be used to allow your company to present different content to different customers and measure the resultant purchase behavior. You could build a web-based business UI that supports the creation of tests and a complete HTML reporting view for each test. With the ability to test and evaluate different site features, promotions or messages, business users will be able to generate actual customer results for a site change that is under consideration. Concurrent testing in ATG Commerce will provide the following broad capabilities:

- Ability to create, manage and review A/B treatment tests
- Ability to compare sales behavior of multiple test audiences
- Use of a standard template to compare different user populations (potentially applicable to other campaign programs – such as email)

Move to a rapid implement, review and refine site marketing process (rapid prototyping lab) and away from over analyzing and over engineering. There's no reason for guessing what kinds of site changes will improve revenue and conversion rates, or for guessing how people are actually using the site when these can be measured with real actionable data. Identify 3-4 high-impact system requirements, including the implementation of propensity-modeling tools to predict who would be the most responsive high-profit customers and create an on-line history of the offers made to them. For instance, which products have been most successful as the primary promotion on the home page? How does changing or displaying a different page in the checkout process help to increase or decrease conversions?

3.6.9 Post Implementation Review

Best Practice

Project team members and business users participate in a review session with ATG to ensure that all of their open issues are addressed. Also discuss recommendations for improvements and future enhancements to the system. The role of ATG in the post implementation review is to assist in the resolution of these open issues.

ATG recommends that you maintain both a development test and a staging environment in addition to the production environment. A staging environment is essential for the testing of new releases, patch releases, and code updates provided by ATG. Establish a procedure to test changes to the software prior to migration to the production environment. The staging environment should simulate the production environment as closely as possible. It is imperative that the configuration replicates the production environment to ensure that the test results are accurate.

ATG suggests that you monitor system performance over the first few months of production, including both interactive response times and back end system processing times. As databases and transaction files increase in size monitor the impact on processing. Typically, performance tuning exercises are most productive after the system has "settled down" and average run time and response time information is accurate. At that time you will have sufficient statistical information to identify potential problem areas.

When you evaluate your site's performance, one of the most important measures is the number of dynamic page builds per minute. The pages per minute count gives you an understanding of the throughput. You can view a running average in the Performance Summary page of the ATG administration interface, at:

http://localhost:8830/atg/dynamo/admin/en/performance.jhtml

In addition to pages per minute, the Performance Summary page also shows server latency (the amount of time the server spent handling each request) and other information such as the number of current sessions. The Performance Summary page provides links to other administration interface pages:

- Internal HTTP Server

- DRP Server
- Performance Monitor
- Servlet Pipeline
- Session Manager

You can also view performance statistics, though in a less legible form, through the ATG Control Center. Start or refresh the /atg/dynamo/admin/en/PerformanceInfo component to get current statistics.

As the users become more familiar with the system, they may find certain procedures cumbersome or difficult to perform. The project team should be available to work with the users and to listen to their suggestions and feedback on the day-to-day use of the system. When problem areas are identified, the project team must be prepared to evaluate the situation and implement corrective actions.

The integrations must be reviewed very carefully by the users to ensure proper treatment of all transaction types. The users must gain confidence in the system and ensure that no transactions are being lost or dropped. A successful reconciliation helps to build that confidence.

Ongoing application maintenance (upgrades, reporting, performance tuning, customizations, etc.) can be performed by ATG's Professional Services Group. Please contact us at 1-800-RING-ATG, we would like to help.

The ATG product catalog is composed of related items that form an organizational and navigational framework, enabling customers to locate and purchase items. A product catalog is usually built from a hierarchical tree of categories and products. The ATG Control Center (ACC) is used to create and modify a catalog for your commerce site.

Designed and built for all the key constituents within your organization, the ATG Control Center provides a single point of access for all of the development, administration, content, and business functions necessary to run your e-business. It provides an interface that combines the functionality your team members need into one Java application. By providing a collaborative environment, the ACC reduces your project overhead and development time.

Using the ACC, business managers can work with your IT team to develop, deploy, and analyze the key personalization, scenario, and commerce initiatives of the Web site. Using built-in reporting and analytics, they can generate detailed metrics to fine-tune campaign effectiveness.

The ACC is an easy-to-use environment in which business professionals can drive campaigns. They can manage content and profile attributes, define business rules for targeted content, and create promotional strategies. More importantly, managers can validate their work by testing the rules they've created by "impersonating" a particular user with specific profile attributes and viewing parts of the application as that user. They can also cross-reference scenarios and view reports to help manage large numbers of active scenarios and avoid potential conflicts.

The ACC is designed to be used on the business manager's desktop and can be used to manage ATG applications running on remote servers. The ACC also provides a console for system administrators to view and monitor server configurations and performance. The ACC component structure allows administrators to configure ATG system attributes, such as load balancing and cache settings.

Deleted: .

The ACC Security Management Interface enables administrators to manage ATG user accounts and privileged information with minimal programming. Using the ACC, they can manage persistent user accounts, look up user identities and associate them with roles, manage access control lists, and tie together multiple security systems against the same user account databases and/or authentication mechanisms.

The developer area of the ACC is an interface for the quick creation of components and pages for ATG projects.

The ACC component browser presents the property values of existing components and allows them to be changed, even in a running application. Developers can manage both objects and their Java classes.

The ACC includes page templates to quickly build your ATG site. Content managers and page developers use the ACC to build dynamic Web pages using JSPs, JavaBeans, and JHTML. A context-sensitive editor and component palette simplify page design. If designers prefer to author pages in a different environment, those pages can be imported to the ACC for translation to JHTML (if desired). This achieves simplified, centralized management of site design elements.

The ACC provides access to all content items (all types of electronic media) for your sites stored in searchable content repositories. Content in these repositories is managed, grouped, and accessed for high performance, personalized delivery through multiple channels, including Web sites, e-mail, and wireless devices.

3.7 Services Support Transition

3.7.1 Overview

After live you will continue to interact with many areas of ATG Software, all dedicated to your success. From Customer Support to our Electronic Customer Support facilities, ATG is committed to our ongoing relationship with you and helping you maximize the return on your software investment.

The main goal of the ATG Customer Support Services Centers is to deliver worldwide support services that provide the highest level of customer satisfaction and loyalty. ATG Technical Support works with you to resolve issues you may encounter during your support and maintenance period—helping ensure your success. For specific information about offerings and services available through ATG Technical Support please refer to the following sections of the ATG Product Support Users Guide. Make sure you familiarize yourself with the support program that has been secured by your organization:

3.7.2 ATG Support Offerings

Premium Support

Premium Support offers an enhanced level of service that includes priority product and technical support from our ATG Technical Support Centers and proactive personalized service from a dedicated Technical Support Account Manager.

- Round the clock access to our expert Support Engineers skilled in troubleshooting, problem diagnosis, and identifying resolutions
- Dedicated Technical Support Account Manager (TSAM)
- Priority status for bug fixes and enhancements
- Enhanced response objectives
- Major and minor product upgrades to your ATG Product Suite
- Membership to eSupport Services
- Subscription to ATG Technical Bulletins

PREMIUM SUPPORT HOURS OF OPERATION		
Access to ATG Technical Support 24 hours a day, 365 days a year		
	Monday through Friday, excluding holidays	9am to 5pm customer's local time for cases via phone or email
		5pm to 9am customer's local time for cases via email
	Weekends and holidays	Via pager for critical issues: Severity 0 and Severity 1
	8 business hours	Not Applicable

Technical Support Account Manager (TSAM)

As a Premium Support customer you can choose to have an ATG Technical Support Account Manager assigned as a dedicated resource to facilitate communication at a technical level between your ATG project team and the ATG Technical Support Group. Your Technical Support Manager will establish a relationship with the primary technical contact within your organization to gain a deeper understanding of your infrastructure and facilitate quicker responses to your inquiries. In addition, the Technical Support Account Manager will escalate and monitor any necessary issues within the ATG Technical Support Group and will communicate to you any changes or enhancements to ATG Technical Support offerings.

Priority Status and Enhanced Response Objectives

ATG will give top priority to the technical support needs of our Premium customers including response times for support cases, product enhancements, and bug-fix requests. You can always count on the fastest reply possible because your issues will receive higher priority at ATG.

Major and minor product upgrades

Upgrades available for download via the ATG web site. Upgrade announcements available through your Technical Bulletin subscription.

Premium Support Response Objective Goals

PREMIUM SUPPORT RESPONSE OBJECTIVE GOALS			
Severity	First Response	Work Around	Solution
Severity 0	Within 1 hour of logging the problem	48 hours	3 Business Days (To reduce to S1 Priority)
Severity 1	Within 2 business hours of logging	5 business days	30 Business Days
Severity 2	4 business hours	On mutual agreement	On Mutual Agreement
Severity 3	4 business hours	Not Applicable	Not Applicable

eSupport Services

Specialized membership to eSupport Services gives you access 24x7 to the ATG's online support center including:

- Bug Database and workarounds
- Product documentation
- Online case reporting
- Product Release Notes
- Knowledge Base Articles
- Troubleshooting and Performance Guides
- Frequently Asked Questions
- Archived Technical Bulletins

3.7.3 Standard Support

Standard Support provides access to technical assistance Monday through Friday, 9a.m. to 5p.m. Customer's local time, product upgrades and patches, as well as a login to our comprehensive online eSupport Services. International offices are staffed by Support Engineers who speak many of the major local languages and understand your local requirements.

- Monday through Friday, 9a.m. to 5p.m. Customer's local time access to our expert Support Engineers skilled in troubleshooting, problem diagnosis, and identifying resolutions
- Major and minor product upgrades to your ATG Product Suite

- Response objectives to meet your business needs
- Membership to eSupport Services and subscription to ATG Technical Bulletins

Standard Support Hours of Operation

Access ATG Technical Support Monday through Friday, excluding holidays (yearly holiday dates posted on support website www.atg.com) 9 A.M. to 5 P.M. Customer's local time for all cases

STANDARD SUPPORT HOURS OF OPERATION		
	Monday through Friday, excluding holidays	9am to 5pm customer's local time for cases via phone or email

Major and minor product upgrades

Upgrades available for download via the ATG web site. Upgrade announcements available through your Technical Bulletin subscription.

Standard Support Response Objectives Goals

STANDARD SUPPORT RESPONSE OBJECTIVE GOALS			
Severity	First Response	Work Around	Solution
Severity 0	Within 2 business hours of logging	2 business days	5 Business Days (To reduce to S1 Priority)
Severity 1	Within 4 business hours of logging	10 business days	30 Business Days
Severity 2	8 business hours	On mutual agreement	On Mutual Agreement
Severity 3	8 business hours	Not Applicable	Not Applicable

eSupport Services

Specialized membership to eSupport Services gives you access 24x7 to the ATG's online support center including:

- Bug Database and workarounds
- Product documentation

- Online case reporting
- Product Release Notes
- Knowledge Base Articles
- Troubleshooting and Performance Guides
- Frequently Asked Questions
- Archived Technical Bulletins

3.7.4 Connectivity with ATG Technical Support

Contacting ATG Technical Support for Assistance

HOW TO CONTACT ATG SUPPORT			
Email	support@atg.com		
Web	Http://www.atg.com/support		
Direct via ATG Software	ATG ACC in Versions 5.5 and greater		
Pager	For Premium Support customers only – please contact support for pager information		
Phone	Americas, Canada and Mexico	1–800–RING–ATG Press 4 for support	+1 617–386–1911
	Europe, Middle East and Africa	00–800–2856–2856	+44 118–956–5119

First Technical Response

Our goal is to provide a First Technical Response from an ATG Technical Support Engineer within our defined objectives. We make every reasonable effort to decrease your time to resolution by providing a problem resolution or solution, suggestions for workarounds or alternative solutions, questions seeking to clarify the problem description and/or a reference to our on-line technical support resources. If we need to do further research we will also let you know within this time.

In the case of Severity 0 or 1 issues, the goal of the First Technical Response is to immediately begin work to restore the site to operation. In most cases this can be accomplished by restarting the ATG server. Once the site is restored operationally, the support engineer will work to identify the problem and provide an acceptable workaround or solution until the problem is fully resolved.

Case Tracking

ATG tracks all cases for every question or request for assistance. A case is defined as a single technical question or instance of a problem. Since each case receives a tracking number please reference this number in all correspondences with ATG. Include in your subject line for all emails

Case Prioritization

We strive to achieve the highest service level possible and match our response times to your needs based on your specific business concerns. In order to help us more clearly identify the nature and extent of your issue and impact to your working environment we have instituted a simple system of issue priorities. Please review the following list and remember to indicate the priority of your issue when you contact ATG Technical Support by web, email or telephone.

Please use the following classifications to identify and prioritize your incoming support requests to ATG Technical Support:

SUPPORT TICKET TYPES

Severity	Priority	Definition	Examples
Severity 0	Emergency	Software is not functioning or is severely impaired	– Production server down and is unable to be restarted – Production server experiencing severe performance degradation – Loss or corruption of data on a production server
Severity 1	Critical	Software is operational, but its functionality is seriously affected.	– Production server is experiencing frequent failures – Production server is experiencing noticeable performance degradation – Development is halted
Severity 2	High	The software is useable, but development or production is impacted. This is the default classification for all cases reported to Technical Support unless otherwise indicated by you.	– Development/production can continue for a reasonable amount of time before the problem becomes critical – A workaround is available and acceptable
Severity 3	Normal	The software is useable, but you have a "how-to" question or would like to submit an enhancement request for our product management team.	– Ask a question about documentation – Clarification on a published workaround

Escalations

There are occasions where your business needs or the specific details of the support for a problem require you to ask for a technical case to be escalated. We have designed the process to be simple. If at any time you wish to escalate a case, contact us via phone or email, tell us the case number, and indicate the reason for the escalation. The case will be marked as escalated which automatically initiates our escalation procedures.

There are several actions that occur once you escalate a case.

- The Support Technical Lead and Manager are notified (and TSAM if applicable).
- A quick review of your business needs and technical case is done and an action plan constructed to quickly progress your case toward resolution.
- Communication of the action plan is discussed with you including deliverables and timelines.
- If the communication is not acceptable you may ask to speak with the Manager, Regional Director or Vice President of Technical Support.

First level: Manager of the Customer Support Center

Second level: VP of Customer Support Services

Types of Questions

ATG Technical Support can assist you on questions related to:

- Product installation and configuration for current supported platforms.
- Assistance with use of documented APIs and examples.
- Assistance with product usage and user interfaces.
- Assistance or guidance in troubleshooting application errors.
- Remote diagnosis and debugging as deemed necessary and appropriate by ATG support engineers.
- Escalation of issues to engineering/specialists as mutually agreed to by ATG and the client.
- Emergency onsite support as deemed necessary and appropriate by ATG and the client. (Additional charges apply.)
- Unqualified patches as mutually agreed to by ATG and the client.

Please contact ATG Professional Services for questions related to:

- Extending or reviewing application functionality via APIs
- Modifying samples, or source code for a specific purpose
- Writing custom/sample code
- Pre-release support for third party products

Feedback To ATG Support

ATG Customer Support Services measures its success by your satisfaction. Feedback is welcome at anytime and can be sent to us via:

- Email support-feedback@atg.com
- Complete a quarterly survey sent automatically upon close of one of your technical support cases
- Contact Technical Support management by calling ATG Technical Support

4 Appendix

4.1 Risk Assessment Table Of Contents

4.2 Process Factors for Large ATG Sites

atg Best Practice

4.2.1 Introduction

Ultimately, the performance of a large ATG site is a function of many factors, and poor application design or problems in the deployment environment have the potential to overcome product and platform strengths. In this section, we have summarized some of the process and organizational guidelines as well as technical best practices that will ensure that the application you build and deploy is based on proven best practices and delivers the best possible performance.

4.2.2 Project Management & Overall Process Factors

Incorporating ATG Expertise in Project Processes

It is clear from a large body of experience that including staff with expertise and experience with ATG itself in the project is an important factor in ensuring the best possible design for performance. ATG partners and ATG itself have this experience, and ATG's own deep product expertise will be applied to the project at critical junctures.

ATG is capable of helping in several ways to insure that your application is as well designed as possible. Our services plans include direct involvement of ATG architects. Our architects have been involved in a wide range of application development efforts, both as mentors and designers. Drawing on their deep experience, and direct access to ATG's product organization, their assistance can be used on a micro level to address individual points of concern, on a macro level to help you assess your whole effort, or on a mentoring and knowledge-transfer basis.

ATG's team also includes ATG Deployment Specialists, who have extensive experience working on some of the largest commerce sites on the Internet. Their skills help you prepare the deployment environment, including the ATG configuration and the underlying platform, and anticipate and defuse potential deployment issues.

In addition to people, knowledge is available in several other formats from ATG. Specifically, ATG provides deployment best practices both through the Administration Guide Documentation and a separate Deployment Checklist. Combined, these two documents insure that you cover the necessary steps to bring your site live, and that you are able to think through your deployment.

4.2.3 Organizational and Process Factors for Effective Personalization Use

Organizational Design

Scenario technology brings the ability to develop and scrip customer experiences much closer to the marketing staff. The result is a great deal of potential, but a requirement for thinking through the right way to manage the Scenario design and development process.

One recommendation is the formation of a new role, Scenario Managers, who operate in much the same way as Content managers do, in that they have expertise with the tool required to create and place Scenarios.

Scenarios do not need to be dedicated to the role, and in practice are usually not. They typically sit in the marketing groups responsible for site content and customer experience, and they require moderate technical aptitude, akin to that required for use of reporting and queuing tools or other scripting tools.

Incremental Implementation

All of the organizations that have made effective use of Scenarios are finding that an evolutionary, incremental approach leads to success. It is neither effective nor efficient to try to develop the comprehensive universe of Scenarios for all segments and all potential experiences at the outset. Proceeding up the learning

curve through initially narrow efforts with specific high-value segments and campaigns will prove the technology, increase the organization's understanding of the tool, and result in the highest return on investment for the effort involved to define, plan and implement Scenarios.

A particular guideline is to establish the goal of using at least some Scenarios in each rendition of the application, ensuring initial momentum and effective incremental growth in the use of the tools.

An initial set of Scenarios can be determined by a combination of cost/benefit analysis, alignment with strategic objectives, and organizational factors such as alignment with advocates in the business organization.

Education

Establishing a model for the marketing organization that reflects the capabilities of Scenarios is essential. Without such a model in place, the technology will tend to be relegated to uses that reflect a simplistic and stifled view of what the technology can do. We strongly advocate a comprehensive program of both education and training. The difference to us is that education is focused on the understanding of the WHAT's and WHY's of Scenarios, and training on the HOW's. Both are essential, and training without the education and understanding will fall short of expectations. ATG Professional Services should also be incorporated to allow your organization to take advantage of our experiences.

Effective Design & Specification Artifacts

The design of Scenarios is best accomplished by a joint development approach that includes a team that identifies, prioritizes, designs and documents planned Scenarios. This approach, often handled in a workshop setting, ensures that seams between conception, definition and development are minimized. In addition, an effective specification and documentation tool for Scenarios is required to capture the desired functionality as well as the required architectural components (content items, triggers, events, profile groups). ATG has developed such a tool, which we call Scenario Models. Whether Scenario Models are specifically used or whether you adopt a customized version that aligns with internal documentation standards, experience has proven that there must be an explicit tool to capture and document the desired functionality of Scenarios.

Integrate both business and technology stakeholders deeply into the Scenario design effort When developing a customer-driven Web application, internal stakeholders from both business and technology functional areas should have input and involvement. The business and technology stakeholders should share ownership of the Web initiative. Business stakeholders should be held accountable for the business vision, prioritizing initiatives based on business value, and ensuring that the Web application is aligned with the overall business strategy.

The technology stakeholders need to be responsible for determining the best way to implement the needs of the business. In so doing, the technologists should be prepared to assess the "difficulty" of a specific initiative and assist the business in prioritizing their needs not only on perceived business value but also on the ease of implementation – or investment required.

To facilitate the communication between the business and technology teams, your organization should develop a shared, common vision and language with which to communicate between teams. Business and technology stakeholders need to be aligned in their efforts, and a shared framework should be used to communicate business requirements and technical designs.

Take a life cycle approach to Targeter & Scenario design, development and Deployment

- Identify Targeter and Scenario components during the Design phase. Effective Targeter and Scenario use is predicated on having the architectural components (content, segments, and events) that Targeters and Scenarios call upon in place in the deployed application. This means that Targeter and Scenario design work should occur during the general Design phase of the development effort.
- Implement Targeter and Scenario components during the Development phase. With Design work complete, the architectural elements required by representative Scenarios should be developed into the general architecture.
- Evolve Targeter and Scenarios over time after deployment. Measure effectiveness; make adjustments based on results; manage concurrent Targeters and Scenarios; add new Targeters and Scenarios

4.2.4 Best Practices for Personalization & Scenarios

Focus Complex Scenarios on the Most Important Segments

When first presented with scenarios, some development teams are a bit overwhelmed by the potential breadth of their application. With scenarios, one could literally provide a completely custom experience for every individual visitor. Except in very particular circumstances, this approach is unlikely to be fruitful. Based on our experience, ATG advises that customers start by examining where the benefits of improving the visitor experience will pay the highest dividends. For example, consider the goal of increasing order size. To focus the problem, start by working on cross selling of your very high margin products, such as extended warrantees, instead of low margin goods such as batteries. Focus more by starting with expensive items that frequently yield large warrantee revenue (i.e. computers) instead of low cost items such as computer disks. Furthermore, focus on the segment of your audience that is likely to be receptive to such an offer. Some customers will never buy an extended warrantee under any circumstances and others will always consider them. If a customer never looks at the terms of an extended warrantee even after multiple offers, he is unlikely to buy one (in fact, he may become annoyed by the system's persistence to send offers he does not want). Focusing your efforts on the people who are most interested in an offer will drive better results, and have a positive side effect of improving the end customer experience of both warrantee buyers and non-warrantee buyers. So, start by designing your campaigns around activities that are likely to maximize the return on your marketer's time, development resources and system bandwidth, which are all scarce resources (no matter how easy scenarios makes the process.

Use Sampling for Intensive but High Value Operations

Some scenarios will be more computationally intensive than others, even when narrowly focused. For example, when launching a campaign for a new product category, you might want to carefully monitor and record every step of the transaction for that category to determine whether or not people are buying and where they are defecting from the purchase process. Scenarios are capable of collecting a wealth of data for off-line charting and analysis. Although the scenario server writes data in large batches, significant database traffic can still be generated, especially if the recorder is triggered frequently. Recording extensive information for every visitor coming to your home page will be resource intensive, but will also yield valuable insight into the viability of the new product category and the response rate of various campaign parameters. In this case, we recommend sampling the data (rather than collecting all of it) by configuring the recorder to:

- Store X% of received events, where an administrator sets X, or
- Store all received events for Y% of all sessions, where an administrator sets Y.

For empirical data, the performance of the site navigation recording was assessed while recording every event, and also while recording only 20% of them. In both cases 540 concurrent users were supported. It can be seen that reducing the sampling frequency improved page latency and reduced database contention.

	Page Throughput (Pages/Sec)	% Gain	Average Page Latecency (ms)	% Gain
Store 100% received events	39.22	n/a	0.536	n/a
Store 20% received events	40.22	2.5%	0.373	30.4%

Structure Your Scenarios Well: Filter out as Many People as Possible Up Front

As described in previously, scenarios act as a channel or pipe through which users flow. At the start of the scenario, the set of all users (whether currently active or not) are in the pipe. Condition elements (such as "in group Young") narrow the set of subjects passing through the pipe; only those subjects who satisfy the condition can proceed further. When possible, use condition elements to limit the group of people moving down the scenario as early as possible.

As subjects move down the channel, they encounter scenario elements that affect their progress in various ways. Events (such as "Registers") prevent the subject from proceeding further until the event occurs. Condition elements (such as "in group Young") narrow the set of subjects passing through the pipe; only those subjects who satisfy the condition can proceed further. Action elements (such as "Give Promotion") carry out some action (usually relative to the subject). Finally, the pipe may fork, in which case subjects flow down multiple branches in the fork, which diverge and then rejoin. Avoid scenarios that never end Consider the scenario:

- User Logs In
- Page Visit myPage.jhtml
- sendEmail

The user can log in multiple times, but never view the page, and therefore the scenario remains indefinitely. Use a branch to end the scenario (shown below) or use a wait element to prevent this problem.

Non Repeating scenarios

Non-repeating scenarios remain persistent in the last state of the scenario. If a timed event does not remove the scenario, it will remain indefinitely.

Scenario Consolidation

Consolidate scenario processing when possible to improve performance. For instance, the following two scenarios could be consolidated into one.

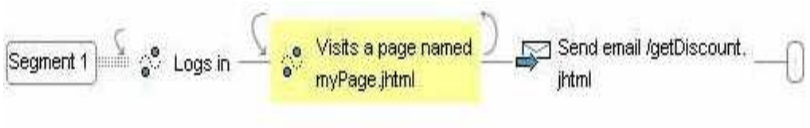
Scenario 1: Visits page x.jhtml -> record event -> end

Scenario 2 Visits page x.jhtml -> fill slot -> end

Consolidated:

Visits page x.jhtml -> record event -> fill slot -> end

Repeating events must be given careful consideration because they can cause multiple individual scenarios to be created for the same user. Consider the following scenario. Each page visit to myPage.jhtml will create a new scenario for the user.



4.3 Detailed Requirements Specification Outline

The following is an outline of a commerce specification that you can use to structure the content of your organization's e-commerce requirements specifications:

1 Introduction

- 1.1 Company Profile
- 1.2 Internet Sales Channel Background
  - 1.1.1 Internet Sales Channel Goals
  - 1.1.2 Web Sites Covered by Completed Design Document
- 1.3 Design Document Purpose
  - 1.3.1 Document Goals
  - 1.3.2 Related Documents
  - 1.3.3 Change Control
- 1.4 Scope
  - 1.4.1 Functionality
  - 1.4.2 Status
  - 1.4.3 Target Audience
- 1.5 Assumptions
  - 1.5.1 Integrated Footprint
  - 1.5.2 Shared Services

2 Administration

- 2.1 Overview
- 2.2 Business Requirements Addressed
- 2.3 Assumptions
- 2.4 Open Issues

3 Architecture

- 3.1 Business Architecture Overview
- 3.2 Business Requirements Addressed
- 3.3 Technical Architecture Overview
- 3.4 Architecture Guiding Principles
  - 3.4.1 Use of Industry Accepted or Defined Patterns When Available3
  - 3.4.2 Leverage of Existing Technology When Appropriate
  - 3.4.3 Followed Service Level Process Design
- 3.5 Logical Architecture
  - 3.5.1 Web Presentation Layer
  - 3.5.2 Application Layer
  - 3.5.3 Database Layer

4.4 JSP Coding Best Practices

4.4.1 Introduction

JHTML, or Dynamo Server Pages, is ATG's proprietary dynamic page compilation technology that allows you to embed dynamic elements to your web front end. J2EE, on the other hand, uses JSP as the dynamic page generation model. The JHTML2JSP converter demonstrates a rather simple mapping from JHTML to JSP using the dsp and core tag libraries. Unfortunately, in the transition from using JHTML to using JSP, many Dynamo developers forget the lessons and best practices they've learned while developing in JHTML. They've fallen into the habit of embedding Java in their pages, putting business logic in the pages, and handling user state logic within the pages.

JHTML makes it harder to do that. The predominant JHTML paradigm, the solution sets, the examples in the documentation, and the existing Dynamo development community all contribute to the commonly accepted JHTML design models as these patterns were instated when Dynamo's page compilation technology first came out.

When JSP first came out, developers saw the opportunity to make life easier for themselves by using embedded Java code. Recompiling servlets and beans became time consuming. Adding Java to the pages was more efficient. JSP lent itself well to furthering bad development habits.

Learning JHTML after learning JSP is harder - because JHTML is more structured. Developers are almost forced to create a clean JHTML design. On the other hand, developers who transition to JSP after using JHTML witness the Second Law of Thermodynamics first hand- all things in the universe tend toward chaos. It's much easier to unlearn the programming structure JHTML provides, and fall into the laziness of poor JSP programming.

JSP is not worse than JHTML by any means; however, it is more dangerous. Because it's less structured, it's easier to forget page design techniques. It requires more discipline to code properly in JSP. Lack of this discipline results in complex pages that can only be modified by developers. Pages that even other developers find difficult to understand. Pages that are difficult to enhance.

It's not difficult to apply standard Dynamo programming paradigms to JSP given the DSP and core tag libraries, which provide JHTML equivalent functionality to JSP. Developers just need to maintain the discipline to use the tools available to them, even when embedded Java is faster and more tempting to use.

Some of the following tips apply to any JSP development. Others are specifically targeted to using JSP on DAF.

1. Separate Visual Design From Programming Logic
2. Use Nucleus Components
3. Use FormHandlers or Droplets to Wrap Business Logic
4. Use an Appropriate Inclusion Mechanism
5. Use Standard HTML Page Design Techniques
6. Use Appropriate Exception Handling
7. If You Must Use Embedded Java...

4.4.2 Separate Visual Design from Programming Logic

When developers are first introduced to JSP, it seems easier to put all the programming logic in the JSP page. All the logic to generate a page is contained in one place. Making a change to the page or adding debug code doesn't require recompiling a class, restarting the classloader, and going through the steps to get to the same state you were in before the change. You just need to modify a file and hit "reload" on your web browser. At first, this would seem to decrease development time. This technique is useful for simple test cases or for small projects with small development teams with functionality that will never change. This style of coding quickly becomes unwieldy.

- Developers are required to make page design modifications.
- Page designers are required to know Java in order figure out what the page is doing.
- Pages become long and confusing, interspersed with both Java and HTML.
- The logic for rendering a page resides in the page, so there is less code reuse, even with included files.
- Because of this, a simple change can result in changing many pages. For example, you can define the title of the application within the page, and include a header that uses that title as the title of the page:

```
<% String title = "My Page"; %>
<%@ include file = "header.jsp" %>
```

Changing the title on an application-wide bases requires modifying all the files.

- Doing simple tasks is overly confusing. Imagine a page developer encountering the following code snippet:

```
<% if ( request.getParameter("state").equals("viewConfirmation") ) {
request.setParameter("previousState","orderConfirmation");
response.sendRedirect("orderConfirmation.jsp");
} else if ( request.getParameter("state").equals("cancel") ) {
%>
<%@ include file="cancelOrder.jsp" %>
<% } else { %>
<jsp:forward page="main.jsp"/>
<% } %>
```

- Depending on the user state, the next page the user sees will either be the order confirmation page, the front page, or the order cancellation page. Admittedly, this is a contrived example. Far worse code can be found in JSP code snippets - when pages become more complex, they become more difficult to understand and maintain. Then developers do stranger and stranger things to incorporate new functionality into the pages.
- There is no good way to change bean values between pages. The developer wants to change the user state depending on the link that the user clicks. The accepted means of doing this is through passing request parameters. Unfortunately, users can easily modify request parameters. A combination of javascript and POST forms with hidden input fields can make it slightly more difficult to alter, but it also increases the complexity of the page. This method also requires much more error checking, since you can't guarantee that the page will be getting the request parameter value that it's expecting.

By separating visual design from programming logic, the pages are much more maintainable. The Java components aren't tied to a specific page, and can be used by other parts of the application - you shouldn't be designing the code to work with the page, you should be designing code that can work with any page.

Remember to use the framework you learned when developing using JHTML: Nucleus components, droplets, formhandlers. And rather than using embedded Java, the core tag library consists of commonly used programming logic constructs (if/then/else, looping, null checks).

The single thing to remember when writing JSP code is to pretend you're writing JHTML, and modify the syntax for JSP. Because JSP makes it easier to implement bad page design, you need to be more conscious of your programming choices.

4.4.3 Use Nucleus Components

Dynamo uses the Nucleus namespace to provide a way to address and manage components, promoting reuse of objects and providing a way to link components. Components are typically JavaBeans with public accessor and mutator methods for private properties. A component may have a property that is a direct reference to another component. This is the usual method for delegation.

In Nucleus you can define a component's scope - global, session, or request, which is similar to JSP's application, session, and page scopes. This is a powerful mechanism for sharing and protecting data. Nucleus handles the creation of components, the initialization of their property values, and the resolution of references to other components.

Nucleus components are an extension of JavaBeans:

- Provide configurable initialization of JavaBean properties
- Allow linking of components
- Provide a component-based framework that is used in many aspects of the DAF, including form handling and Dynamo Servlet Beans ("droplets"), which are similar to custom tag libraries. These both minimize the amount of Java code that page developers put into their code, but droplets provide for selective output, depending on business logic.

4.4.4 Use Form Handlers or Droplets to Wrap Business Logic

By using form handlers or droplets to wrap business logic, you can use generalized versions of these components across JSPs, across other portions of the application, and across all your applications.

A handler method acts like the Controller in the Model-View-Controller design pattern, providing a wrapper business logic and basic error handling logic. Because there are typically a number of form handler components, the control logic is divided among them. From the server page, an HTML input element of type submit will set one of the form handler bean's properties, which will invoke that property's handler method. For example, a form handler could log in a user, and redirect a user to a passed in property, depending on a login success or failure.

The actual business logic can be within the form handler (or droplet) or in a separate service class.

Dynamo Servlet Beans (droplets) combine the standard J2EE servlet technology with JavaBeans. This enables a servlet to be treated as a component in Dynamo, which allows it to be configurable and to link to other components.

An advantage of Dynamo Servlet Beans over servlets is the ability to nest them in a Dynamo Server Page, inside open parameters. A Dynamo Servlet Bean can also invoke another Dynamo Servlet Bean. This expressive power encourages structured page design and modular (i.e., non-monolithic) servlet design. This is similar to custom tag libraries, but because droplets are a bit more flexible, there is never any need to embed any HTML within the droplet code. On the other hand, you often see HTML within custom tags.

There are also many pre-existing custom tag libraries available that you can use in your application to remove the business logic from the JSP.

4.4.5 Use an Appropriate Inclusion Mechanism

Taking out common JSP or HTML code and putting them into a single file allows you to make changes in only one location.

The include directive includes the content of a specified file the JSP source file. You should use the include directive when including straight text or if you have a common code snippet that you can reuse across multiple pages. Since you include the content of any file specified by the include directive before compilation, variables and other values specified in the calling JSP can also be utilized in the included content.

```
Syntax: <%@ include file="relativeURL" %>
```

The include action executes the specified JSP first; it then places the generated response in the calling JSP. Because the include action includes the generated response rather than the source content itself, variables and other values specified in the calling JSP are not available to the page included with the include action.

```
Syntax: <jsp:include page="{ relativeURL | <%= expression %> }" flush="true" />
```

The <dsp:include> tag is similar to the include action, except that it allows you to include files outside your servlet context and it allows you to pass Object parameters - not just String parameters - to the included page. In addition, paths in the included page are interpreted relative to that page. In the <jsp:include> tag, paths are interpreted relative to the enclosing page.

4.4.6 Use Standard HTML Page Design Techniques

Use Templating

A template mechanism allows for a common file to control layout. Then, when you want to change the layout, you need to modify only one file, and all the other pages will reflect the layout change. This helps in developing more maintainable code, and provides a consistent layout for pages within the application.

Stylesheets

Stylesheets enable page designers to place appearance control in a single location. Use Cascading Style Sheets (CSS) to control such items as font families, font sizes, and table characteristics.

HTML

To make your pages easier to understand:

- Use indenting to match tags
- Make sure you have valid HTML - some browsers are more forgiving if you forget a </td> tag, for example
- Comment your pages!
- Use HTML comments when you want the comment to appear in the HTML page source: <!-- -->
- Use Java comments when you want the comment to appear in the Java source: <% /* */ %>
- Use JSP comments when you want the comment to only appear in the JSP page source: <%-- --%>
- Remember that using JSP tags within HTML comments will require valid JSP

4.4.7 Use Appropriate Exception Handling

Users should be given useful information if an exception occurs during page processing. To handle runtime exceptions thrown in a JSP page, use the errorPage attribute to your page directive. Any runtime exceptions encountered while displaying the page will be available to this page in the exception object.

```
<%@ page errorPage="errorPage.jsp" %>
```

In general, your business logic should handle errors and exceptions gracefully - through a logging mechanism. If you need exceptions thrown up to the page, use the page directive to display a user-friendly message. When a runtime exception is encountered, the JSP engine will display the page defined in the errorPage attribute. This page must have the isErrorPage attribute set to true in the page directive.

```
<%@ page isErrorPage="true" %>
```

With this directive, you can use the exception object in this page. You can choose display a friendly error message and log the stack trace to a flat file or database. You can choose to ignore the exception, and tell the user to try again (not recommended). You can log the stack trace and redirect the user to another page.

In a Dynamo Application Server environment, exceptions may occur in a number of places:

- Within the servlet pipeline.
- While rendering a page (e.g., in a Dynamo Servlet Bean or a form handler component)
- Within a service component
- In a schedulable component's thread

It is important to determine which exceptions are important for the user to see, which exceptions should be hidden from the user, and which exceptions should be logged. Typical errors that the user should see result from invalid input submitted in a form, or from there being no items available in the inventory. Such errors would not be logged as errors (but could optionally be logged as debugging messages).

On the other hand, a system-level error (such as the database being inaccessible) should be logged in the error log in as much detail as possible, while the end user only needs to be informed that something is amiss. Here are some guidelines for how to handle exceptions within particular types of components.

Exception Handling by Components Used on a Page

As a general rule, classes at the top of the presentation layer should be designed to catch any exceptions that might occur, and display an appropriate message in the HTTP response. In other words, components used on a page (i.e., Dynamo Servlet Beans and form handler components) are generally responsible for handling exceptions that occur during request processing.

The corollary to this rule is that lower-level objects used by these components (e.g., business objects, services, data access managers) should throw exceptions when appropriate, and should allow certain exceptions to propagate up, rather than catching all exceptions and returning null values or empty lists. Among the hazards of returning null values or empty lists is that the rendered page may be degenerate (e.g., if a drop-down list contains no options, or a menu of links offers no items).

Underlying classes should, as a general rule, not overuse exceptions. The Java exception-handling mechanism is computationally expensive and, in a server environment, could degrade performance. Exceptions should only be used to communicate a truly exceptional condition.

Note that IOExceptions that are thrown when writing to the http response object should never be caught, and if they are, they should be re-thrown. This condition can happen if a user has pressed the stop button on their browser. If IOExceptions are caught, then threads will hang because the DRP server is never notified that the request has ended.

Exception Handling by Pipeline Servlets

Because the servlet pipeline is invoked before the page itself is processed, a pipeline servlet needs to handle exceptions differently than would a component on the requested page. Pipeline servlets must catch all exceptions, log an error message, and return false (thus to abort the processing of a request).

Again, note that IOExceptions that are thrown when writing to the http response object should never be caught, and if they are, they should be re-thrown. This condition can happen if a user has pressed the stop button on their browser. If IOExceptions are caught, then threads will hang because the DRP server is never notified that the request has ended.

Optionally, the pipeline servlet can issue a redirect (e.g., to display an error page as the response).

Exception Handling in Service Components

A service component should throw exceptions when appropriate, and should allow certain exceptions to propagate up, rather than catching all exceptions and returning null values or empty lists instead. Among the hazards of a returning null value or an empty list is that the calling method may not interpret the value correctly (or, worst of all, may attempt to call a method on the null object, resulting in a NullPointerException).

A service that constructs new threads must take care that the thread catches all exceptions, log an error message, and take any further action necessary to abort the offending request.

Exception Handling in Schedulable Components

The Dynamo Scheduler service executes a schedulable component in a thread outside the HTTP request-handling thread pool. Schedulable components must catch all exceptions and log an error message.

A component with a periodic schedule or calendar schedule can optionally disable itself, which may be appropriate in cases where the exceptional condition is severe or persistent and recovery seems unlikely. In such situations the component would disable itself (until either the component is restarted by hand, or the Dynamo server is restarted).

4.4.8 If You Must Use Embedded Java

There are some cases where embedded Java makes more sense than writing a separate class for the functionality you need. And if you need to use embedded Java, don't chain your method calls. Make sure you do error checking - NullPointerExceptions occur frequently in JSP pages.

If you try something like:

```
<img src="<%= env.getImage( request.getParameter("myId") ).getImageURL() %>"/>
```

JSP doesn't fail quite as nicely as JHTML does if there is no "myId" parameter. In JHTML, the equivalent implementation:

```
<img src="param:image[param:myId].imageURL"/>
```

would at worst return an empty string. The JSP will fail with a NullPointerException, and return a stacktrace to the user. You can use dsp and core tag libraries to perform appropriate checking or use the equivalent DSP Tag Library syntax.

4.4.9 Conclusion

If all this sounds obvious or familiar... it should. This is how you write JHTML pages. Remember your Dynamo coding guidelines, and they don't change because the dynamic page compilation technology is now JSP.

4.5 Performance Tuning DAS

4.5.1 Defining objectives

There are many ways that performance of a site can be defined. This is a set of guidelines on getting the data you need to clearly define the testing objectives and making sure the client understands what they are.

4.5.1.1 Client objectives

These are things that must be defined by the client. However, don't be afraid to argue with them if their expectations are unrealistic. If you start the testing process with ridiculous objectives, it will be very hard later to convince them that the problem is with the objectives, not the testing process or application.

4.5.1.2 Maximum number of users

The client should have an estimate of the maximum number of simultaneous users they expect the site to support. Note that this is not the maximum number that could ever appear at once, but the maximum number that should be supportable without a severe degradation of the user experience.

In the ATG world, this number maps to sessions, but that may not be true for the load testing tool. Some tools use a vocabulary at odds with the rest of the web world. That being the case, you should keep all discussions of this topic using the phrase "concurrent users" to make sure everyone is on the same page. Avoid "simultaneous users", as some non-technical people will assume that means a whole bunch of people hitting the submit button at exactly the same time. Just explain concurrency if you need to.

A reasonable maximum is 200*<# of instances>. That's assuming 40 DRP threads per instance, page responses less than one second, and an average request rate of one page every five seconds from the users. The real math won't come out this neat, but don't let a client force you to commit to more than this. Many sites do much better, but that's not always possible.

4.5.1.3 Server response time

This is the only response time that can be reliably measured, so it is important to define. At a minimum, you need the maximum average time that is acceptable. Even better is to break that time down by subsection of the site. For example, browsing a catalog is often significantly faster than either order checkout or account administration. If your goals can be defined with that in mind, you will have a much easier time meeting them.

The cleanest way to measure server response time is with the performance monitor. Since that data is not recorded in any way, it is important that you save off the HTML pages after every test run. Also, make sure you are recording the data on enough instances to rule out skewed results from a single instance.

A reasonable goal for server response times is less than one second on average. If a client requires more aggressive timings than that, they need to be prepared to possibly reduce functionality below what they really want.

4.5.1.4 User response time

This is the number that clients usually care about the most and is the number that load tools will supply. However, this number includes a lot of variables outside the direct control of the application, such as network latency and user bandwidth. Some testing tools can attempt to simulate different bandwidth limits on users, but their results are somewhat questionable. While it is fine to establish objectives for these numbers, make sure that any requirements are based on the server side numbers.

4.5.1.5 Minimum transaction throughput

Some clients will have a particular set of (usually time consuming) user behaviors that they particularly care about. Sometimes these are so important that they choose to define performance goals around the number of completed transactions in a period of time. This sort of goal is very bad, as it often depends on systems outside the control of the application tuning. Whenever possible, avoid having this sort of objective formalized as a

requirement. If it is a formal requirement, make sure responsibility for meeting it is spread across all the involved systems.

4.5.2 Tuner objectives

These objectives are things that the client cares nothing about. However, tuning towards these objectives will help you achieve the client's goals.

4.5.2.1 Reduce full garbage collections

You don't want full GCs to occur more often than every 10 minutes, at most. If you can manage to get them to occur at most every hour, then you've done the best you can possibly expect. You can't always get to the 10 minutes, but that should be your goal.

4.5.2.2 Increase generations

Increasing the number of generations in the survivor spaces is often the best way to reduce the number of full GCs. If your running application has four or more generations most of the time, then you are done. If it consistently has only one generation, then you need to keep tuning the survivor spaces until you can get at least two generations most of the time.

4.5.2.3 Reduce impact of minor garbage collections

There are essentially two ways to reduce the impact of minor GCs: spread them out or make them shorter. These options are essentially opposites, and which you pursue will depend heavily on the per request memory profile of the application. You spread out minor GCs by increase the size of the new space, but you make them shorter by decreasing the size of the new space.

4.5.3 Running tests

4.5.3.1 Initial JVM Arguments

This set of arguments should give you a starting point for tuning:

```
-Xms1024m –Xmx1024m –XX:NewSize=256m –XX:MaxNewSize=256m –XX:PermSize=128m –XX:MaxPermSize=128m –
XX:SurvivorRatio=16 –XX:DisableExplicitGC –verbose:gc –XX:+PrintTenuringDistribution
```

Keep in mind that with Java arguments, the last one wins, so these can just be appended to any existing arguments.

4.5.3.2 Test procedure

Any load testing tool can be used, as long as it lets you set up a realistic test. Here are some suggestions for assembling the best test you can.

- Write tiny test cases that exercise a single element of the site, such as login, adding something to the cart, or browsing around. Test each of these pieces in isolation (or as close as possible) to ensure that the test case functions on its own. For optimal results, you should have at least one test case for every page on the site.
- Wherever possible, write the test cases to use random data or pull data from the page. For example, if you need to browse a catalog, write that case to either be driven off a precreated list of ids, or else parse the page for navigation options.
- Assemble the test cases into scenarios that mimic real users. If an existing site is being replaced, use data from that site and adjust as necessary to support the new functionality. Otherwise, take a guess based off of experience with similar sites.

- Verify each of the scenarios individually, end to end. Make absolutely sure that the scenario is behaving as intended. Don't assume that because the tool reports no errors the site is functioning correctly, verify with the logs and the database that everything really works.
- Assemble the scenarios into a test suite with proper ratios for different kinds of users. Take client predictions of the future with a certain grain of salt. For example, some commerce customers will insist on a very high mix of order placement in the testing, even though a conversion rate of four percent is astounding.

4.5.3.3 Getting a baseline

In general, you want to start testing at well below your desired capacity goals. A good starting point is 10% of the most aggressive goal, if you have the goals defined multiple ways. From there, increase load until you find the point at which performance no longer meets the requirements. Rerun that test to verify results, and make sure that you save off all logs, including garbage collection information and performance monitor data for analysis by yourself and the development team.

4.5.3.4 Iterative testing for improvement

As you make changes that are expected to improve site performance, you should always restart testing at the last test. This will let you verify that you have improved things, and give you an idea of how much you should attempt to increase the load for the next test.

4.5.4 Finding bottlenecks

4.5.4.1 Application server bound behavior

The application server is bound primarily by CPU capacity. It is possible to have other problems, but on server class hardware they generally require a hardware failure of some sort, which should be detected directly.

4.5.4.2 Symptoms

- CPU is over 90% busy with higher spikes. This is the obvious case, where all CPU is being used. Depending on the OS and hardware, some modern CPUs are impossible to max entirely, so don't count on hitting 100% and staying there. At this point, average server side response times should start to stabilize, but client side times should continue to scale up with load.
- CPU is under 90% busy, but response times are behaving as above. Usually this is one of the other bounding problems below, but this can mean that the application is extremely contentious and the available tools are reporting that contention as idle time for the CPU. Usually the only way to differentiate this condition is to either eliminate the other options, or else notice that the pages with the worst times tend to be those with the heaviest CPU load, rather than the largest page size or most database queries, for example.

4.5.4.3 Solutions

- Use static includes where reasonable. Dynamic includes, both JSP style and DSP style, add several layers to the call stack and create many temporary objects. If you don't actually need the functionality they add, replace them with static includes. That means that the include will be resolved and processed at page compile time instead of at runtime and can save a significant amount of CPU for common includes such as page headers and footers. Don't get too happy with the static includes, though, or you'll run into the 64 KB limit on method size when compiling pages. It is best to only statically include page fragments that don't include any other fragments themselves.
- Make extensive use of the Cache droplet. The repository cache is a great database abstraction, but keep in mind that it does not abstract application logic. If complex logic goes into a piece of the display, the odds are good that it isn't the only piece of complexity, and those can add up. For most applications these days, memory limitations are not really an issue, so the best way to address that front end complexity is by caching the rendered segments. It's hard to make a general recommendation on usage, but be as aggressive with caching as the business requirements will allow. And don't be afraid to try something the business requirements won't allow if the site is in bad shape and needs the help. If you

can prove that a relaxation in the data freshness requirements will provide a dramatic increase in performance, many businesses are willing to make that choice.

- Use the Performance Monitor. During a load test, the performance monitor can be your best friend for finding out exactly which pages are functioning slowly. Usually the quick wins can be found by using static includes or the Cache droplet, but sometimes that isn't enough. Add some instrumentation to the custom code, if necessary, and work your way down the pile of information coming from the Performance Monitor to find the logic that is so costly. You can't always solve the problem, but at least you will be able to point to a specific business requirement as the source of the performance problem.

4.5.5 Network bound behavior

4.5.5.1 Symptoms

- Long "Send reply to DRP server" times in the Performance Monitor. This entry is horribly named, but what it represents is the time spent transmitting the response to the web server and waiting for an acknowledgement. If the average time for this is ever more than a few milliseconds, you are either CPU bound or have a network problem.
- User response times significantly longer than server response times. If the average server response time plus the average "Send reply to DRP server" time is not within a few tenths of a second of the average client response time, that can indicate a network bottleneck somewhere between the systems. The exception would be if the load test is deliberately simulating low bandwidth users.

4.5.5.2 Solutions

- Check network infrastructure. All interfaces should be explicitly set to 100 Mb (or 1 Gb if you are so lucky) full-duplex. That includes firewalls and switches. Also, firewalls and switches should be checked for packet collisions and other signs that they are bottlenecking themselves. Your network person will really hate you for this.
- Remove white space. You can drastically reduce the required bandwidth for the application by preprocessing JSP pages to remove all excess white space. Runtime stripping is far too costly... if you are going to remove white space, it must be part of the build process.
- Use efficient includes. Make sure your CSS and JavaScript files are as efficient as possible. Try not to define styles or functions in the page itself, to avoid downloading the same text over and over.
- Outsource static content. Anything fairly static, such as images, CSS, or JavaScript can be outsourced to someone like Akamai. They have fairly sophisticated procedures for dealing with updates that can usually be integrated into most processes. This is a big cost decision for a client, however, so it is unlikely that it will be implemented at the phase of a project when performance testing usually happens.

4.5.5.3 Database/backend bound behavior

Backend bottlenecks will sometimes be revealed by the Performance Monitor, but only if the relevant code has been instrumented. If you find yourself with unexplainable response times, try taking a stack dump. See if many of the long response times are from threads involved in communication with some other system.

Unfortunately, there is no standard set of solutions for these sorts of problems. The best you can do is familiarize yourself with the ATG side of the integration (make sure there's nothing stupid there), then take as precise a description of the issue as possible to the experts for the other system.

4.5.6 Possible optimizations

These are things that don't necessarily fit any particular problem area, but can be helpful in a general way.

4.5.6.1 JVM options

- Heap tuning. This is a fairly complex process, but when all else fails, you can sometimes get a lot out of time invested in tweaking the heap. Read up on all the options for the JVM you are currently using, as they change with every major version.

- Alternate garbage collection methods. In particular, parallel GC works very well on systems with more than one CPU per application instance. Make sure you read up on and understand the nature of each method, or you can be very surprised by the results.

4.5.6.2 ATG tuning

- Set DrpServer.acknowledgeSendReply=false. This flag defaults to true, which is a hedge against unreliable web servers. There is really no need for this on a modern site, since if you have unreliable web servers these days you have much worse problems than a few responses being lost.
- Set DrpServer.outputBufferSize to greater than your typical page size. Don't forget that the number is in bytes.
- Double check your repository cache statistics. If you have the memory, keep increasing cache sizes until the hit ratio is over 95%.Remember that for any cache (query or item) with a hit ratio under 50%, you are better off disabling the cache entirely, since most of the time you are taking the time to look in the cache and missing anyway.

4.5.6.3 Database tuning

- Don't forget to add indexes for custom tables. Keep in mind that an index on a rapidly changed table can cost far more to keep current than allowing the table scans on queries. That can mean that if you change the way an OOTB table is used, you may need to remove or alter an existing indexes.
- Don't be afraid of stored procedures. Repositories now have support for defined queries. If the generated SQL for a common operation is a bottleneck, replace it with a stored procedure and get results that way. Putting explicit SQL into the Java code is generally not a good idea, as keeping it current with changes to the database can be a hassle. A stored procedure, on the other hand, can be put in the same file as the table structures, making it much easier to keep them coordinated.

4.5.6.4 Operating system tuning

Operating systems have a huge number of parameters related to network operations. If you don't know a particular OS well enough to know what to ask about, just ask the system administrators if there are any settings that could be limiting the number of simultaneous connections or causing excessively long timeouts.

4.5.7 References

4.5.7.1 Java

- http://java.sun.com/docs/hotspot/VMOptions.html
- http://java.sun.com/docs/hotspot/gc1.4.2/
- http://www.hp.com/products1/unix/java/java2/hpjtune/

4.5.7.2 Load Tools

- http://www.mercury.com/us/products/performance-center/loadrunner/
- http://www.empirix.com/ecd/ecforms/process/ets-process.asp
- http://www.segue.com/products/load-stress-performance-testing/silkperformer.asp

4.6 ATG Reference Guides

4.6.1 Install

- Installation and Configuration Guide for ATG Application Server - Describes how to install and run ATG 7 applications on ATG Application Server. Includes information about server management, database configuration, session management, performance planning, and troubleshooting.
- Installation and Configuration Guide for BEA WebLogic Server - Describes how to install and configure ATG 7 applications running on BEA WebLogic Server.
- Installation and Configuration Guide for IBM WebSphere - Describes how to install and configure ATG 7 applications running on IBM WebSphere.

4.6.2 J2EE

- J2EE Development and Deployment Guide - Discusses all aspects of developing and deploying J2EE applications in ATG Application Server.

4.6.3 Development

- Page Developer's Guide - Provides an introduction to developing JavaServer Pages on the ATG 7 platform. Intended primarily for JSP developers, but should also be read by programmers who are new to the ATG product suite.
- Programming Guide - Presents a detailed description of Nucleus programming concepts for developers and other advanced users. Includes examples and reference information about developing applications with the ATG Application Framework.
- Web Services and Integration Framework Guide - Describes ATG 7 support for creating and calling Web services, to share data and business logic with external applications. Also describes other data integration tools in ATG 7.
- Personalization Programming Guide - Describes programming tasks for the ATG 7 Personalization and Scenarios modules. Includes information on setting up profile repositories, creating targeting rules and services, configuring scenario servers, and adding custom scenario events and actions.
- Personalization Guide for Business Users - Designed to help business users understand and work with the ATG 7 Personalization and Scenarios modules. Describes how to use the ATG Control Center to perform typical tasks such as segmenting site visitors, defining rules for personalizing site content, and using scenarios to create promotional campaigns.
- Repository Guide - Describes the ATG Repository API, the heart of ATG's Data Anywhere Architecture. Presents programming concepts for advanced users, including SQL repositories, LDAP repositories, secured repositories, and composite repositories. Includes examples and reference information to help programmers develop applications using the Repository API.
- Quincy Funds Demo Documentation - Describes how to use the Quincy Funds demo, a financial services Web site assembled from standard Personalization and Scenarios components. Also provides sample JavaServer Pages from the demo.

4.6.4 Portal

- Portal Administration Guide - Describes how to install and configure ATG Portal, create, edit, and administer user communities, and construct custom portal pages for those communities.

- Portal Development Guide - Presents concepts and guidelines for developing Web applications or "gears" that will appear within pages served by ATG Portal.
- Portal Demo Documentation - Describes the three ATG Portal demos - ROIMax (a business-to-enterprise portal), Geomatrix (a channel sales portal), and Krystal Communications (a wireless portal).

4.6.5 Commerce

- Commerce Administration and Development Guide - Describes how to use ATG Business Commerce and ATG Consumer Commerce to build and manage commerce sites. Intended for all ATG Commerce users, including store administrators, business users, page developers, and programmers.
- Business Commerce Reference Application Guide - Introduces Motorprise - the ATG Business Commerce Reference Application. Provides details about how the site is built using ATG Business Commerce main components and customized components. The site is intended to be used as an example or template for your own Web site.
- Consumer Commerce Reference Application Guide - Introduces the Pioneer Cycling store - the ATG Consumer Commerce Reference Application. Provides details about how the site is built using ATG Consumer Commerce main components and customized components. The site is intended to be used as an example or template for your own Web site.
- CSR Reference Application Guide - Describes the functionality and default configuration of the Customer Service Module, the ATG CSR Reference Application that can be run with either ATG Business Commerce or ATG Consumer Commerce.

4.6.6 Content Administration

- Content Administration Programming Guide - Describes how to set up and customize ATG Content Administration and its browser-based user interface, the ATG Business Control Center. Also describes how to deploy content to a production Web site. Intended for system administrators, developers, and page developers.
- Content Administration Guide for Business Users - Designed to help business users understand and work with ATG Content Administration. Describes how to use the ATG Business Control Center to create and manage Web site assets. Intended for content developers and editors.

4.6.7 Technical Reference

- Entity Relationship Diagrams - ER diagrams showing the default database schema and table relationships for the entire ATG 7 product suite.
- API Reference - Javadoc descriptions of the ATG 7 classes.

4.7 Supported Environments for ATG 7.0

ATG supports the deployment of ATG applications on the following J2EE application servers:

ATG Dynamo Application Server 6.3.0 patch 2
IBM WebSphere Application Server 5.1.1
BEA WebLogic Server 8.1 SP3

ATG recommends that customers deploy on platforms that have passed our internal qualification process. The product levels below have been tested and approved by ATG.
Because other products frequently ship fixes, updates, and new releases, we cannot test every possible configuration. Product versions which are minor point releases above those listed are fully supported by ATG. If you experience a problem with ATG software and a listed product version, or a point release above a listed version, please let ATG Customer Support Services know. We'll do our best to resolve the issue.
Please be aware that all support for products or their point version upgrades that differ from the levels indicated on this page is best-effort only. If an unsupported third-party product is determined to be the cause of any problem, then moving to a supported platform or contacting the third-party product provider will be required. Certain configurations of third party software may not be compatible. ATG recommends that you confirm support for all component versions of your configuration with any vendors that have provided those components. ATG supports some platforms to help customers evaluate its products and develop applications, but currently does not recommend or support their use in deployed production environments.

Supported Browsers for all Configurations

For Windows/PC environments: Internet Explorer 6, Mozilla 1.7, Firefox .9, Netscape 7.2

For Mac environments: Safari 1.2*

* supported with ATG DAS only, not supported with WebSphere or WebLogic

Supported Environments when Running ATG Dynamo Application Server

When deploying the ATG platform and applications on the ATG Dynamo Application Server, the following environments are supported:

ATG Dynamo Application Server 6.3.0 patch 2

Production Environments

Operating System [Hardware]
Java 2 Platform
Database Server[1]
HTTP Server[2]

HP-UX 11i [PA-RISC]
HP SDK for Java 1.4.2.05 (32-bit only)
Oracle 9i (v. 9.2.0.4)
Oracle 8i Release 3 (v. 8.1.7)
Microsoft SQL Server 2000 SP3a
IBM DB2 Enterprise Server 8.1, Fix Pack 5
Sybase ASE 12.5.2
Apache 1.3.29

Sun Java Sustem Web Server 6.1 SP1
(formerly Sun ONE Enterprise Edition 6.1)
Microsoft Windows Server 2000 SP4 [Intel]

Windows Advanced Server 2000 SP4 [Intel]
Sun SDK 1.4.2_05 (32-bit only)
Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

MS SQL Server 2000 SP3a

IBM DB2 Enterprise Server 8.1, Fix Pack 5

Sybase ASE 12.5.2

Microsoft Internet Information Server 5.0

Sun Java System Web Server 6.1 SP1

(formerly Sun ONE Enterprise Edition 6.1)

Red Hat Enterprise Linux Server AS 3.0 update 1 [Intel]

Red Hat Enterprise Linux Server ES 3.0 update 1 [Intel]

Red Hat Enterprise Linux Server AS 2.1 [Intel]

Red Hat Enterprise Linux Server ES 2.1 [Intel]

Sun SDK 1.4.2_05 (32-bit only)
Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

MS SQL Server 2000 SP3a

Sybase ASE 12.5.2
Apache 1.3.29

Sun Java System Web Server 6.1 SP1 (formerly Sun ONE Enterprise Edition 6.1)
Sun Solaris 8 [Sparc]

Sun Solaris 9 [Sparc]
Sun SDK 1.4.2_05 (32-bit only)
Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

MS SQL Server 2000 SP3a

IBM DB2 Enterprise Server 8.1, Fix Pack 5

Sybase ASE 12.5.2

Apache 1.3.29

Sun Java System Web Server 6.1 SP1 (formerly Sun ONE Enterprise Edition 6.1)

Notes

[1] ATG supports both the Standard and Enterprise editions of the Oracle databases.

[2] Dynamo Connection Modules support request handling communication between HTTP servers and Application Server. Connection Module support is operating system specific and independent of the Dynamo operating system support.

Additional Supported Environments for Evaluation and Development Only

The following environments are supported for Development only, in addition to all environments supported for Production.

Operating Systems

Java 2 Platforms

Database Server/JDBC Driver

Microsoft Windows 2000 Professional

Microsoft Windows XP

Red Hat Enterprise Linux WS 3.0 update 1

Sun SDK 1.4.2_05 (32-bit only)

Solid FlowEngine 3.7 with Solid JDBC 2.0 driver

All Supported Production Environments

OS specific Jikes™ and JRE

Solid FlowEngine 3.7 with Solid JDBC 2.0 driver

Database Servers and JDBC Drivers

Database support is independent of the database operating system. The corresponding supported JDBC driver options for each database are listed below.

Database Server[1]

JDBC Driver

IBM DB2 Universal Database v.8.1

FixPack 5

IBM DB2 UDB v.8.1 Type 2

MS SQL Server 2000 [Intel] SP3a

i-Net OPTA XS v. 6.02 JDBC driver for MS SQL Server

Oracle 9i (v. 9.2.0.4)

Oracle 9i: Oracle OCI and Oracle Thin Drivers 9.2

Oracle 8i Release 3 (v. 8.1.7)

Oracle 8i: Oracle OCI and Oracle Thin Drivers 8.1.7

Sybase ASE 12.5.2

iConnect 5.5

SOLID FlowEngine 3.7[1]

Solid 3.7: JDBC 2.0 Driver

Notes
[1] The Solid FlowEngine and JDBC driver are supported for evaluation and development-only purposes.

LDAP Servers (optional)

Dynamo can be optionally configured to use an LDAP repository to authenticate users and to authorize access.

LDAP Server

Supported Operating Systems

Sun ONE Directory Server 5.2[1]

HP-UX, Solaris, Linux

MS Windows 2000 Active Directory Server

Windows 2000

Notes
[1] ATG Dynamo LDAP Repository works on all supported LDAP platforms as a way to provide access to an LDAP server using a Dynamo Repository Interface. ATG Dynamo LDAP UserDirectory is only supported on Sun ONE Directory Server 5.2.

Supported Environments when Running IBM WebSphere Application Server

When deploying the ATG platform and applications on the WebSphere Application Server, the following environments are supported:

IBM WebSphere Application Server
WebSphere Application Server 5.1.1
WebSphere Application Server 5.1.1 Network Deployment
WebSphere Application Server 5.1.1 Enterprise

Production Environments

Operating Systems

Java 2 Platform

Database Server

HP-UX 11i [PA-RISC]

HP-UX SDK for Java v. 1.4.2

IBM DB2 Enterprise Server v.8.1 FixPack 5

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

IBM AIX 5L 5.2

IBM Developer Kit 1.4.2

IBM DB2 Enterprise Server v.8.1 FixPack 5

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

Microsoft Windows 2000 [Intel] Server SP4

Microsoft Windows 2000 [Intel] Advanced Server SP4

IBM Developer Kit 1.4.2

IBM DB2 Enterprise Server v.8.1 FixPack 5

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

Red Hat Enterprise Linux AS 3.0 update 1 [Intel]

Red Hat Enterprise Linux ES 3.0 update 1 [Intel]

Red Hat Enterprise Linux Server AS 2.1 [Intel]

Red Hat Enterprise Linux Server ES 2.1 [Intel]

IBM Developer Kit 1.4.2

IBM DB2 Enterprise Server v.8.1 FixPack 5

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

Sun Solaris 8 [Sparc]

Sun Solaris 9 [Sparc]

IBM Sun JDK 1.4.2
(32-bit)

IBM DB2 Enterprise Server v.8.1 FixPack 5

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

Additional Environments for Evaluation and Development Only

The following environments are supported for Development only, in addition to all environments supported for

Operating Systems

Java 2 Platforms

Database Server/JDBC Driver

Microsoft Windows 2000 Professional

Microsoft Windows XP

IBM Developer Kit 1.4.2

Solid FlowEngine 3.7 with Solid JDBC 2.0 driver

All Supported Production Environments

OS specific Jikes™ and JRE

Solid FlowEngine 3.7 with Solid JDBC 2.0 driver

Database Servers and JDBC Drivers

ATG supports the listed JDBC drivers in accordance with the application server vendor's supported configurations

Database Server

JDBC Driver

IBM DB2 Enterprise Server v.8.1 FixPack 5

IBM DB2 UDB v.8.1 Type 2

MS SQL Server 2000 SP3a

IBM WebSphere embedded Connect JDBC 3.1

Oracle 9i (v. 9.2.0.4)

Oracle 9i: Oracle OCI and Oracle Thin Drivers 9.2

Oracle 8i Release 3 (v. 8.1.7)

Oracle 8i: Oracle OCI and Oracle Thin Drivers 8.1.7

Sybase ASE 12.5.2

iConnect 5.5

SOLID FlowEngine 3.7[1]

Solid 3.7: JDBC 2.0 Driver

Notes
[1] The Solid FlowEngine and JDBC driver are supported for evaluation and development-only purposes.

Solid FlowEngine is not available on the AIX 5L 5.2 platform.

HTTP Servers
ATG supports the HTTP servers that are supported by IBM WebSphere Application Server 5.1. Please check wit HTTP servers.

Supported Environments when Running BEA Weblogic Application Server

When deploying the ATG platform and applications on the WebLogic Application Server, the following environments are supported:

BEA WebLogic Application Server
WebLogic Application Server 8.1 SP3

Production Environments
BEA WebLogic 8.1 SP3

Operating Systems

Java 2 Platform

Database Server

HPUX 11i [PA-RISC]

HP SDK 1.4.2.04 (32-bit only)

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

HPUX 11i v.2.0 [Itanium]*

*works with BEA WebLogic 8.1 SP2

HP SDK 1.4.2.04 (32-bit only)

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

Microsoft Windows 2000 [Intel] Server SP4

Microsoft Windows 2000 [Intel] Advanced Server SP4

Sun SDK 1.4.2_05 (32-bit only)

IBM DB2 Enterprise Server v.8.1 FixPack 5

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

Red Hat Enterprise Linux AS 3.0 update 1 [Intel]

Red Hat Enterprise Linux ES 3.0 update 1 [Intel]

Red Hat Enterprise Linux Server AS 2.1 [Intel]

Red Hat Enterprise Linux Server ES 2.1 [Intel]

Sun JDK 1.4.2_05

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

Sun Solaris 8 [Sparc]

Sun Solaris 9 [Sparc]

Sun SDK 1.4.2_05
(32-bit only)

IBM DB2 Universal Database v.8.1 FixPack 5

MS SQL Server 2000 SP3a

Oracle 9i (v. 9.2.0.4)

Oracle 8i Release 3 (v. 8.1.7)

Sybase ASE 12.5.2

Additional Environments for Evaluation and Development Only

The following environments are supported for Development only, in addition to all environments supported for Production.

Operating Systems

Java 2 Platforms

Database Server/JDBC Driver

All Supported production environments except HPUX 11i [Itanium] and HPUX 11i v. 2.0 [Itanium]

Solid FlowEngine 3.7 with

Solid JDBC 2.0 driver

Microsoft Windows XP

Sun JDK or BEA JRockit

Solid FlowEngine 3.7

with Solid JDBC 2.0 driver

Database Servers and JDBC Drivers

ATG supports the listed JDBC drivers in accordance with the application server vendor's supported configurations

Database Server	**JDBC Driver**
IBM DB2 Enterprise Server v.8.1 FixPack 5	DB2 UDB v.8.1 Type 2
MS SQL Server 2000 SP3a	i-Net OPTA XS v. 6.02 JDBC driver for MS SQL Server
Oracle 9i (v. 9.2.0.4)	Oracle 9i: Oracle OCI and Oracle Thin Drivers 9.2
Oracle 8i Release 3 (v. 8.1.7)	Oracle 8i: Oracle OCI and Oracle Thin Drivers 8.1.7
SOLID FlowEngine 3.7[1]	Solid 3.7: JDBC 2.0 Driver
Sybase ASE 12.5.2	Sybase jConnect 5.5

Notes: The Solid FlowEngine and JDBC driver are supported for evaluation and development-only purposes. S available on HP-UX 11i [Itanium] or HP-UX 11i v2.0 [PA-RISC].

HTTP Servers

ATG supports the HTTP servers that are supported by BEA WebLogic Application Server. Please check with BEo servers.

4.7 Next Steps

ATG Services provides a number of avenues for you to continue study on this topic including:

- **ATG Education** offers a number of instructor-led and onsite courses designed to meet the training needs of every member of your team. ATG offers education classes on the topics covered in this document, and can put together a comprehensive package of on-demand training to insure you get the highest quality results from your development team.
- **ATG Professional Services** offer a wide array of consulting capabilities for your project, including those covered in this document. Our consultants have unparalleled, real-life experience with ATG technology to implement the best practices covered here. You can reach your local professional services representative either through your Account Executive or by calling them directly:

VP:	Doug Gaffney	+312 604 5740
EMEA:	Andrew Hackett	+44 (0) 118 956 5040
East:	William Leong	+1 617 386 1157
Central:	Tom Gay	+1 312 604 5741
West:	Jon Feldman	+1 415 216 1731