ORACLE®

ATG WEB COMMERCE

Commerce Reference Store

Version 10.2.1

ASA Overview

Oracle ATG
One Main Street
Cambridge, MA 02142
USA

ASA Overview

Product version: 10.2.1
Release date: 01-17-14
Document identifier: ASAOverview1402110752

Copyright © 1997, 2014 Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Portions of this product may contain the following: EditLive Authoring Software Copyright © 2004 Ephox Corporation. All rights reserved. Some code licensed from RSA Security, Inc. Some portions licensed from IBM, which are available at http://oss.software.ibm.com/icu4j/. This product may include software developed by the Apache Software Foundation (http://www.apache.org/). Spell checking software from Wintertree Software Inc. The Sentry Spell Checker Engine © 2000 Wintertree Software Inc. This product also includes software developed by the following: Free Software Foundation, GNU Operating System, Incanto, JSON.org, JODA.org, The Dojo Foundation, Adobe Systems Incorporated, Eclipse Foundation and Singular Systems.

The software is based in part on the work of the Independent JPEG Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support: Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

The MIT License

Copyright (c) 2007 FlexLib Contributors. See: http://code.google.com/p/flexlib/wiki/ProjectContributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Table of Contents

1 Introduction

The Oracle ATG Web Commerce Assisted Selling Application (ASA) is an extension of Oracle ATG Web Commerce Reference Store which provides an interface to the existing Commerce Reference Store backend in the form of a new Oracle ATG Web Commerce server module. This module is a concrete example of how you can leverage Oracle ATG Web Commerce REST Web Services to make data available to a client application. It also provides an example of a client in the form of an iOS Universal Application that consumes data provided by the server module.

Assisted Selling is geared towards retailers who have low to medium traffic and the ability to interact with customers on an individual basis.

Benefits of Assisted Selling include:

- Customer insight for registered shoppers
- Insight into inventory availability by location
- Move POS onto the sales floor with engaged selling ability for store associate
- Enable associates with traditional shopper-facing commerce features
- Enable associates to access a subset of the traditional contact center (agent-facing) functionality
- Strengthen relationships with key clients

Assisted Selling provides a hybrid of agent/associate and customer functionality. This means that in-store employees can experience the some of the same product catalog, shopping cart, and other traditional customer-facing commerce functionality and also access enhanced agent-facing functionality to fully serve the needs of the customer.

From a developer standpoint, Assisted Selling provides an Objective-C REST client that encapsulates the server invocations made from the iOS client. This encapsulation hides many of apps inherent complexities from the implementer.

About this Document

This document discusses the features of Assisted Selling and the tools used to implement them. Each chapter builds on the information in previous chapters, so it is recommended that you read the chapters in order. The document includes the following chapters and appendix:

Defines the Assisted Selling product.

Version Compatibility (page 5)
Provides a links to update to date version information for Assisted Selling

Installing Assisted Selling (page 7)
Describes how to install Assisted Selling

High-Level Architecture (page 13)
Broadly discusses the architecture of Assisted Selling at a high level.

Low-Level Architecture (page 17)
Examines in more detail how different components are composed.

UI Features unique to Assisted Selling (page 25)
Examines the unique features in Assisted Selling that are different from Oracle ATG Web Commerce Service Center (CSC) and Commerce Reference Store iOS Universal App (CRS-IUA).

The Assisted Selling iOS Application (page 47)
Examines the Xcode project that make up and help organize Assisted Selling.

Assumptions

Since Assisted Selling is an extension of Commerce Reference Store, it is assumed that you are familiar with the Commerce Reference Store and Commerce Service Center products. For more information, see the *ATG Commerce Reference Store Overview*, the *ATG Commerce Reference Store Installation and Configuration Guide*, and the *ATG Commerce Service Center User Guide*. Similarly, while the section of this book called Endeca integration discusses CRS-IUA components of the ATG/Endeca integration, it is assumed that you are familiar with the Endeca product suite. See the *ATG Endeca Integration Guide* and other related Oracle Endeca installation documentation for more details. Since there are similarities and some shared code with Oracle ATG Web Commerce Reference Store iOS Universal Application (CRS-IUA) you should also review the *CRS-IUA Overview*.

Audience

This document was written for developers, mobile developers, and interested highly technical business control persons. These people include:

- **Web Designer/Page Developer.** Page developers create the content pages, integrating the scenario elements created by business users and the code elements created by programmers. Page developers are also responsible for overall Web site appearance.
- **Site Producer or Business Analyst.** Responsible for getting a site up and running, maintaining it day to day, troubleshooting issues, and making administrative changes not performed by an administrator or page developer.
- **Application Developer.** Programmers create the code elements that allow the system to provide dynamic, personalized site content. They also configure repositories and perform database administration duties.

2 What is Assisted Selling?

The Oracle ATG Web Commerce Assisted Selling Application (ASA) is an in-store reference application that merges the functionality of the Oracle Commerce Platform with the Oracle ATG Web Commerce Service Center server data on an iPad tablet device. This app provides:

- A server module (`Store.Mobile.DCS`) with a set of REST configurations to allow invocation from any JSON-consuming client.
- An Objective-C REST client for simplified invocation of server-side functionality by an iOS application.
- An iOS application providing Commerce Service Center functionality that consumes CRS JSON-formatted data produced by the `Store.Mobile.DCS-CSR` module.
- The ability to leverage the Commerce Service Center's out of the box web services.

Assisted Selling's server module is based on the existing Commerce Reference Store application's Nucleus components and configuration. If you are not already familiar with this application, please refer to:

- The *ATG Commerce Reference Store Overview*.
- The *CRS-IUA Overview* which documents the iOS version of CRS.
- The *CRS-M Overview*, which documents the mobile web version of CRS.

Commerce Reference Store iOS Universal Application (CRS-IUA) and Assisted Selling Compared

Commerce Reference Store iOS Universal Application (CRS-IUA) is a native iPhone and iPad application that interacts with the web application's backend to send and receive data. A Universal app runs on both the iPhone/iPod Touch and the iPad. From a developer's perspective, it is an iPhone and iPad app built as a single binary. Assisted Selling is designed to run only on the iPad. Assisted Selling differs in that it is intended to be used by an in-store employee who assists a shopper with his or her purchases.

ATG and Endeca Integration

As with CRS-IUA, a key feature of this product is the integration of Endeca features into Assisted Selling, specifically:

- Endeca Guided Search is used in both the desktop and iOS applications.
- Business users can use the Endeca Workbench Experience Manager to drive the page content management and layout of the search results and category pages.
- Business users can include additional Endeca storefront elements such as dimensions and breadcrumbs.
- CRS-M, CRS-IUA, and Assisted Selling demonstrate the usage of core Endeca cartridges.
- CRS-M, CRS-IUA, and Assisted Selling deliver multiple ATG-specific cartridges that can be leveraged as-is or extended by customers in their own implementations. These cartridges provide product-supported integration points between the ATG and Endeca platforms and help accelerate development at customer sites.

3 Version Compatibility

For information on supported operating systems, application server versions, JDK versions, and Endeca versions, see the Oracle ATG Web Commerce Supported Environments document in the My Oracle Support knowledge base (https://support.oracle.com/).

4 Installing Assisted Selling

Assisted Selling is dependent on Oracle ATG Web Commerce Reference Store and Oracle ATG Web Commerce Service Center.

The server module is at `CommerceReferenceStore/Store/Mobile/DCS-CSR`. The iOS source code is available in a separate distribution you can find on the Oracle Software Delivery Cloud at:

https://edelivery.oracle.com/.

The CIM Configuration (page 10) section refers to the existing ATG Installation and Configuration Guide to guide you through the setup and configuration of the server with which the iOS client interacts.

Licensing

Oracle ATG Web Commerce Assisted Selling Application is provided to you as an iOS project file that you may choose to use as a framework for creating your iOS application for your end users. If you choose to create an iOS application, then you must separately enter into a license and distribution agreement with Apple. You must complete and sign your finalized iOS application with your own iOS enterprise certificate. You bear all risks associated with the development of an iOS application.

Platform

The Assisted Selling client application is represented by several Xcode projects. These projects are built on iOS version 6.0.x and may be targeted for devices running iOS 6.0 or higher.

Configuration Options

There are several configuration options.

Two Commerce Reference Store sites

Commerce Reference Store provides three sites in its out-of-the-box configuration: ATG Mobile Home, and ATG Mobile Store, each with a different skin. Assisted Selling provides a site switcher for switching between ATG Store and ATG Home. These sites are also available in Spanish. The German version is not yet supported by Assisted Selling.

Internationalization

Assisted Selling currently supports Spanish and English, but the traditional Java resource bundle does not apply to iOS. See the *CRS-IUA Overview* for details on using `Localizable.strings` to externalize strings in your iOS code.

Translations for the promotional content items that appear on Assisted Selling's homepage are not located in resource bundles, but rather in the `Store.Storefront/data/catalog-i18n.xml` file, leveraging the same mechanism used in Commerce Reference Store. For more information, see the *Internationalization* chapter in the *ATG Commerce Reference Store Overview*.

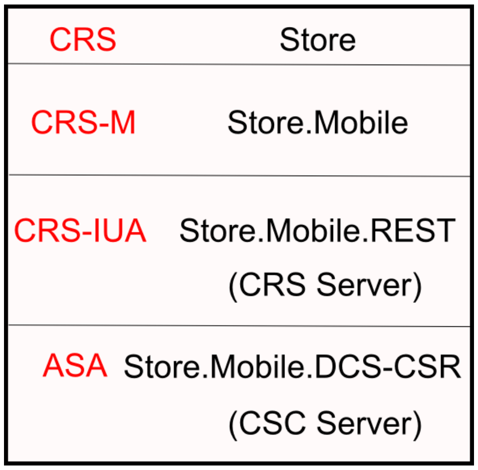
Note: Unlike CRS-M, Assisted Selling does not provide a language picker. In the associate settings popup, the associate is able to configure the catalog language, which changes the language of all the strings returned from the ATG server. Changing the language in the iOS Settings app only changes the client-side language.

Assisted Selling, CRS-IUA, CRS-M and Commerce Reference Store

Assisted Selling basically builds on top of what CRS-IUA and Commerce Service Center provides, which builds on top of what Commerce Reference Store provides. This means that Assisted Selling client relies in part on CRS-IUA and the entire server side structure is dependent on the Commerce Service Center server, including the multisite setup, form handlers, droplets and so on -- what we do in the server part of Assisted Selling is expose those things through REST.

Module Dependencies

The root Commerce Reference Store module is the `Store` module that you see under `<ATG10dir>/CommerceReferenceStore/Store` when you first install the application. This establishes a hierarchy of module dependency between the root module and the mobile and CRS-IUA/Assisted Selling modules, as shown in the following figure:



Learning about Dependencies

The dependencies that any given module has are specified in its `MANIFEST.MF` file. For example, in the manifest for `Store.Mobile.DCS-CSR` we see:

```
ATG-Product: The Mobile Commerce Store DCS-CSR module
Manifest-Version: 1.0
ATG-Config-Path: config/config.jar
ATG-Class-Path: lib/classes.jar
ATG-Required: Store.Mobile Store.DCS-CSR REST
```

This shows that `Store.Mobile.DCS-CSR` is dependent the module's root parent's `Mobile` module, and is also dependent on REST, meaning the `Store.Mobile` and the ATG REST module.

Prerequisites

To install and configure Assisted Selling, first download and install the following products:

- Endeca 3.1.2
- ATG Web Commerce 10.2
- ATG Web Commerce Search 10.2
- ATG Web Commerce Service Center 10.2
- ATG Web Commerce Reference Store10.2.1
- ATG Assisted Selling Reference App 10.2.1

See the *ATG Installation and Configuration Guide*, the *ATG Commerce Reference Store Installation and Configuration Guide*, and the *ATG Commerce Service Center Installation and Programming Guide* for details.

The following section discusses the options you need to select during setup and configuration after the base products are installed.

Setup and Configuration

Since Assisted Selling is dependent on Commerce Reference Store, the setup and configuration process for Assisted Selling is very similar to that for Commerce Reference Store. Defined here are only the installation and configuration steps that differ from the Commerce Reference Store steps that are detailed in the *ATG Commerce Reference Store Installation and Configuration Guide*. Follow all steps in that section for your chosen environment and add-on options. There are some additional required steps in the *Configuring ATG Products* section of the *ATG Commerce Reference Store Installation and Configuration Guide*, where CIM is used for setup. These steps are:

1. Choosing ATG Endeca Integration.

2. Choosing the Mobile Reference Store Add-On.

3. Configuring Mobile Reference Store Web Services.

In the sections that follow this installation, the steps required to build the Assisted Selling client application are provided.

CIM Configuration

CIM simplifies product configuration by providing scripts that configure Assisted Selling. The scripts allow you to identify the components used within your environment, as well as to add on additional applications. Using CIM ensures that all necessary steps are completed and are performed in the correct order.

CIM handles the following configuration steps:

- Creates data sources according to the database connection information you supplied, including those needed for applications you may add
- Creates database tables and imports initial data
- Creates and configures ATG application servers, including a dedicated indexing server, a lock manager and required loader servers
- Assembles your application EAR files for each ATG server, including modules for the Agent, Production and Data Warehouse load servers, as well as DCS-CSR, Fulfillment and UI modules
- Deploys EAR files to your application server and allows you to start up the agent-facing, customer-facing and load servers
- Allows you to add custom modules

Refer to the CIM script help and the *ATG Installation and Configuration Guide* for additional information on CIM.

To install Assisted Selling using CIM, do the following:

1. Install your application server.
2. Install your application files.
3. To start CIM, go to <ATG10dir>/home/bin and launch the CIM script:

```
./cim.sh | bat
```

1. Select the products you want to install.
2. Select the add-ons that you want to install.
3. Follow the CIM script according to the prompts. You can type H at any prompt for additional information.

Products and options to install

Select the following products:

- ATG REST
- ATG Site Administration
- ATG-Endeca Integration
- Oracle ATG Web Commerce Service Center
- Oracle ATG Web Commerce Reference Store

And then select the following ATG Commerce Reference Store AddOns and Mobile Reference store web services:

- Storefront Demo Application
- Mobile Reference Store
- REST Web Services for Native Applications

Obtaining the Xcode Source

The iOS source code is available in a separate distribution you can find on the Oracle Software Delivery Cloud site, edelivery.oracle.com.

iOS Client Setup

The preceding steps described how to configure the server side Store.Mobile.DCS-CSR module of Assisted Selling. It is important that Assisted Selling's iOS client is configured and built through Xcode so that the configuration matches the server side setup.

To get started, open the `Agent` project located in the directory where you unzipped the files, at:

```
<home>/MobileCommerce/Agent/Agent.xcodeproj
```

Configuration

To get Assisted Selling up and running in your environment, configure the following `Agent-Info.plist` entries:

- `ATG_REST_SERVER_HOST` is the hostname of the server running the Store.Mobile.DCS-CSR server module that you wish to connect to
- `ATG_REST_SERVER_PORT` is the port number of that server

Additionally, for a release build intended for production `ATG_USE_HTTPS` should be set to "YES" in order to enable SSL. This is strongly recommended and requires additional supporting configuration within your infrastructure to fully enable SSL.

Use this method of configuration when distributing an app, since the values of the `Agent-Info.plist` are used by default until they are manually overridden in the settings bundle.

Settings Bundle

The iOS client includes a settings bundle located `<home>/MobileCommerce/Agent/Agent/Settings.bundle`, where <home> is the directory where you extracted the distribution. This bundle manages server settings, such as ports, host, and Site ID, from the Settings app. This bundle is included in Assisted Selling for demonstration purposes. Once Assisted Selling is installed on the iOS device, touch Assisted Selling in Settings and configure the Host and Port server settings to point to your dedicated server.

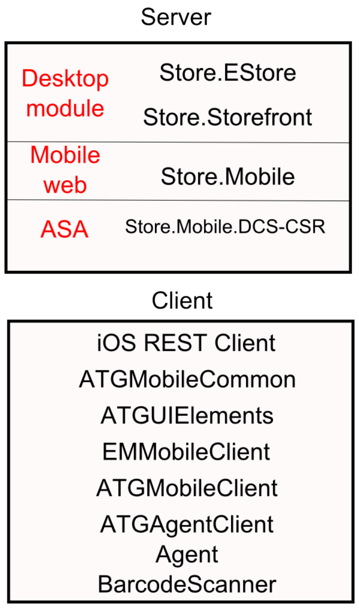
Use the Settings bundle for configuration when demonstrating the application since you can quickly change server settings without rebuilding the app and reinstalling. You may want to remove the settings bundle when releasing an app and use the values defined in `Agent-Info.plist`.

5 High-Level Architecture

At a high level, Assisted Selling consists of three parts that work together to provide remote communication between server and client:

1. A server module (`Store.Mobile.DCS`
2. An Objective-C REST client that communicates between web services and iOS clients
3. An iOS application that uses both of these components to demonstrate how commerce business logic can be implemented on a mobile client

The following figure illustrates the components within the server and client in Assisted Selling.



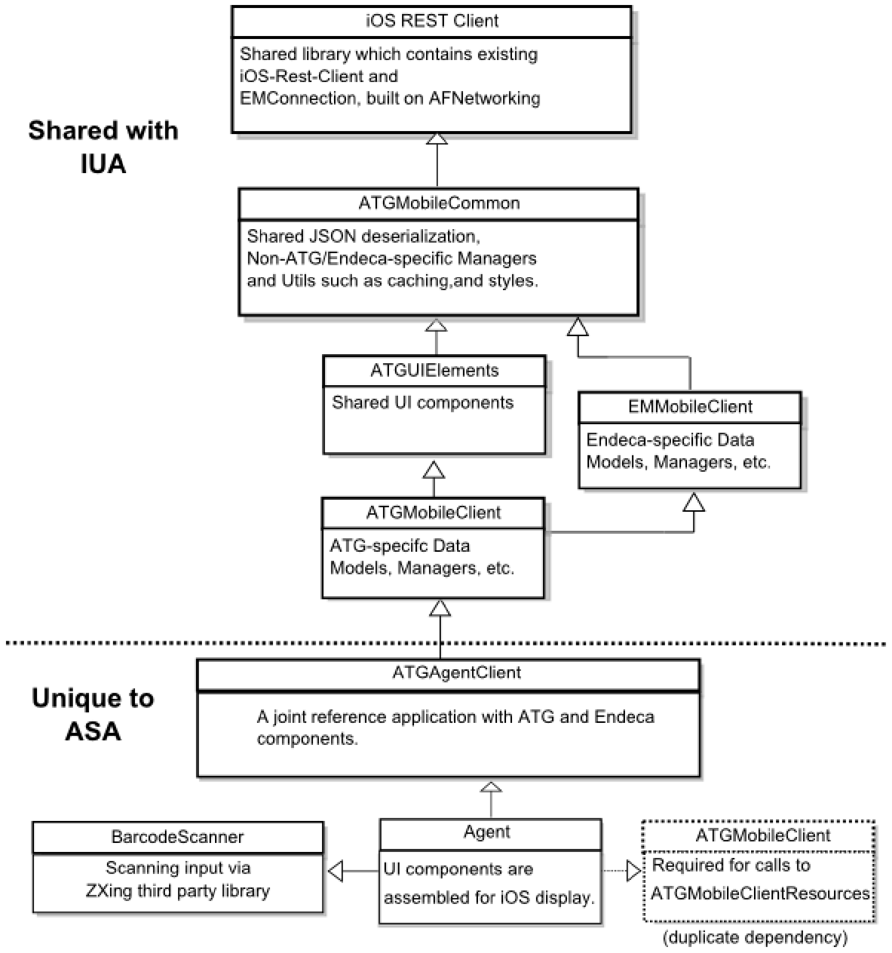
Server and client structure

The use cases and user experience (UX) design of Assisted Selling are largely the same as those for CRS-M and CRS-IUA. This allows the reuse of much of the backend configuration. For example, the same custom mobile form handlers are used by Assisted Selling to provide the same address and credit card functionality.

The architecture discussion in this document assumes familiarity with concepts related to Apple's iOS, including the Objective-C language.

File and Component Structure

The following is a high-level discussion of the structure of Assisted Selling, which is discussed in greater detail in the *Low-Level Architecture* (page 17) section. This high-level structure is shown in the following illustration:



Shared with IUA
iOS REST Client
Shared library which contains existing iOS-Rest-Client and EMConnection, built on AFNetworking
ATGMobileCommon
Shared JSON deserialization, Non-ATG/Endeca-specific Managers and Utils such as caching,and styles.
ATGUIElements
Shared UI components
EMMobileClient
Endeca-specific Data Models, Managers, etc.
ATGMobileClient
ATG-specifc Data Models, Managers, etc.
Unique to ASA
ATGAgentClient
A joint reference application with ATG and Endeca components.
BarcodeScanner
Scanning input via ZXing third party library
Agent
UI components are assembled for iOS display.
ATGMobileClient
Required for calls to ATGMobileClientResources
(duplicate dependency)

File and Component Structure

ATG/Endeca Network Connection

The ATG/Endeca Network Connection:

- Contains the low-level open source network layer.
- Contains ATG REST code and additional classes for connecting with an Endeca Assembler.
- Manages network connections, requests, and responses.
- Parses response objects into Apple `NSObjects` of type: `NSDictionary`, `NSArray`, `NSString`, `NSInteger`, etc.

Common Binding/Manager/Util

The Common Binding/Manager/Util:

- Contains interfaces for converting objects from `NSObjects` to typed objects.
- Contains base classes for managers which encapsulate non-(ATG/Endeca) specific concepts.
- Common Util layer for putting non-(ATG/Endeca) specific utilities, categories, caching, etc.
- Does not contain UI level code, shared or otherwise.
- Contains style manager, style classes, and style configuration.

ATGMobileClient

The `ATGMobileClient` contains:

- ATG-specific Managers, Utils, etc.
- ATG base model classes.
- Shared UI code (between CRS-IUA and Assisted Selling), and (ATG) generic classes.

ATGAgentClient

`ATGAgentClient` performs the same basic function as `ATGMobileClient`, except that it is tailored for the data returned from the Agent server, and configured to post to the Agent server instead of the Production server.

EMMobileClient

The `EMMobileClient` contains:

- Endeca- specific Managers, Utils, etc.
- Endeca base model classes.
- Very minimal UI code, and only well-designed Endeca generic classes.

UI/Style

The UI/Style contains:

- Well-designed generic UI classes.

Endeca Guided Search

The iOS search tab is driven by Endeca Guided Search.

Shared Code in Assisted Selling

The code shared between CRS-IUA and Assisted Selling is in the `ATGMobileClient` project, in `<home>\MobileCommerce\ATGMobileClient\ATGMobileClient\UI` folder. This folder contains both code and resources such as nibs and images. This code is accessible from the top-level projects `Agent` and `ATGMobileStore` using this import syntax:

```
#import <ATGMobileClient/MyClass.h>
```

Resources are built as part of a special target named `ATGMobileClientResources`. To access these resources from a top-level project, you must use this resource bundle instead of the main app bundle. To make this easier, we provide two methods that offer access to this resource bundle:

```
+ (NSBundle *) atgResourceBundle;
```

This is part of the `NSBundle` category `NSBundle+ATGAdditions`, and is part of `ATGMobileClient`. It returns the `ATGMobileClientResources` bundle, allowing for resources such as nibs to be loaded.

```
+ (UIImage *)locateImageNamed:(NSString *)pName;
```

This is part of the `UIImage+ATGAdditions` category in `ATGMobileCommon`. `locateImageNamed` looks for images in any resource bundle specified in the ATG_RESOURCE_BUNDLES property in the application `plist` file. Out of the box, `ATGMobileClientResources` is the only resource bundle specified in the application `plist` files, meaning that the behavior of `locateImageNamed` is as follows:

- Calls `[UIImage imageNamed]`, if the image is found, returns it.
- If no image is found, looks in `ATGMobileClientResources`. If found, the image is returned, otherwise nil is returned.

The styling framework uses `[UIImage locateImageNamed]` and any images specified in the `Theme.json` file is found automatically if contained inside `ATGMobileClientResources`.

Shared Code in Assisted Selling and CRS-IUA

There are two specific areas of shared code functionality:

- The Search and Browse code is shared between the two apps. There are some small differences in how these features are presented in the two apps, but the actual features are fully shared.
- The Product Details page is shared, but with some differences. The CRS-IUA code presents the ability to compare products, and to add to wish lists and gift lists, while the Assisted Selling version offers fulfillment options. The Product Details Page is made up of a number of classes, and the general approach is that base classes exist in `ATGMobileClient`, and the top-level projects contain subclasses where product-specific features can be added.

Other Code Sharing

Assisted Selling and CRS-IUA share code in other areas. The `ATG*Manager` classes used in CRS-IUA that are responsible for sending and receiving data to and from the server are re-used where possible. Assisted Selling has some unique Manager classes, but most of these are subclasses of Manager classes found in `ATGMobileClient` used in CRS-IUA. These subclasses simply modify the actors being used, and in some cases introduce different caching rules. The subclasses may also add extra functionality in some cases, such as overriding the path to a REST service.

6 Low-Level Architecture

This chapter explores the major components of Assisted Selling in more detail.

Store.Mobile.DCS-CSR Server Module

In Assisted Selling, the entire server side structure is dependent on the Commerce Service Center server, including the multisite setup, form handlers, droplets, etc. By design the server components of Assisted Selling are exposed through REST using the `Store.Mobile.DCS-CSR` server module.

REST Security and AgentLoggedInAccessController

All of our secure rest calls are secured using the access controller `AgentLoggedInAccessController`, which ensure that the user making the request is logged in and is an ‘Agent’ with the appropriate permissions.

```
/atg/rest/userprofiling/AgentLoggedInAccessController
```

Extending Commerce Service Center web services

Assisted Selling has been designed to extend many of the Commerce Service Center out-of-the-box web services. For more information on these services, see the REST section of the *ATG Web Services and Integration Framework Guide*.

Form handler extensions

These are the three form handlers that have been extended for Assisted Selling.

```
/B2CStore/Mobile/DCS-CSR/src/atg/projects/store/mobile/commerce/csr/order:
```

`MobileCSRCartModifierFormHandler.java`
`MobileCSRPaymentGroupFormHandler.java`
`MobileCSRShippingGroupFormHandler.java`

MobileCSRCartModifierFormHandler.java

`CSRCartModifierFormHandler` was extended to add a `shippingGroupNickname` property, so that when specified, it sets this form handler's shipping group to the associated value from the shipping group container map. This allows you to associate a `commerce item` with a shipping group when adding it to the order. For example, by setting `shippingGroupNickname="Home"`, the item is added to the "Home" shipping group contained in the SG map container.

Note that the SG map container must have been initialized with the `ShippingGroupDroplet`.

MobileCSRPaymentGroupFormHandler.java

`CSRPaymentGroupFormHandler` was extended to suppress the `ORDERAMOUNTREMAINING` relationship so that amounts less than the full invoice amount can be billed to any card, even the default card, without having the full unpaid amount be billed to the default card.

MobileCSRShippingGroupFormHandler.java

`CSRShippingGroupFormHandler.Mobile` was extended to allow you to apply shipping methods to by shipping group ID.

REST Actors in Assisted Selling and their Structure

Like CRS-IUA, Assisted Selling encapsulates the REST configuration and customizations required for communication between iOS applications and Oracle ATG Web Commerce server in JSON format. In Assisted Selling, these extensions or configurations would be in the `Store.Mobile.DSC-CSR` module which depends on `Store.Mobile.REST` module.

See the *CRS-IUA Overview* for more information on the implementation of REST.

Learning about Actor Chains in Assisted Selling

You can learn more about how actors are used in Assisted Selling by examining the registry which configuration and enables all of the actors used by the apps.

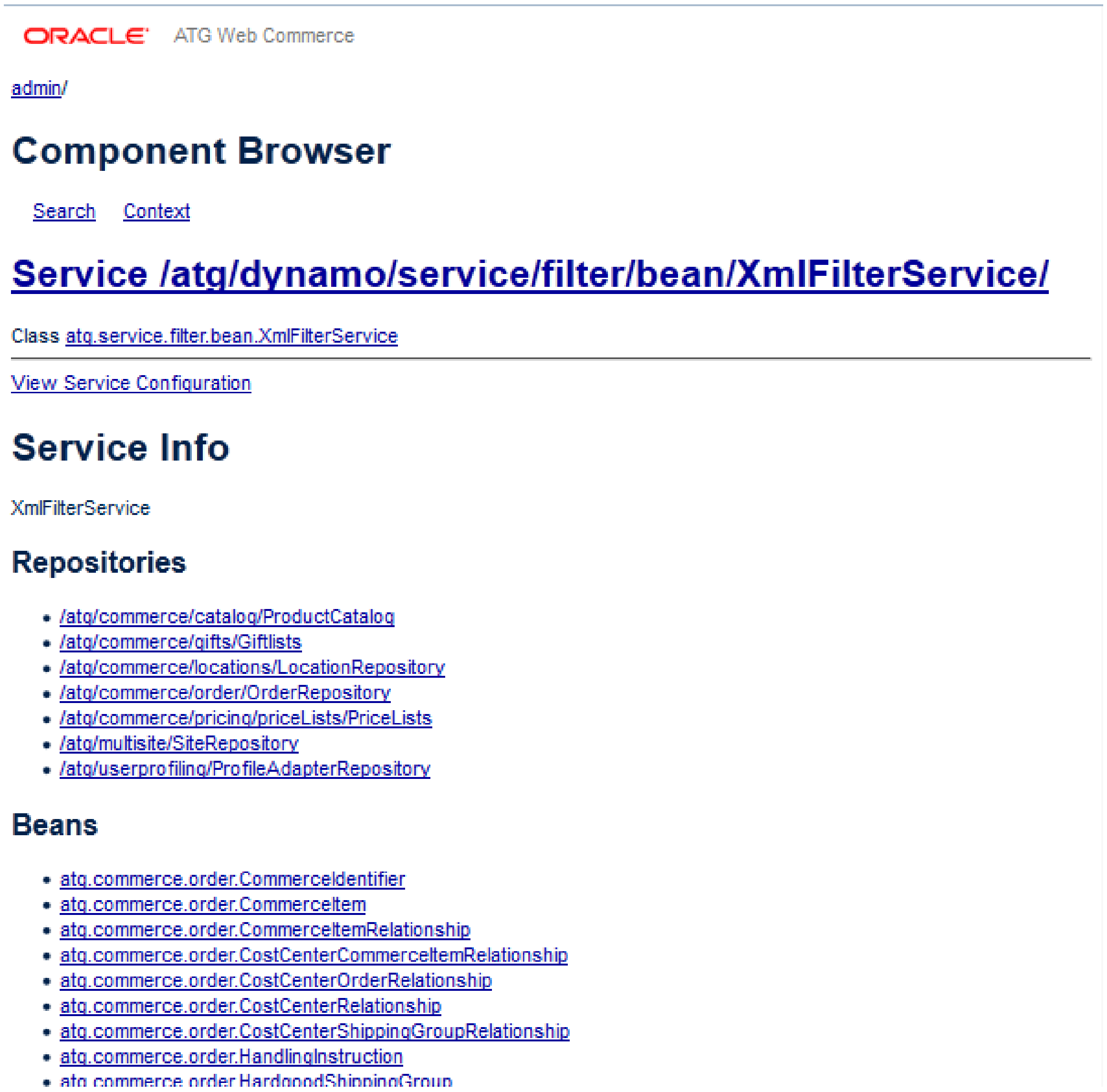
Assisted Selling actors are registered here:

```
/atg/rest/registry/ActorChainRestRegistry
```

There is a properties file for this in `Store.Mobile.DCS-CSR` for the Assisted Selling app and one in `Store.Mobile.REST` module for CRS-IUA.

A wealth of information about actors in your environment is available through the Dynamo Server Admin browser view. Start by viewing the filtering configuration, which determines which properties output for each component or repository item by using this URL, modified for your domain and configuration:

```
http://servername:serverport/dyn/admin/nucleus/atg/dynamo/service/filter/bean/
XmlFilterService/
```


ORACLE ATG Web Commerce

admin/

Component Browser

Search Context

Service /atg/dynamo/service/filter/bean/XmlFilterService/

Class atg.service.filter.bean.XmlFilterService

View Service Configuration

Service Info

XmlFilterService

Repositories

- /atg/commerce/catalog/ProductCatalog
- /atg/commerce/gifts/Giftlists
- /atg/commerce/locations/LocationRepository
- /atg/commerce/order/OrderRepository
- /atg/commerce/pricing/priceLists/PriceLists
- /atg/multisite/SiteRepository
- /atg/userprofiling/ProfileAdapterRepository

Beans

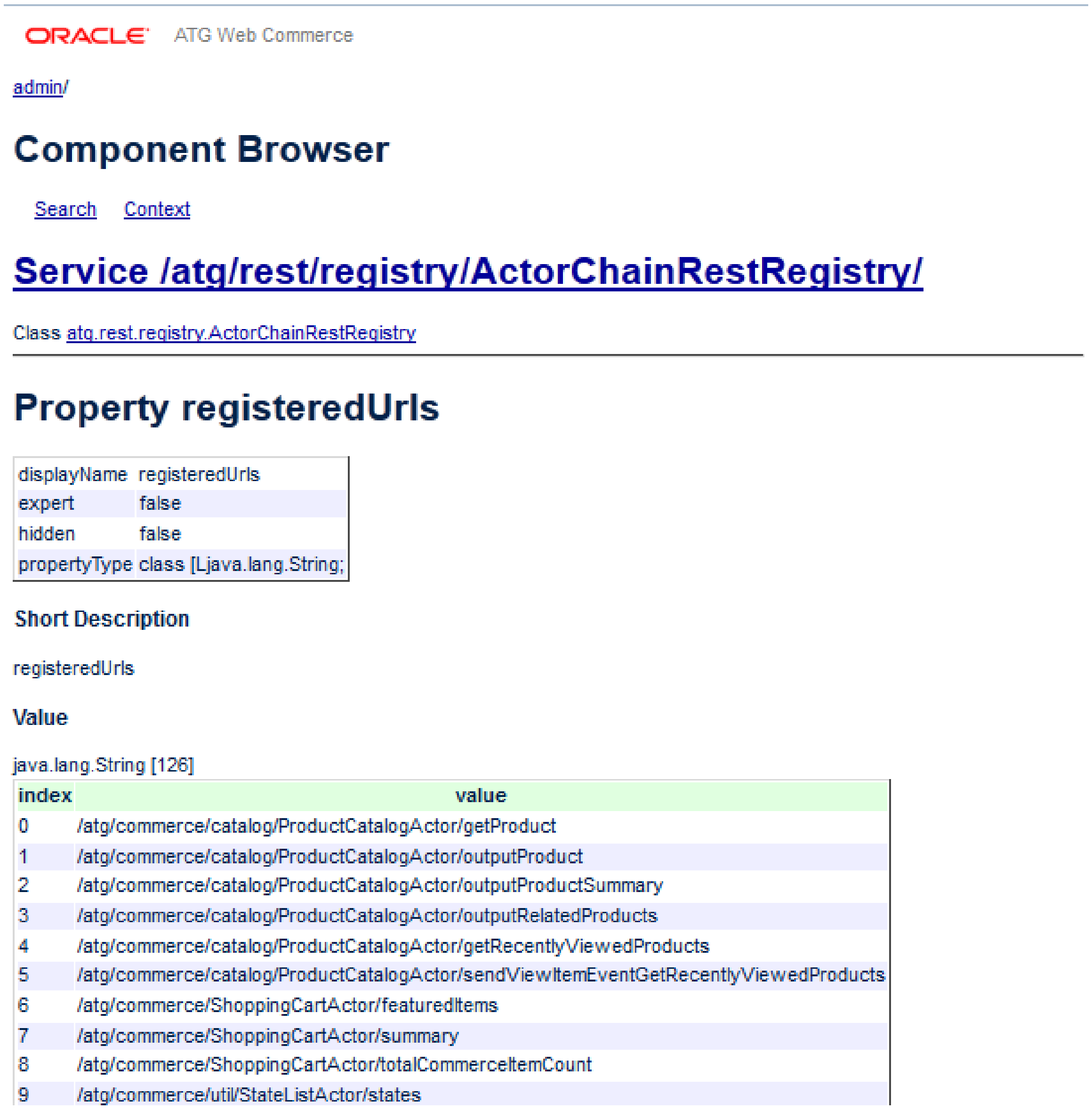
- atg.commerce.order.CommerceIdentifier
- atg.commerce.order.CommerceItem
- atg.commerce.order.CommerceItemRelationship
- atg.commerce.order.CostCenterCommerceItemRelationship
- atg.commerce.order.CostCenterOrderRelationship
- atg.commerce.order.CostCenterRelationship
- atg.commerce.order.CostCenterShippingGroupRelationship
- atg.commerce.order.HandlingInstruction
- atg.commerce.order.HardgoodShippingGroup

Dynamo Server Admin browser view

Use this URL to view a list of registered actors:

```
http://servername:serverport/dyn/admin/nucleus/atg/rest/registry/
ActorChainRestRegistry/?propertyName=registeredUrls
```

The list of registered actors is shown in the following illustration.



Component Browser

Service /atg/rest/registry/ActorChainRestRegistry/

Class atg.rest.registry.ActorChainRestRegistry

Property registeredUrls

displayName	registeredUrls
expert	false
hidden	false
propertyType	class [Ljava.lang.String;

Short Description

registeredUrls

Value

java.lang.String [126]

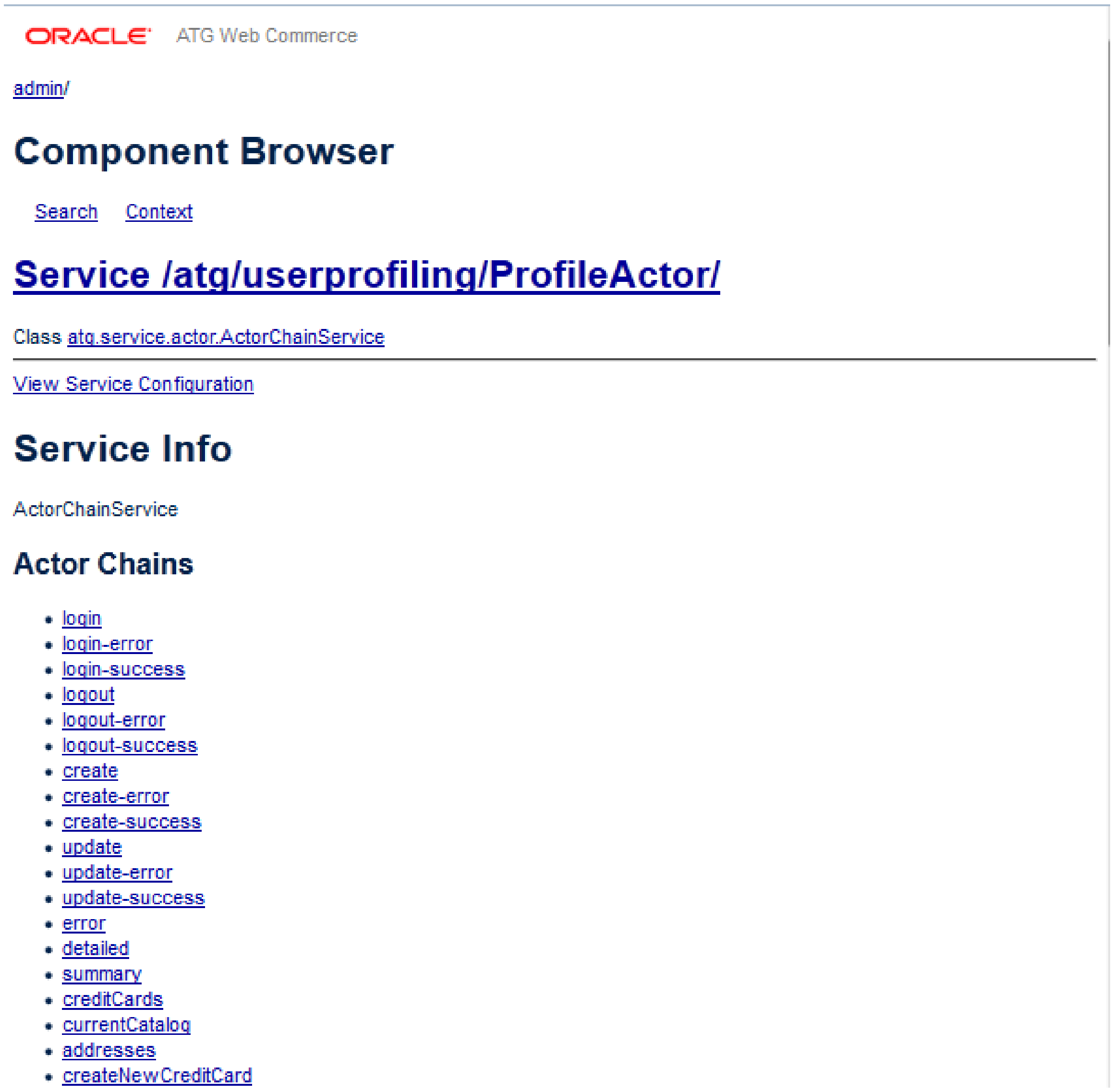
index	value
0	/atg/commerce/catalog/ProductCatalogActor/getProduct
1	/atg/commerce/catalog/ProductCatalogActor/outputProduct
2	/atg/commerce/catalog/ProductCatalogActor/outputProductSummary
3	/atg/commerce/catalog/ProductCatalogActor/outputRelatedProducts
4	/atg/commerce/catalog/ProductCatalogActor/getRecentlyViewedProducts
5	/atg/commerce/catalog/ProductCatalogActor/sendViewItemEventGetRecentlyViewedProducts
6	/atg/commerce/ShoppingCartActor/featuredItems
7	/atg/commerce/ShoppingCartActor/summary
8	/atg/commerce/ShoppingCartActor/totalCommerceItemCount
9	/atg/commerce/util/StateListActor/states

Dynamo Server Admin browser view – components

Then you can view a particular actor to see the available chains:

```
http://servername:serverport/dyn/admin/nucleus/atg/userprofiling/ProfileActor/
```

The browser view of actors and available views is shown in the following illustration.


ORACLE ATG Web Commerce

admin/

Component Browser

Search Context

Service /atg/userprofiling/ProfileActor/

Class atg.service.actor.ActorChainService

View Service Configuration

Service Info

ActorChainService

Actor Chains

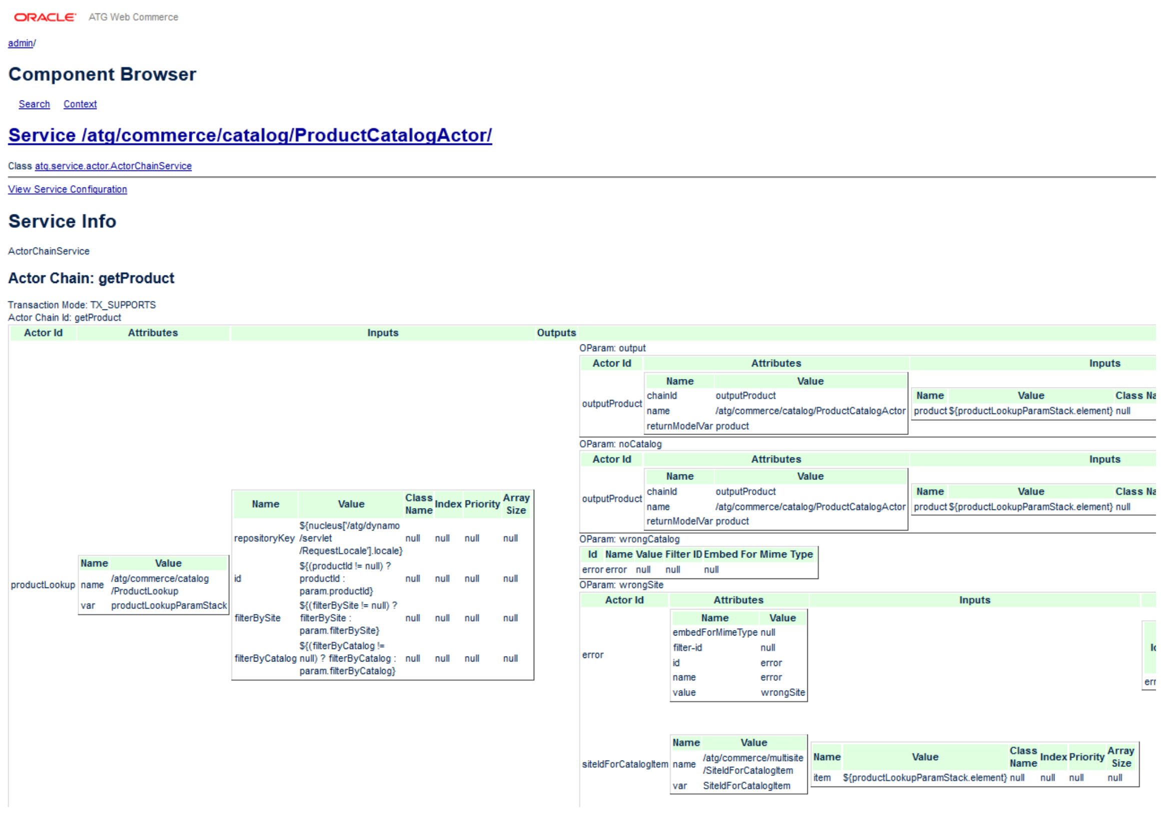
- login
- login-error
- login-success
- logout
- logout-error
- logout-success
- create
- create-error
- create-success
- update
- update-error
- update-success
- error
- detailed
- summary
- creditCards
- currentCatalog
- addresses
- createNewCreditCard

Dynamo Server Admin browser view – Actor Chains

For example, for products:

```
http://servername:serverport/dyn/admin/nucleus/atg/commerce/catalog/
ProductCatalogActor/?chainId=getProduct
```

The browser view of product actor catalog is shown in the following illustration.



ATG Web Commerce

admin/

Component Browser

Search Context

Service /atg/commerce/catalog/ProductCatalogActor/

Class atg.service.actor.ActorChainService

View Service Configuration

Service Info

ActorChainService

Actor Chain: getProduct

Transaction Mode: TX_SUPPORTS
Actor Chain Id: getProduct

Actor Id	Attributes	Inputs	Outputs
productLookup	(see below)	(see below)	(see below)

Attributes:

Name	Value
name	/atg/commerce/catalog /ProductLookup
var	productLookupParamStack

Inputs:

Name	Value	Class Name	Index	Priority	Array Size
repositoryKey	${nucleus['/atg/dynamo /servlet /RequestLocale'].locale}	null	null	null	null
id	${(productId != null) ? productId : param.productId}	null	null	null	null
filterBySite	${(filterBySite != null) ? filterBySite : param.filterBySite}	null	null	null	null
filterByCatalog	${(filterByCatalog != null) ? filterByCatalog : param.filterByCatalog}	null	null	null	null

Outputs:

OParam: output

Actor Id	Attributes	Inputs
outputProduct		

Name	Value
chainId	outputProduct
name	/atg/commerce/catalog/ProductCatalogActor
returnModelVar	product

Name	Value	Class Na
product	${productLookupParamStack.element}	null

OParam: noCatalog

Actor Id	Attributes	Inputs
outputProduct		

Name	Value
chainId	outputProduct
name	/atg/commerce/catalog/ProductCatalogActor
returnModelVar	product

Name	Value	Class Na
product	${productLookupParamStack.element}	null

OParam: wrongCatalog

Id	Name	Value	Filter ID	Embed For Mime Type
error	error	null	null	null

OParam: wrongSite

Actor Id	Attributes	Inputs
error		Id err
siteIdForCatalogItem		

error — Attributes:

Name	Value
embedForMimeType	null
filter-id	null
id	error
name	error
value	wrongSite

siteIdForCatalogItem — Attributes:

Name	Value
name	/atg/commerce/multisite /SiteIdForCatalogItem
var	SiteIdForCatalogItem

siteIdForCatalogItem — Inputs:

Name	Value	Class Name	Index	Priority	Array Size
item	${productLookupParamStack.element}	null	null	null	null

Dynamo Server Admin browser view – Actor Catalog

Configuration unique to Assisted Selling

There is an additional layer of configuration Assisted Selling which is captured in this file:

```
/atg/dynamo/service/filter/bean/beanFilteringConfiguration.xml
```

These configurations help return only the data that is relevant to the Assisted Selling app.

Filtering

You can use filter IDs with Assisted Selling actors to control how Assisted Selling data is returned. One filtering approach is to layer in properties on top of those defined in `Store.Mobile.REST` in `Store.Mobile.DCS-CRS`, using the same filter IDs, since properties added in `Store.Mobile.DCS-CSR` are only be added on the Agent server. Also, since the same model objects are used on the client, it is helpful to have the same base configuration.

Objective-C REST Client

The basic configuration of the REST client in Assisted Selling is the same as in CRS-IUA. See the *CRS-IUA Overview* for more information.

Objective-C JSONPath

The use of the Objective-C JSONPath in Assisted Selling is the same as in CRS-IUA, and is only used in places that are common between CRS-IUA and Assisted Selling. See the *CRS-IUA Overview* for more information.

JSON Parsing

The basic use of JSON parsing in Assisted Selling is the same as in CRS-IUA. See the *CRS-IUA Overview* for more information.

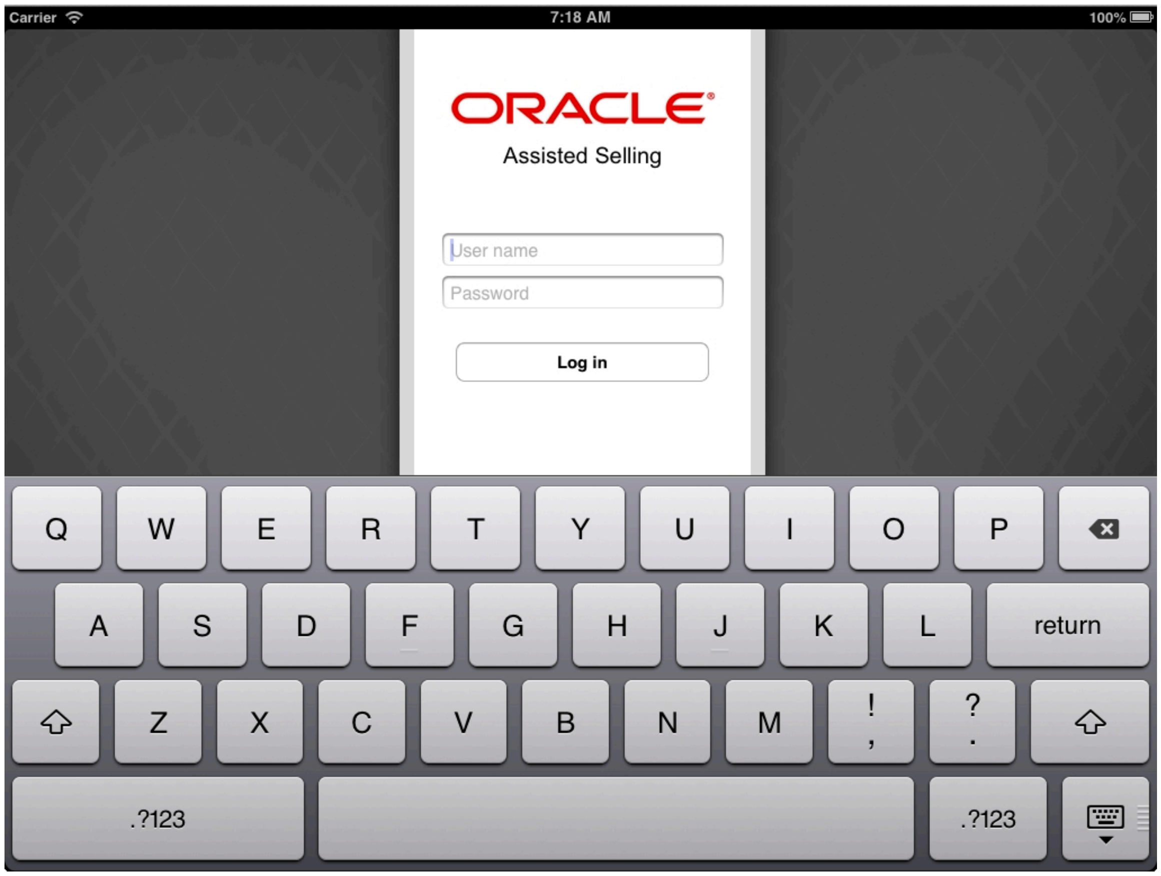
7 UI Features unique to Assisted Selling

While many features in Assisted Selling are shared with Commerce Service Center and CRS-IUA, there are several unique features.

Login

The sales associate login and password fields are limited to 40 characters, as in Commerce Service Center. If the session times out, the sales associate is prompted for a password and the application returns to the application.

Once logged in, the associate can identify shoppers or shop for a customer anonymously.



Assisted Selling login

Associate and Shopper Profiles

Associate profiles are created using the ATG Business Control Center. For more information see the *ATG Business Control Center User's Guide*. Price list and catalog is tied to profile. If the value is null, the configured default catalog is used instead.

Unlike CRS-IUA, Assisted Selling uses the internal profile handler for profile control.

Associate settings

The associate can chose a default physical store and default online site, and can select the preferred language for that site's catalog.

Associate Dashboard Page

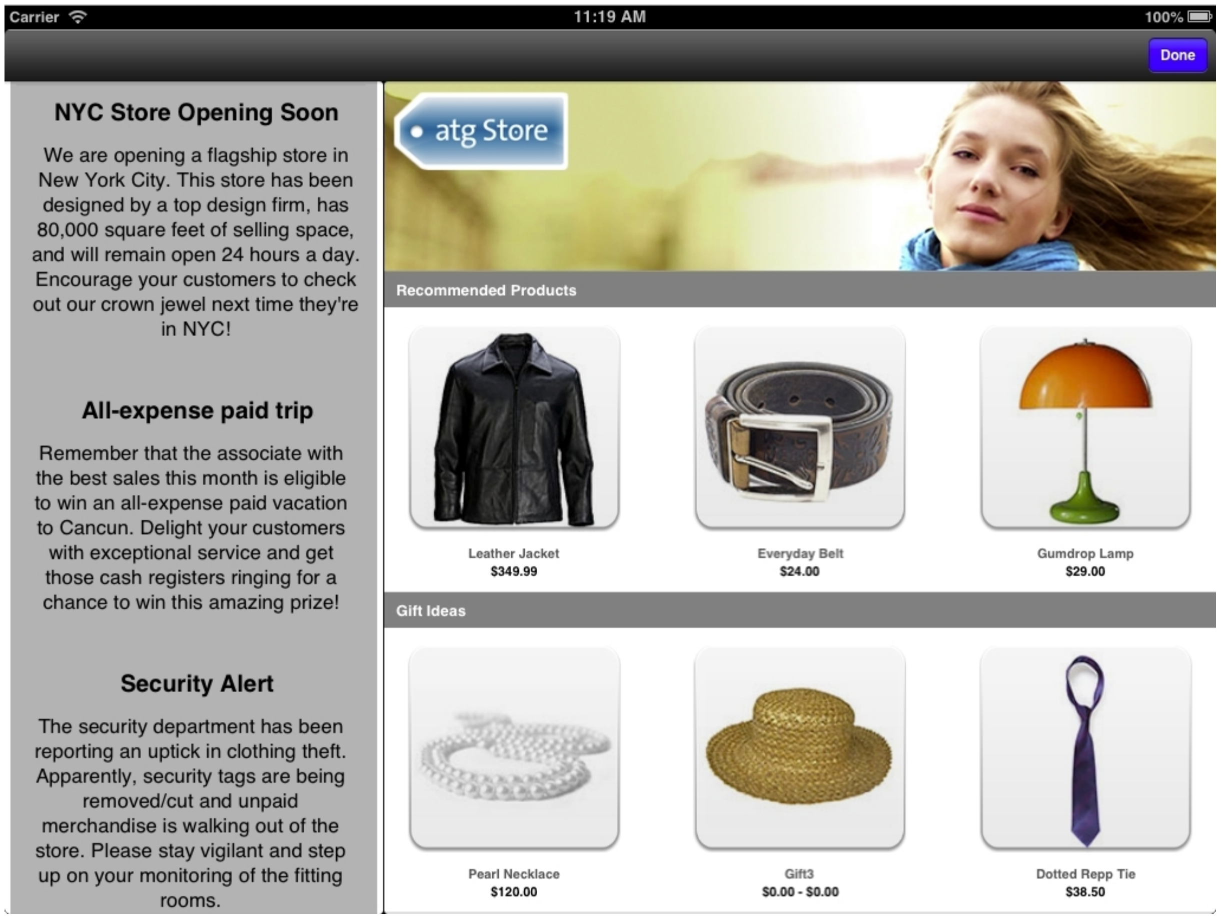
After login, information intended for the sales associate displays, such as:

- Specific products to promote

- Store or company text announcements
- Products on sale

This page is created using these design elements:

- A branded site-specific media banner
- A horizontal panel showcasing recommended products based on user segment
- A horizontal panel showcasing products from the Gift Shop category on the specified site or from any other category of the store merchandiser’s choice.
- A vertical list of store specific or company-wide announcements or news.



NYC Store Opening Soon

We are opening a flagship store in New York City. This store has been designed by a top design firm, has 80,000 square feet of selling space, and will remain open 24 hours a day. Encourage your customers to check out our crown jewel next time they're in NYC!

All-expense paid trip

Remember that the associate with the best sales this month is eligible to win an all-expense paid vacation to Cancun. Delight your customers with exceptional service and get those cash registers ringing for a chance to win this amazing prize!

Security Alert

The security department has been reporting an uptick in clothing theft. Apparently, security tags are being removed/cut and unpaid merchandise is walking out of the store. Please stay vigilant and step up on your monitoring of the fitting rooms.

Recommended Products

Leather Jacket $349.99
Everyday Belt $24.00
Gumdrop Lamp $29.00

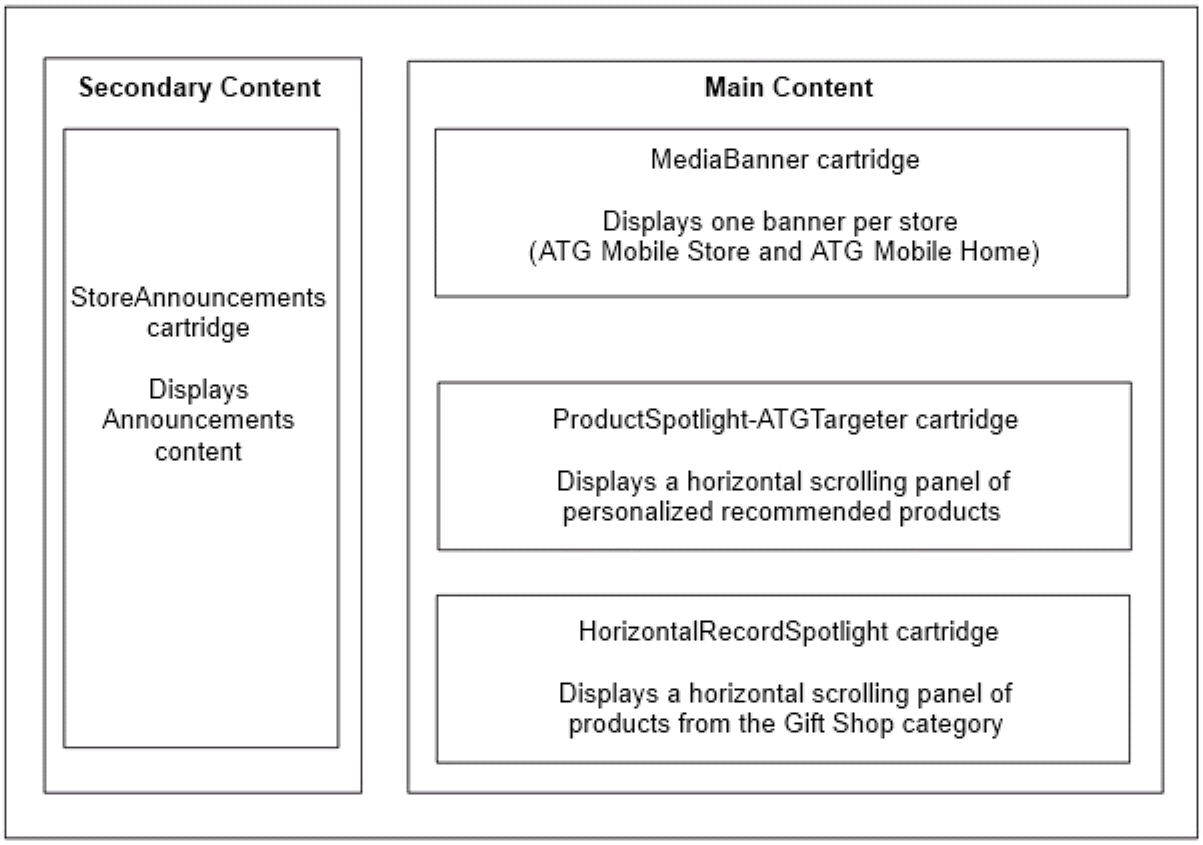
Gift Ideas

Pearl Necklace $120.00
Gift3 $0.00 - $0.00
Dotted Repp Tie $38.50

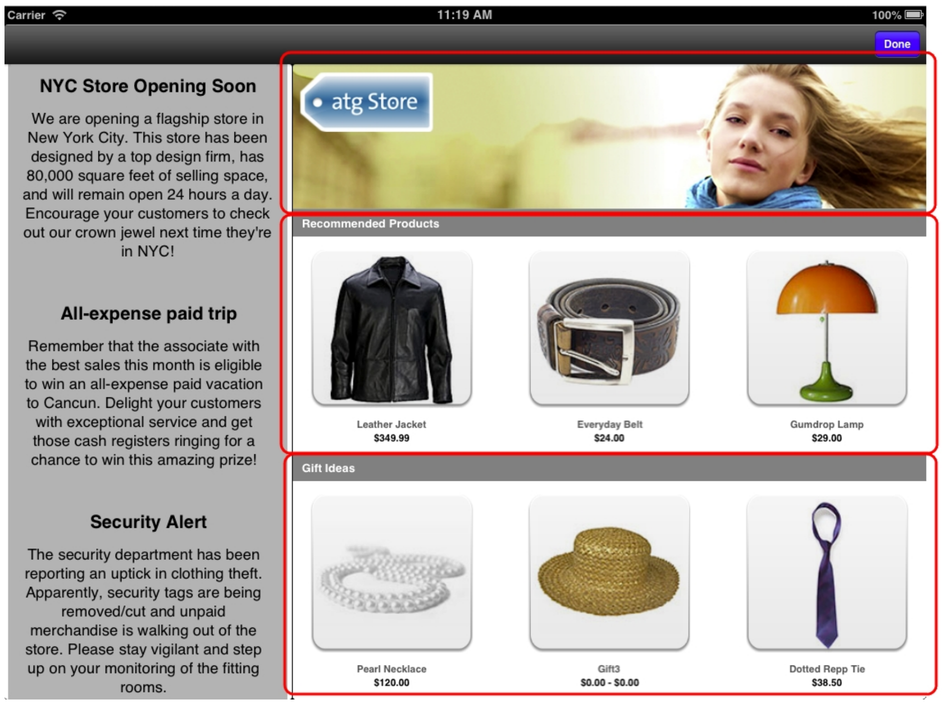
Associate Dashboard Page

Layout and cartridges used

The Assisted Selling Dashboard Page is configured in Experience Manager using the `MobilePage` template, using existing Experience Manager Cartridges and renderers, except for the “Announcements” panel. The `MobilePage` template layers two columns into the foreground and background to appear as a one column template:


Secondary Content
StoreAnnouncements cartridge
Displays Announcements content
Main Content
MediaBanner cartridge
Displays one banner per store (ATG Mobile Store and ATG Mobile Home)
ProductSpotlight-ATGTargeter cartridge
Displays a horizontal scrolling panel of personalized recommended products
HorizontalRecordSpotlight cartridge
Displays a horizontal scrolling panel of products from the Gift Shop category

Dashboard structure



NYC Store Opening Soon

We are opening a flagship store in New York City. This store has been designed by a top design firm, has 80,000 square feet of selling space, and will remain open 24 hours a day. Encourage your customers to check out our crown jewel next time they're in NYC!

All-expense paid trip

Remember that the associate with the best sales this month is eligible to win an all-expense paid vacation to Cancun. Delight your customers with exceptional service and get those cash registers ringing for a chance to win this amazing prize!

Security Alert

The security department has been reporting an uptick in clothing theft. Apparently, security tags are being removed/cut and unpaid merchandise is walking out of the store. Please stay vigilant and step up on your monitoring of the fitting rooms.

Leather Jacket $349.99	Everyday Belt $24.00	Gumdrop Lamp $29.00
Pearl Necklace $120.00	Gift3 $0.00 - $0.00	Dotted Repp Tie $38.50

MediaBanner Cartridge is used to display a site specific media banner with both a custom cartridge and renderer.

ProductSpotlight-ATGTargeter Cartridge is used to display targeter-driven products in the recommended products panel with content from the ATG repository using both a custom cartridge and renderer.

HorizontalRecordSpotlight Cartridge is configured to display a horizontal scrolling panel of products from the Gift Shop.

Dashboard page details

The StoreAnnouncements Cartridge

In Experience Manager select the `StoreAnnouncements` cartridge and renderer to configure the Announcements panel. This cartridge displays in Experience Manager only if you are running Commerce Service Center and the native mobile apps. A new item descriptor is created to hold store announcements. This only displays in the merchandising UI in the BCC for customers running Commerce Service Center and the native mobile apps.

The MediaBanner cartridge

This cartridge displays one banner per store (ATG Mobile Store and ATG Mobile Home).

ProductSpotlight-ATGTargeter cartridge

This cartridge displays a horizontal scrolling panel of personalized recommended products. The products displayed are based on the user segment of the customer that the associate is currently logged in as. This cartridge pulls content from the ATG repository using the ATG targeter `MobilePromotionChild1`.

HorizontalRecordSpotlight cartridge

This cartridge displays a horizontal scrolling panel of products from the Gift Shop category.

Modules for the Dashboard Feature

The Dashboard feature uses these modules:

Store.Mobile.DCS-CSR.Common

This module contains the definition of the `storeAnnouncement` repository item descriptor and runs on the Agent and Publishing servers.

Store.Mobile.DCS-CSR.Versioned

This module versions the `storeAnnouncement` repository item descriptor, is partially configured with BCC and runs on the Publishing server.

Store.Mobile.DCS-CSR.Common.International

This module modifies the `storeAnnouncement` repository item descriptor to allow for internationalization and runs on the Agent and Publishing servers.

Store.Mobile.DCS-CSR.Common.International.Versioned

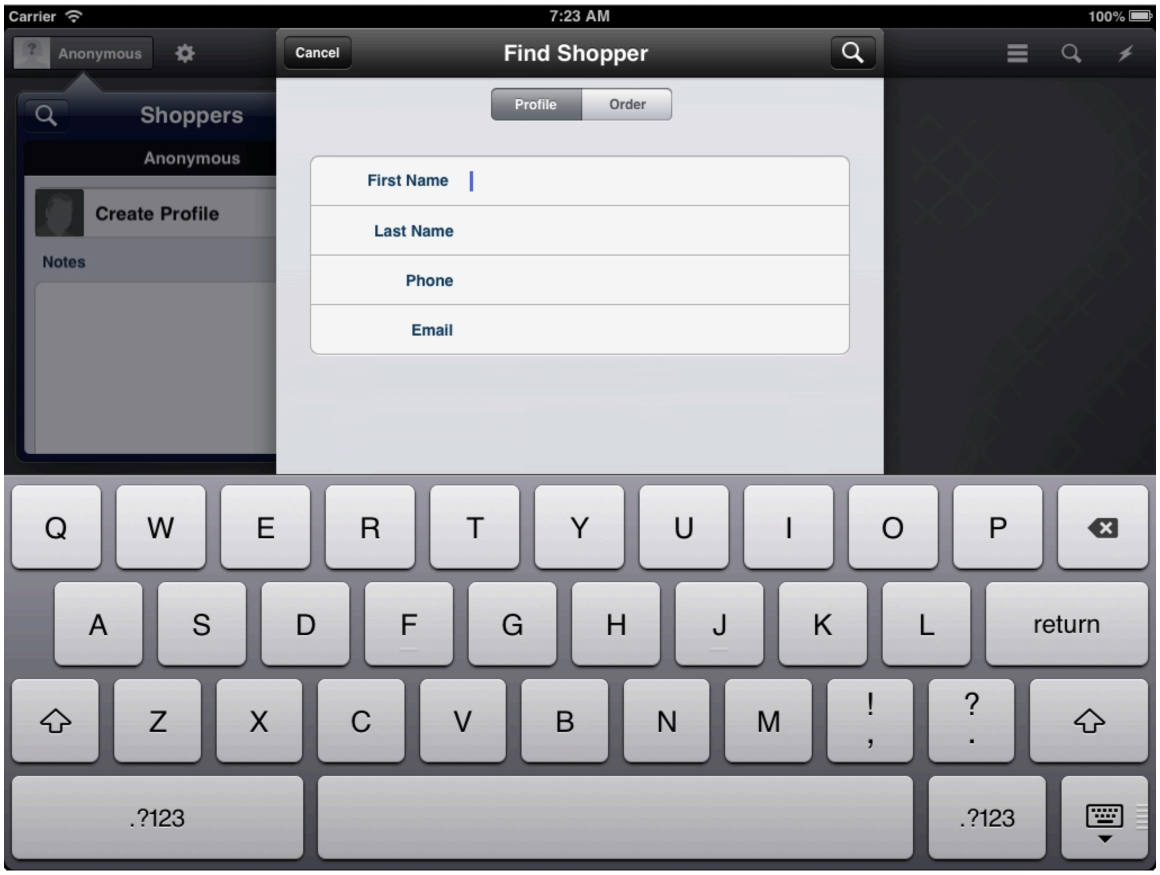
This module versions the modifications made in Store`.Mobile.DCS-CSR.Common.International` and runs on the Publishing server.

The associate can return to the Dashboard from the carousel view by touching the Dashboard icon.

Locating a shopper

The associate can look up the shopper's profile by:

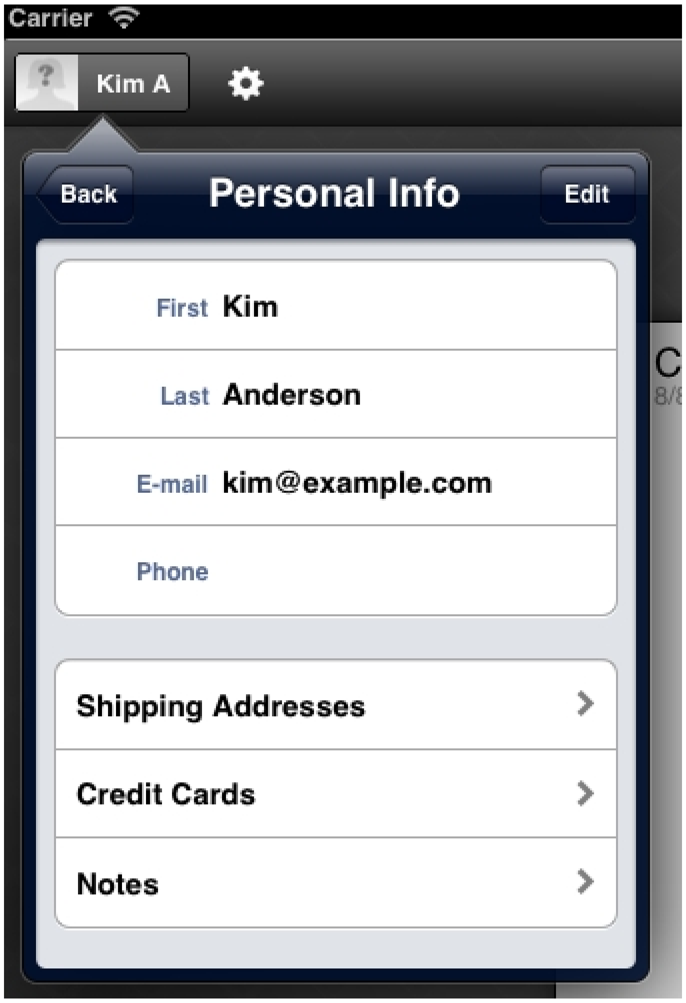
- First Name
- Last Name
- Phone Number
- Email
- Previous order



Finding a shopper

Once the shopper is found, Assisted Selling loads all of the applicable information associated with a registered shopper's profile:

- Profile information (name, email, birth date, profile ID, login, phone, password reset)
- Notes (sorted by date, newest to oldest)



First Kim
Last Anderson
E-mail kim@example.com
Phone

Profile edit

The associate can:

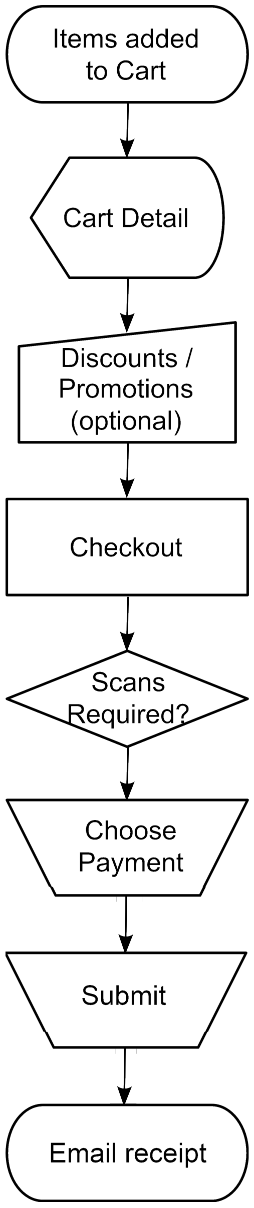
- View an existing profile
- Add a new profile
- View, modify ,delete, and add addresses and credit cards
- View order history
- View and modify an existing shopping carts

Note: Catalog and price list are applied at the time of the shopper selection.

Returns and exchanges are calculated and handled exactly as in Commerce Service Center.

High Level Cart flow

The next section describes the work flow from item selection to receipt of payment.



High Level Cart flow

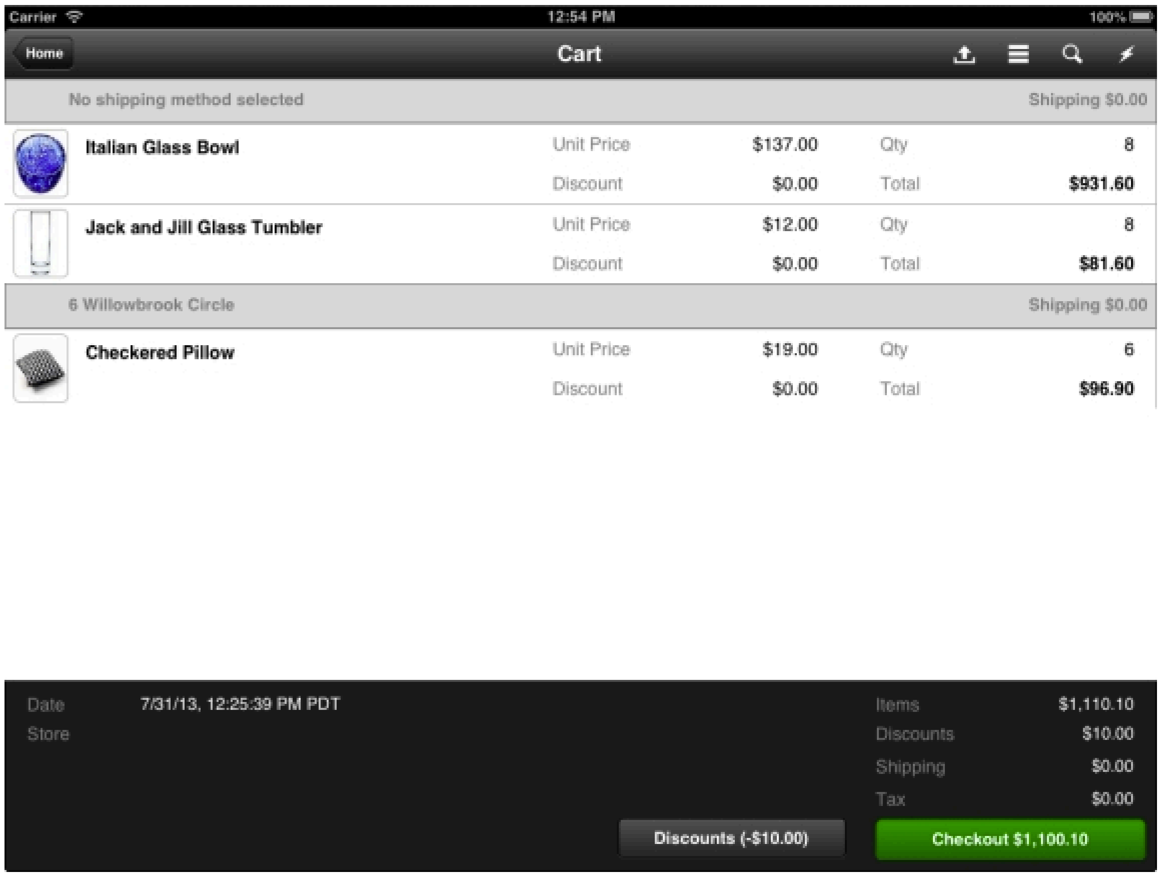
Shopping using an anonymous profile and cart

By default, all new shopper sessions start as an anonymous shopper profile.

- All of the items added while in anonymous mode can be saved to a newly created profile or existing shopper profile.
- The shopper may checkout anonymously as a guest shopper or the associate may identify and login a shopper at any point during the session, including checkout.
- Any carts that are created for an anonymous shopper are discarded at the end of a session if they are not saved by identifying and registering the shopper or by creating a new user account.

Shopping Cart, Bar Code Scan, and Search and Browse Inventory

In Assisted Selling the shopping cart is considered a type of list. The shopping cart is primarily used for items that the associate is assisting the shopper with, anticipating that the cart items will be purchased shortly using checkout.



No shipping method selected					Shipping $0.00
Italian Glass Bowl	Unit Price	$137.00	Qty	8	
	Discount	$0.00	Total	$931.60	
Jack and Jill Glass Tumbler	Unit Price	$12.00	Qty	8	
	Discount	$0.00	Total	$81.60	
6 Willowbrook Circle					Shipping $0.00
Checkered Pillow	Unit Price	$19.00	Qty	6	
	Discount	$0.00	Total	$96.90	

Date	7/31/13, 12:25:39 PM PDT	Items	$1,110.10
Store		Discounts	$10.00
		Shipping	$0.00
		Tax	$0.00

Cart

The associate is able to:

- Access the shopper’s online cart. The online cart includes items that the shopper may have added earlier from her desktop and mobile apps.
- Scan a product using the in-built iPad camera directly into the shopper’s online cart. Any item that is successfully scanned from the shop floor is flagged by the system as a store item. The scanner functionality uses the native iPad camera for scanning. Once the SKU is in focus, the item is added.
- Search for an item using the online catalog and add the item to shopper’s cart. Any item that is added from the online catalog is flagged by the system as an online item.
- Browse for an item using the online catalog and add that item to shopper’s cart.
- Add an item from the product detail page into the shopper’s cart.
- Adjust the order by changing item details such as order amount or apply a discount or credit to the order.
- Start the checkout process
- Manually refresh the cart contents using the pull-to-refresh gesture.

After the associate has reviewed the cart details with the shopper, the associate can continue to checkout from the cart page.

Note: The Endeca search and guided navigation is a direct reuse from CRS-IUA.

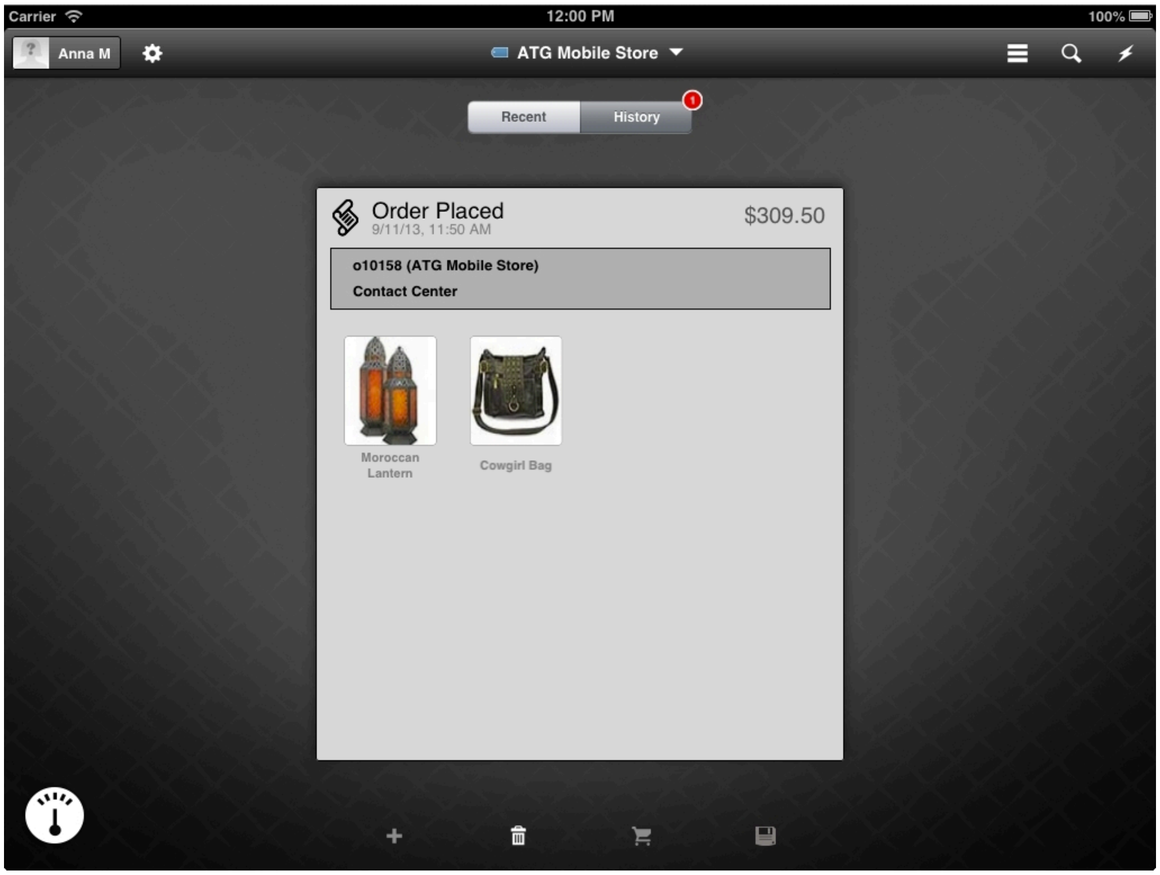
Multiple carts

While only one cart can be active at a time, the associate may have several carts in a carousel and move from cart to another with the swipe gesture. Only one anonymous cart can be in the carousel at any one time.

Order History

Commerce Service Center attempts to discover the price list and sale price list that was used to price the order. The order history for Assisted Selling is displayed in the carousel area.

Note: ATG Search is used when searching for a user by order on the Find Shopper dialog.



Order Placed

9/11/13, 11:50 AM

$309.50

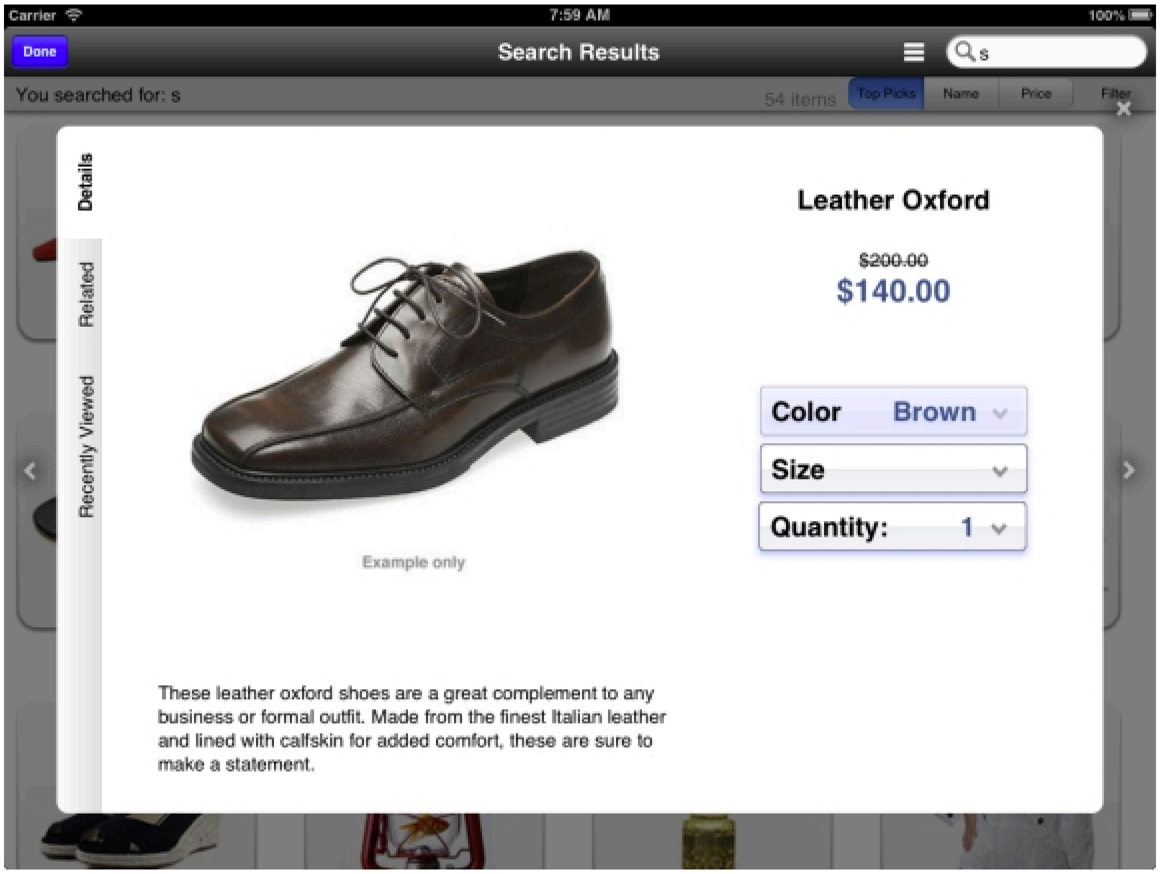
o10158 (ATG Mobile Store)

Contact Center

Order placed

Product Details

The Product Details functionality is inherited from CRS-IUA. Product detail is simply a view of the individual product that includes name, product image, description, price, color, size, etc. It also allows the associate to add the item to the cart, or compare items. The product detail page contains all the details available for the product. Unique to Assisted Selling is the ability to select a fulfillment and ship to method, and select different addresses for different items.

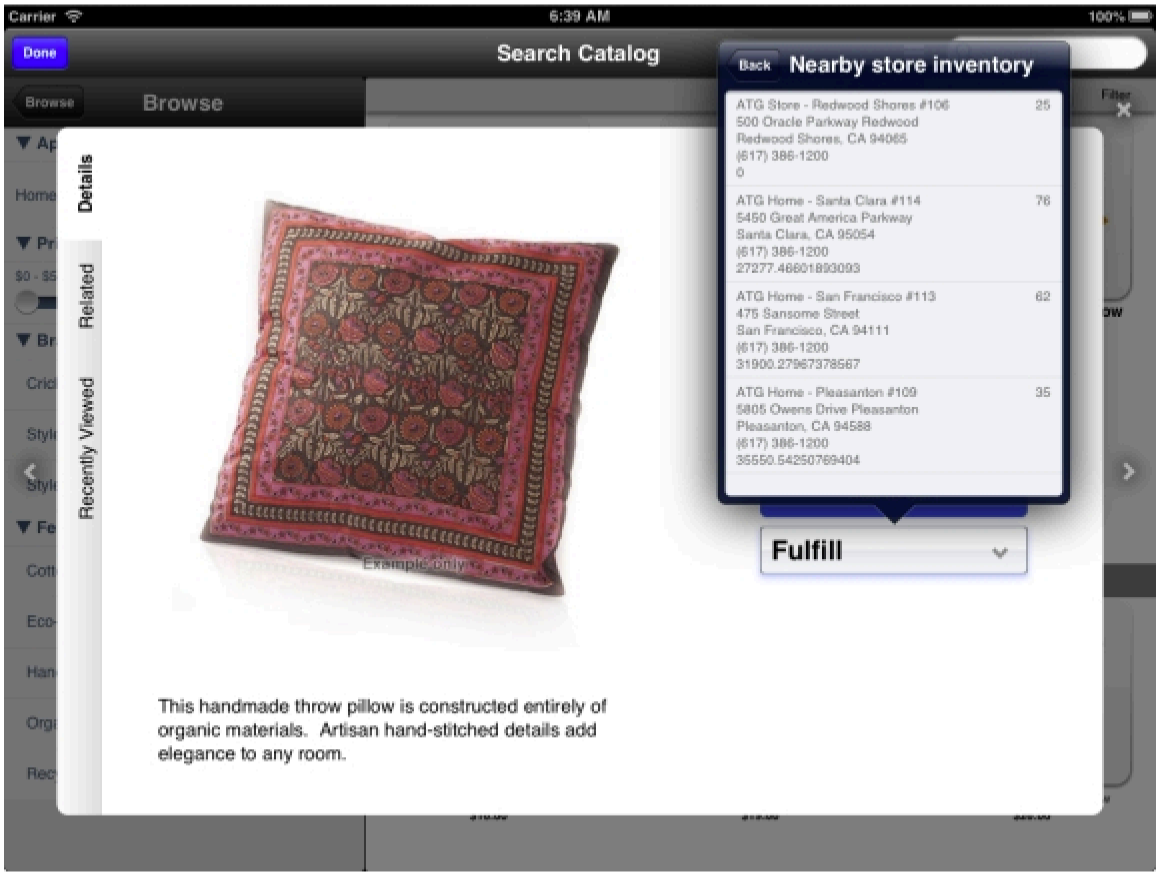


These leather oxford shoes are a great complement to any business or formal outfit. Made from the finest Italian leather and lined with calfskin for added comfort, these are sure to make a statement.

Product Detail page

Assisted Selling uses Web services from CRS and Commerce Service Center for the product display page. The functionality is primarily reused with a few exceptions. In Assisted Selling the associate can:

- View all product attributes (such as product image, title, price, description, SKU pickers) that are currently displayed on the consumer facing product detail page.
- Check whether the product is available from the online fulfillment location (typically warehouse).
- Check whether the product is available in the current store or a nearby store.



Nearby store inventory

ATG Store - Redwood Shores #106 25
500 Oracle Parkway Redwood
Redwood Shores, CA 94065
(617) 386-1200
0

ATG Home - Santa Clara #114 76
5450 Great America Parkway
Santa Clara, CA 95054
(617) 386-1200
27277.46601893093

ATG Home - San Francisco #113 62
475 Sansome Street
San Francisco, CA 94111
(617) 386-1200
31900.27967378567

ATG Home - Pleasanton #109 35
5805 Owens Drive Pleasanton
Pleasanton, CA 94588
(617) 386-1200
35550.54250769404

This handmade throw pillow is constructed entirely of organic materials. Artisan hand-stitched details add elegance to any room.

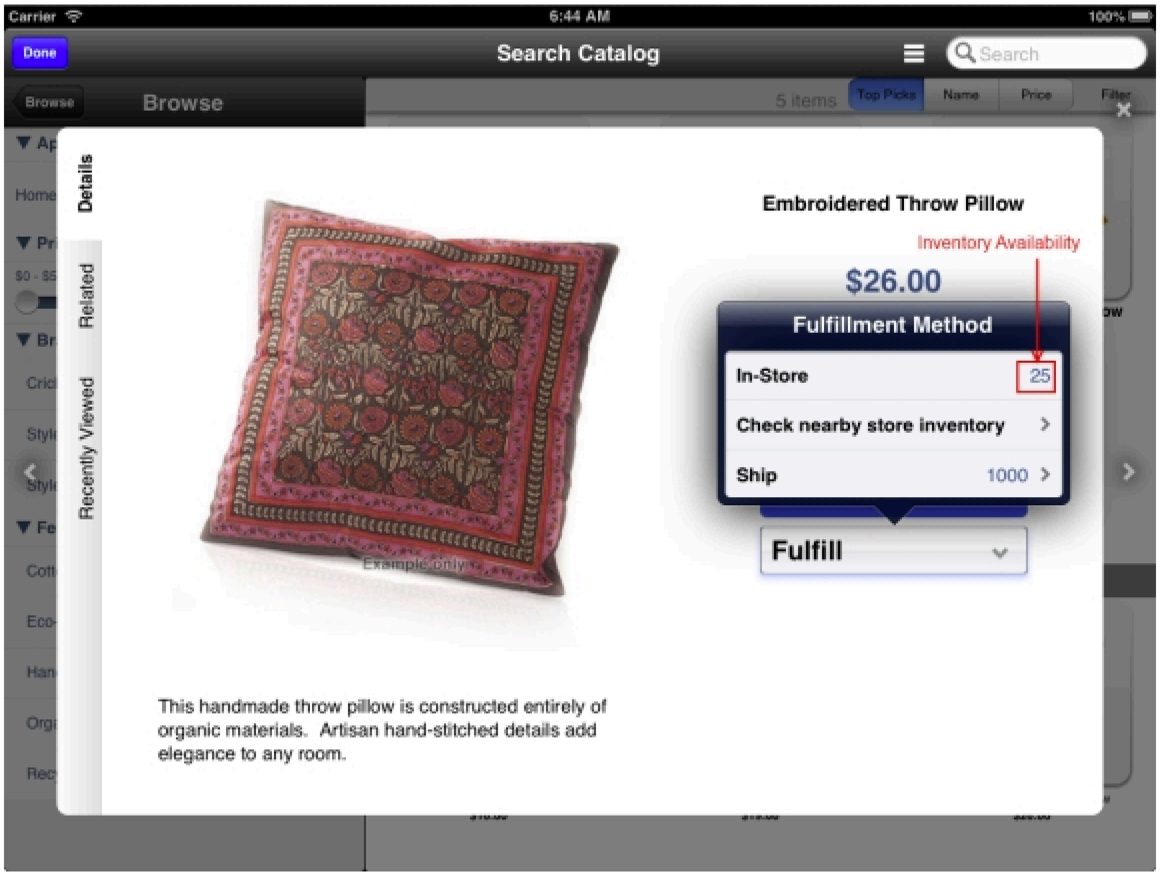
Store inventory

The associate can also:

- Identify that a product is part of the "online" portion of a shopper's or part of the "store" portion of a shopper's cart.
- View the Recently Browsed or Recently Viewed items on the product detail page, just as they are displayed on the consumer-facing iOS and mobile web apps.
- Search for other products from a product detail page.
- Add the product she is viewing to a comparison list from the product detail page.
- Select a fulfillment and ship to method.
- Select different ship to addresses for different items.

Unique to Assisted Selling

Unlike CRS-IUA, the product detail page in Assisted Selling includes inventory availability.



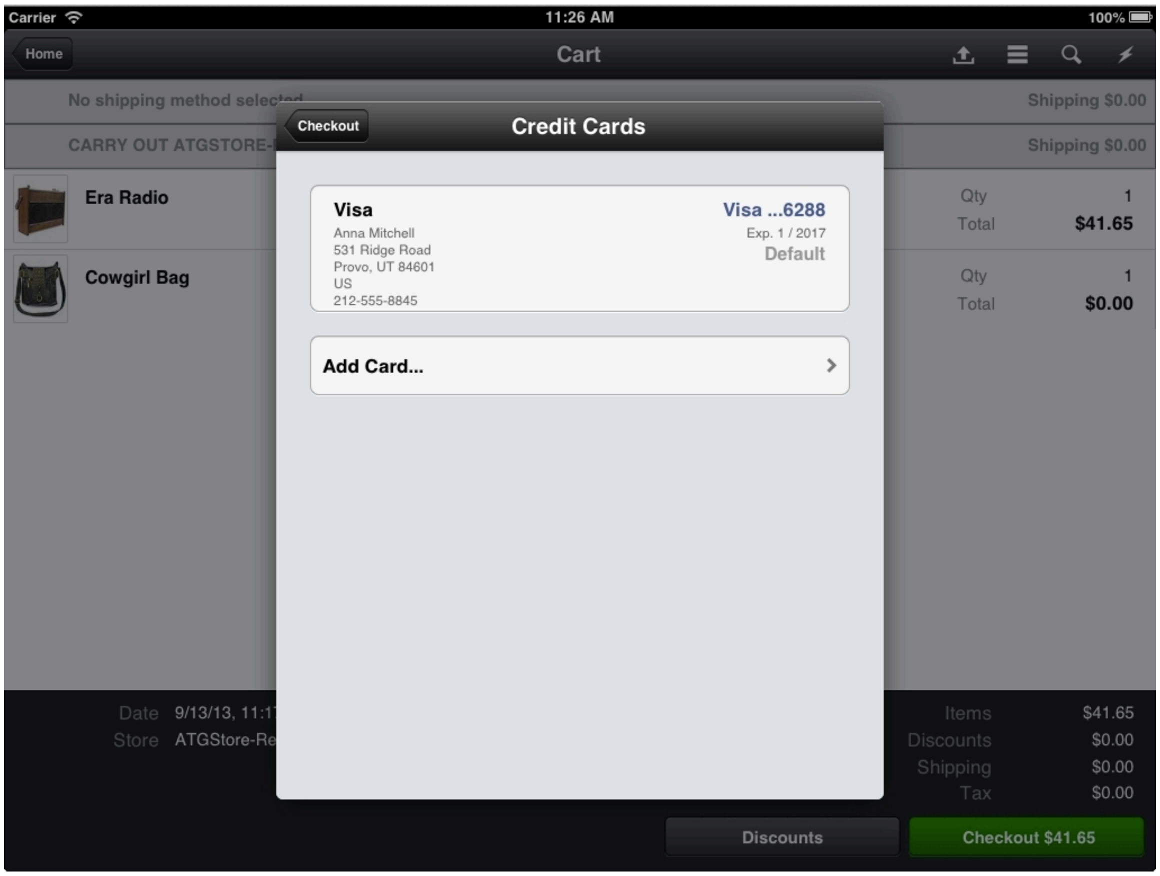
This handmade throw pillow is constructed entirely of organic materials. Artisan hand-stitched details add elegance to any room.

Inventory availability

Checkout

Once the shopper has completed shopping, the associate can begin the checkout process, either as an identified registered shopper or anonymously. The associate:

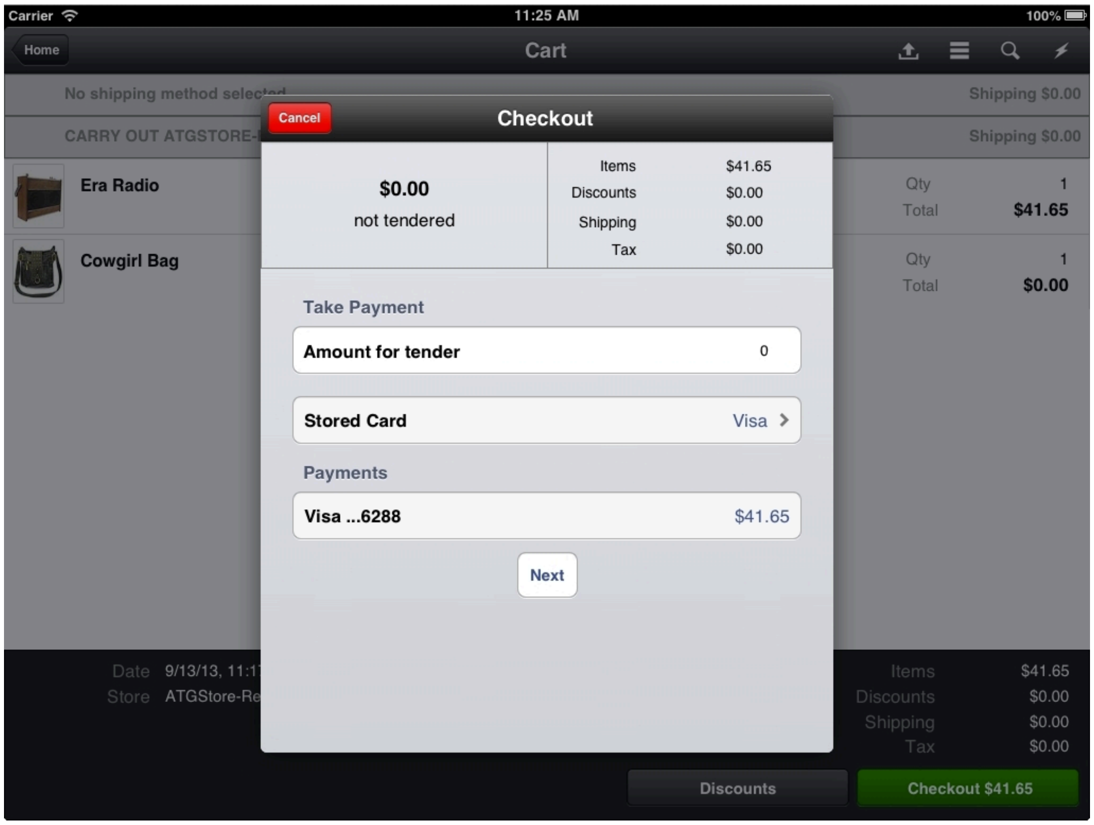
- Selects a shipping addresses, either using existing one or creating a new address. Optionally, the associate can choose multiple shipping addresses for different items.
- Determines payment type using an existing credit card or store credit, or creating new credit card entry.
- May split the tender amount across different credit cards.



Visa
Anna Mitchell
531 Ridge Road
Provo, UT 84601
US
212-555-8845

Visa ...6288
Exp. 1 / 2017
Default

Credit card



No shipping method selected — Shipping $0.00

CARRY OUT ATGSTORE- — Shipping $0.00

Era Radio	Qty 1	Total $41.65
Cowgirl Bag	Qty 1	Total $0.00

$0.00
not tendered

Items	$41.65
Discounts	$0.00
Shipping	$0.00
Tax	$0.00

Take Payment

Amount for tender	0
Stored Card	Visa

Payments

Visa ...6288	$41.65

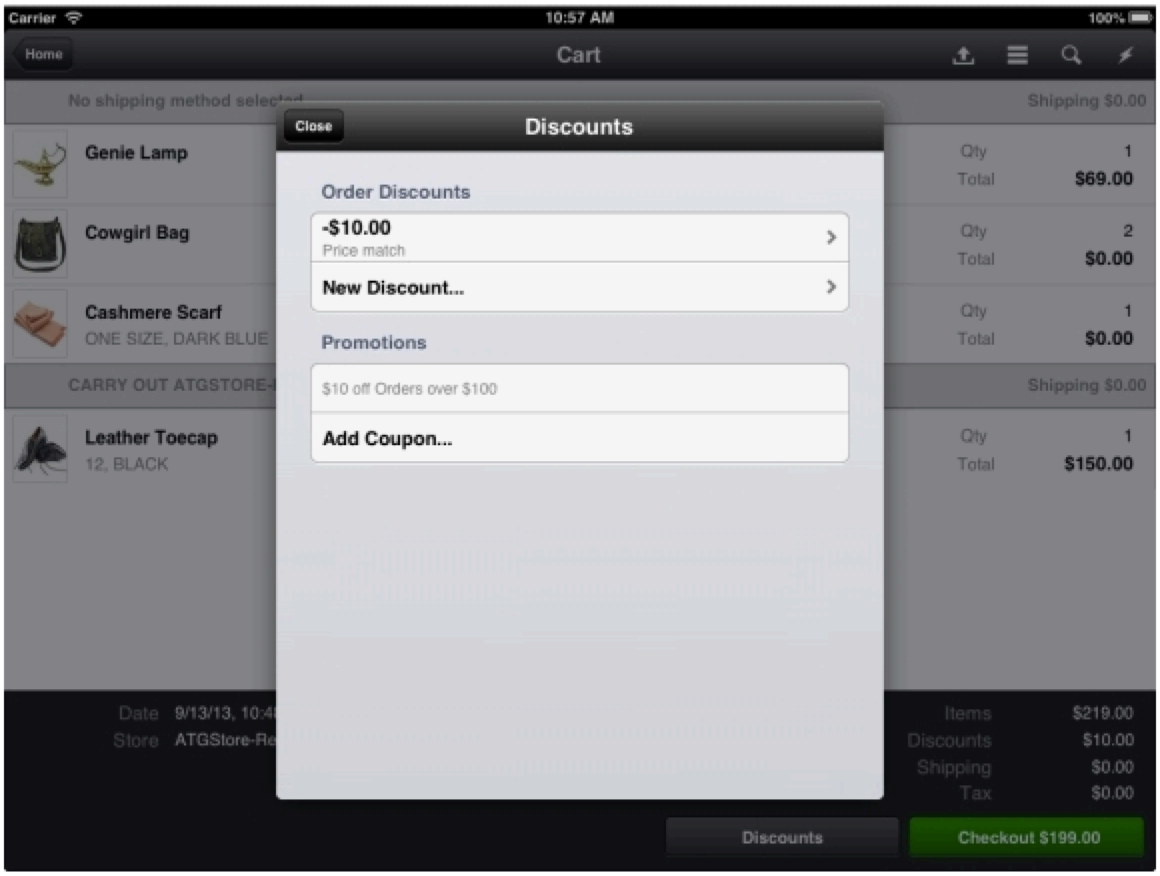
Date 9/13/13, 11:1
Store ATGStore-Re

Items	$41.65
Discounts	$0.00
Shipping	$0.00
Tax	$0.00

Checkout

Discounts and Promotions

During the checkout process, the associate can apply coupons and make currency based order discounts. The subtotal order amount is reduced by the currency discount amount. Note that like discounts, coupons may be automatically-applied.



Discounts

“In-Store” Items

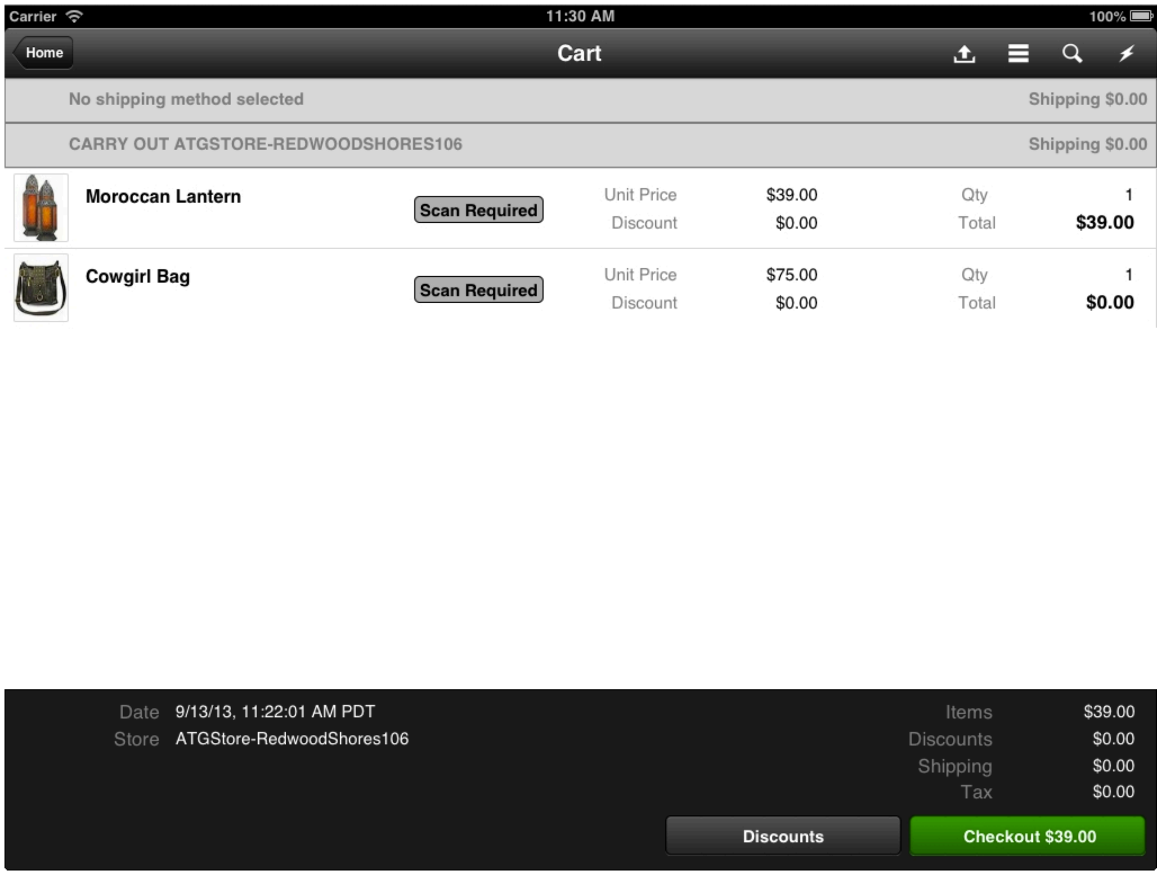
These are items from the shopper’s existing cart that they wish to take home from the store. These items are tagged as carry out items that need to be scanned are tracked through the unscanned Items view. The associate is prompted during the final steps of checkout to scan any unscanned items, but may optionally skip scanning.

Payments

Payments are made in the form of stored or new credit cards.

Scanning in store items

In Assisted Selling, the associate can scan a product using the in-built iPad camera directly into the shopper’s online cart. The scanner functionality uses the native iPad camera for scanning. Any item that is scanned from the shop floor is flagged by the system as a store item. An item marked as an in-store item is prompted for a scan during the checkout process, if it has not already been scanned.


Carrier 11:30 AM 100%

No shipping method selected — Shipping $0.00

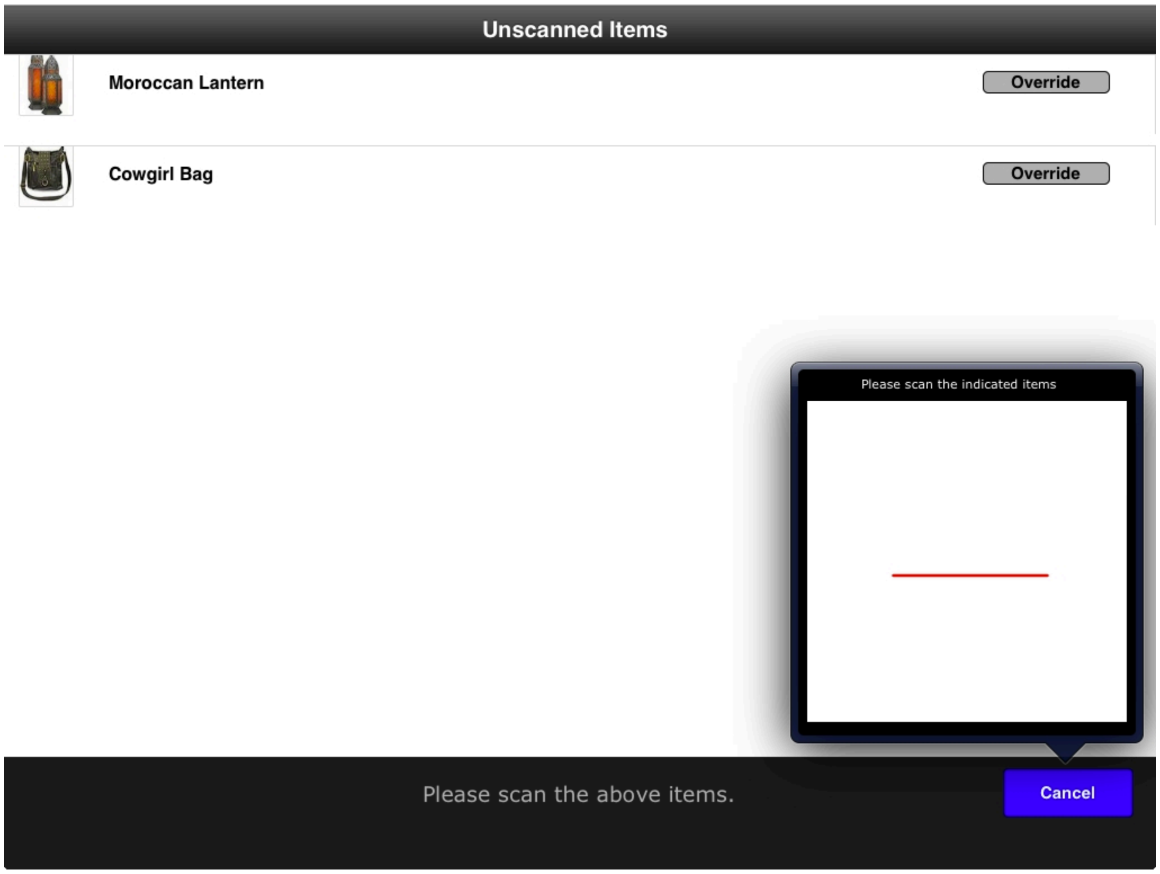
CARRY OUT ATGSTORE-REDWOODSHORES106 — Shipping $0.00

Moroccan Lantern	Scan Required	Unit Price	$39.00	Qty	1
		Discount	$0.00	Total	$39.00
Cowgirl Bag	Scan Required	Unit Price	$75.00	Qty	1
		Discount	$0.00	Total	$0.00

Date 9/13/13, 11:22:01 AM PDT
Store ATGStore-RedwoodShores106

Items	$39.00
Discounts	$0.00
Shipping	$0.00
Tax	$0.00

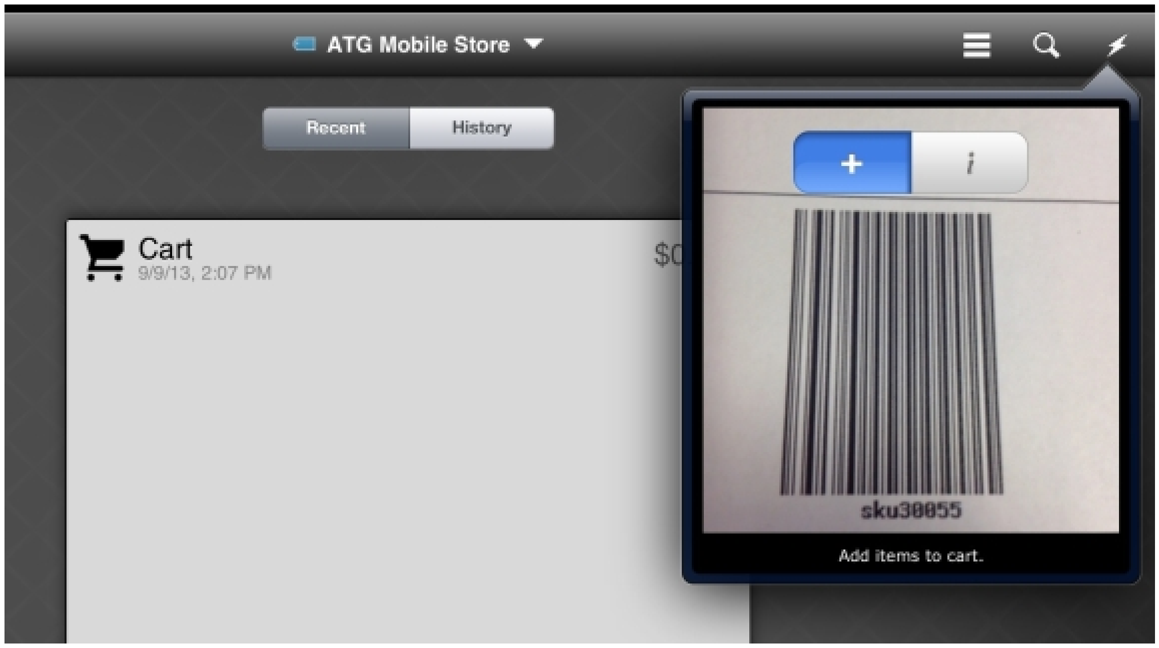
Scan required



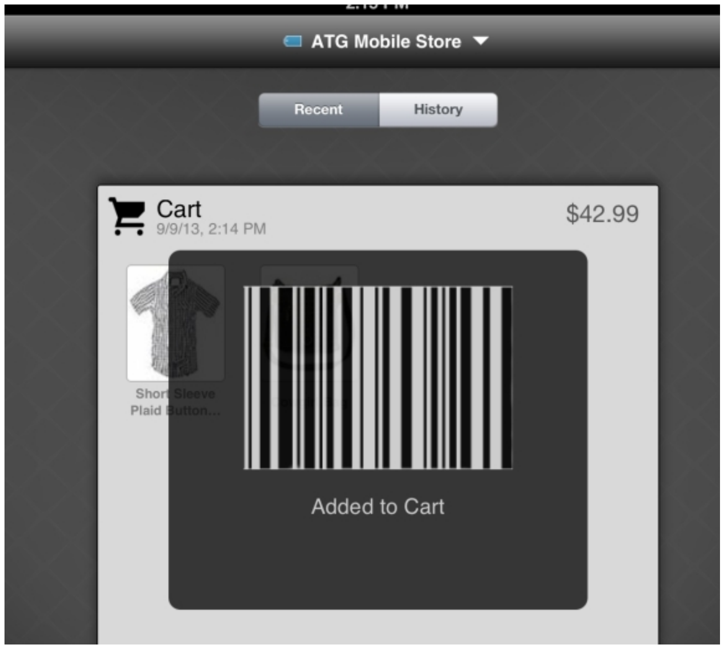
Scan at checkout

An item may be scanned automatically when the camera is able to focus on the SKU. Once the SKU is in focus, the item is added.

This functionality is enabled with a cross-platform third party library called ZXing.



Scanning an item



Scanned item added to cart

The ZXing library

ZXing is the cross-platform third party library used for barcode scanning. Assisted Selling uses an isolated library (`BarcodeScanner`) to wrap the ZXing project and presents the ZXing interfaces.

BarcodeScanner

`BarcodeScanner` is used as a library within a project, there are four classes available for use:

- `ATGOverlayView`

  This class wraps ZXing's `OverlayView` with custom drawing for the scanning view.

- `ATGBarcodeScannerViewController`

  This class wraps ZXing's `ZXingWidgetController` that manages the scanning view.

- `ATGBarcodeScannerHandler`

  This class is an interface used to obtain a single instance of the scanner.

- `ATGBarcodeScannerHandlerDelegate`

  This class is an interface for callback handling from the scanner/parser. The class that initializes the scanning process is responsible for ensuring that the delegate implements and register the instance using `ATGBarcideScannerHandler.`

Search by `sku.repositoryId` is supported, so the `BarcodeScanner` simply returns the scanned barcode and lets the main app handle the search/rendering of the parsed result.

ATGOverlayView Interface

```
// Values: OneD, TwoD, DataMatrix
- enum BarcodeScannerMode
// Parameter pModes is NOT mutually exclusive. That is, passed in values can be OR'd
 together.
- (id)initWithModes:(BarcodeScannerMode)pModes;
```

ATGBarcodeScannerViewController Interface

```
(id)initWithDelegate:(id<ZXingDelegate>)delegate modes:(BarcodeScannerMode)pModes;
```

ATGBarcodeScannerHandler Interface

```
// Handler is a singleton value. However, the delegate can be updated/changed by
 invoking this method with the desired callback listener.
+ (ATGBarcodeScannerHandler *)initWithDelegate:
(id<ATGBarcodeScannerHandlerDelegate>)pDelegate;
// Launches the scanner screen that handles the scanning/parsing of barcodes.
// Method invoker is responsible for presenting the returned view controller.
- (ATGBarcodeScannerViewController *)performBarcodeScan;
```

ATGBarcodeScannerHandlerDelegate Interface

The callback handler is responsible for dealing with the passed in view controller, including dismissing it.

```
- (void)didSuccessfullyParseScan:(ParsedResult *)pParsedResult withController:
(UIViewController *)pViewController;
```

```
- (void)didCancelScanRequest:(UIViewController *)scannerViewController;
```

8 The Assisted Selling iOS Application

The Assisted Selling iOS application is composed internally of several Xcode projects:

- `Agent.xcodeproj` (the main project)
- `ATGMobileClient.xcodeproj` (business logic)
- `EMMobileClient.xcodeproj` (UI layer)
- `ATGMobileCommon.xcodeproj`
- `iOS-rest-client.xcodeproj`
- `ATGAgentClient.xcodeproj`
- `BarcodeScanner.xcodeproj`
- `AgentClient.xcodeproj`
- `ATGGUIElements.xcodeproj`

Agent Project

The `Agent` project is a collection of managers, each having responsibility for a particular area of business logic.

Managers

The `iOSRestManager` has the important responsibility of establishing a connection to the server. Each of the other managers uses the `iOSRestManager` to make a request to the server via REST. The `iOSRestManager` also includes a number of helper methods for frequently performed tasks. For example, `checkForError` checks a REST response for errors.

Managers encapsulate request creation and response management logic that `ViewControllers` can use to handle the creation of requests for data, and the receipt of their responses. `ATGManagerRequest` is a wrapper around `ATGRestOperation` and is the base class for all other requests in Assisted Selling. `ATGManagerRequest` gets a reference to the result of the call. For example, the `ATGProfileManager.getProfile` method returns a reference to an `ATGProfileRequest`, which contains the actual profile representation in its `requestResults` property.

Each manager uses a corresponding subclass of `ATGManagerRequest` for its server communication. For example, `ATGProfileManager` uses `ATGProfileRequest`. The definitions of the managers and their requests are located in the `Managers` directory of the `ATGMobileClient` project.

The following table shows the managers that are provided by Assisted Selling and a brief summary of their responsibilities.

Manager	Responsibility
`ATGCommerceManager`	All commerce-related functions such as cart actions, billing, shipping, and checkout.
`ATGExternalProfileManager`	User-related functions such as login/logout and account management functions such as viewing and editing a profile.
`ATGInternalProfileManager`	This is the same as `ATGExternalProfileManager`, except that it operates against internal users (including agent), and the `ATGExternalProfileManager` operates against external users (shoppers).
`ATGLoginDelegate`	Provides callback for actions that require a login.
`ATGManagerRequest`	The base class for all manager requests.
`ATGProductManager`	Retrieves product information including inventory levels and registering for back-in-stock notifications. Also retrieves product comparison requests.
`ATGProfileManager`	User-related functions such as login/logout and account management functions such as viewing and editing a profile.
`ATGRestManager`	Manages connections to the server.
`ATGKeychainManager`	Saves and retrieves values from the iOS keychain.
`ATGStoreManager`	Retrieves store information.
`ATGAgentCommerceManager`	Agent-specific version of the commerce manager class.
`ATGAgentEnvironmentManager`	Manager for agent environment related server calls.
`ATGAgentEnvironmentManagerDelegate`	Delegates callback when the current catalog has been set.
`ATGAgentEnvironmentManagerRequest`	The results of requesting environment data.
`ATGAgentSiteManager`	Manages site related server calls.
`ATGAgentSiteManagerDelegate`	The callback delegate for site related server calls.
`ATGAgentSiteManagerRequest`	Request handler for `ATGAgentSiteManager`.
`ATGCustomerManager`	Manages customer related server calls.
`ATGCustomerManagerDelegate`	Callback delegate for customer related server calls
`ATGCustomerManagerRequest`	Request handler for `ATGCustomerManager`.

Each manager is implemented as a singleton. Consequently, `ViewControllers` use the appropriate manager's singleton instance to request data from the server. Most importantly, when the `ViewController` requests that the manager fetch data for it, it sets itself as the request delegate so that it receives the appropriate callback, either a success or failure block, when the response to the asynchronous request is available. When a response is returned from the server, the manager executes the success or failure block.

Each manager includes a delegate protocol definition that identifies those callbacks that should be invoked for each response from the server. In the case of the `ATGProfileManager`, this is the `ATGProfileManagerDelegate`. For example:

```
@protocol ATGProfileManagerDelegate <NSObject>
```

The delegate protocol defines callback methods for the manager. The protocol does not actually implement the method, rather they are implemented by other classes.

From within a delegate callback, the `ATGManagerRequest` handles the response by executing the success or failure block.

Index