argix-logo-rpts

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/24/2007 | 1.0 | Created | Jim Heary |
| 11/29/2007 | 1.1 | Updated | Jim Heary |
| 04/23/2013 | 2.0 | Updated | Jim Heary |
| 07/31/2013 | 3.0 | Updated | Jim Heary |
|  |  |  |  |

Table of Contents

Overview 4

Web Application 5

Overview 5

User Interface 5

Site Navigation 5

Content Management: Master Pages 7

Personalization: Profiles and Themes 7

Object Model 8

Object State 9

MembershipUser State 9

Session State 9

Configuration 10

Web Service 11

Overview 11

SOAP Envelope 11

Object Model 11

Configuration 12

Security 13

Overview 13

Authentication 13

Authorization 14

Web Application and Web Service 14

Tsort Database 14

Membership Database 14

Password Expiration 14

Deployment 15

Overview 15

Environment 15

Web Servers 15

Database Servers 15

Configuration 16

# Overview

Carton Tracking is a web-based application that provides Argix Logistics clients and vendors with services for tracking freight that is transported by Argix. The application includes a publicly accessible web application (<https://extranet.argixdirect.com/tracking>) and an associated web service (<https://extranet.argixdirect.com/tracking/carton.asmx>). The site is accessible to users from the Argix Logistics web site, <http://www.argixdirect.com>, by accessing the site and selecting the Tracking link.

**New Users (Guests)**

Upon entering the Tracking site, new users are directed to a Login page. From the Login page, new users select the option to register and are directed to a Registration page. When the registration form is completed and submitted, the user is redirected to a Confirmation page; they are then entered into the Membership database as a Guest user, and a registration receipt email is sent to them. The user cannot enter the Tracking site until the site administrator grants them membership.

The site administrator (or a user in the administrator role) enters the site and logs in. From the default tracking page, the administrator selects the Administrators Page link and is redirected to the Manage Users page. From the Manage Users page, the administrator selects Add User; if there are Guest users in the Membership database, the administrator is redirected to the Add User page. The administrator selects the registered user, and completes and submits the user profile form. Once submitted, the Guest user is converted to a Member user, their profile information is saved, and a welcome email is sent. The user can now enter the Tracking site as a registered user.

**Registered Users (Members)**

Upon entering the Tracking site, registered users are directed to a Login page. After submitting a valid username and password, the user is redirected to the default tracking page. From here, the user can track freight by carton# or Argix label#, or they can select the option to search for freight by store#.

**Password Expiration**

Users are forced to change their passwords every 60 days.

# Web Application

## Overview

The Carton Tracking web application is an ASP.NET 2.0 web application developed with Visual Studio 2005. The web application provides the following features:

* Carton tracking by carton#, Argix label#, or a search by store
* New user registration
* User password management
* User membership and profile management (administrators only)

## User Interface

### Site Navigation

The drawing below shows the web pages of the Carton Tracking application. The Default is a redirect page: Aeropostale clients are redirected to the Aeropostale page; all other clients are redirected to the Tracking page. The SearchByStoreSummary page includes an additional column with links to the CartonContents reports for Tommy Hilfiger clients only. The ChangePassword page is accessible for all users, and users are redirected to it when there password has expired.



### Content Management: Master Pages

ASP.NET master pages provide a consistent layout for the pages in the application. A single master page defines the look and feel and standard behavior for one, all, or a group of pages in the application. Master pages actually consist of two pieces, the master page itself and one or more content pages. Content pages contain application content embedded within the layout of the master page; when users request a content page, it is merged with the master page to produce output that combines the layout of the master page with the content from the content page.

Master pages can be set:

At the page level using the @page directive and masterpagefile attribute as follows:

<%@ page masterpagefile="~/App\_MasterPage/MasterPage.master" language="C#" %>

At the folder level using the <pages> directive and the masterPageFile attribute in the folders web.config file as follows:

<system.web><pages masterPageFile="~/App\_MasterPage/MasterPage.master”></pages></system.web>

At the application level by using a configuration entry in the application’s main (root) web.config file.

Carton Tracking uses the following master page, MasterPage.Master. This page includes:

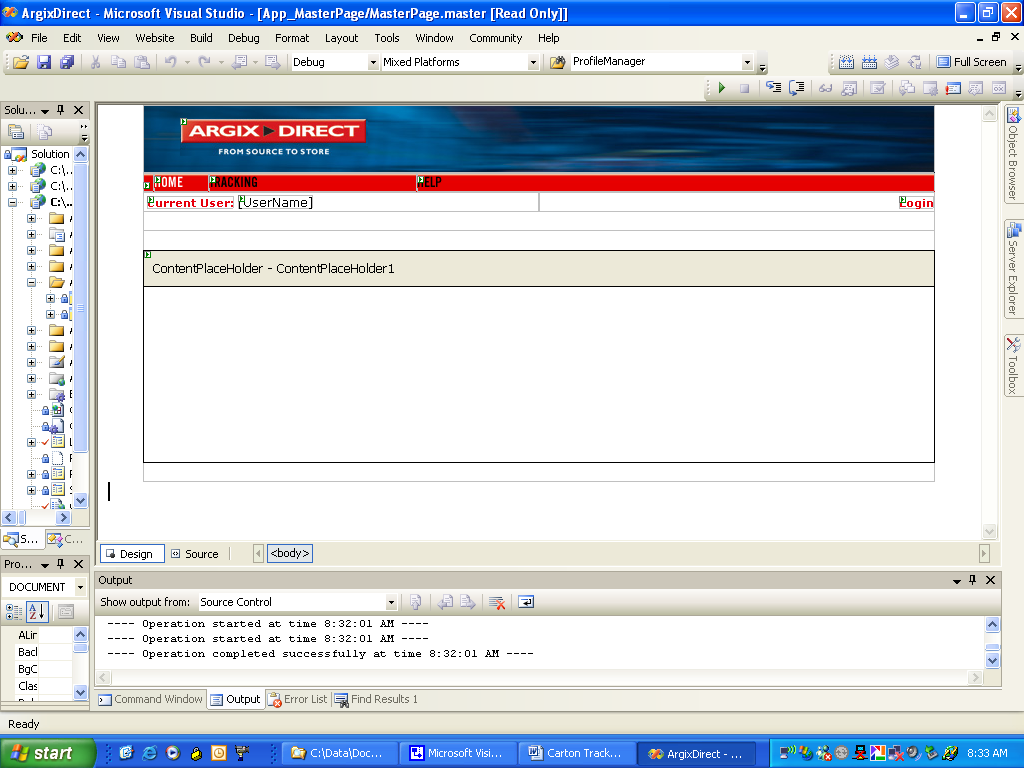
LoginName control- Displays the value of the System.Web.UI.Page.User.Identity.Name property.

LoginStatus control- detects the user's authentication state and toggles the state of the link to log in to or log out of the web application.

ContentPlaceHolder control- defines a region for content in the master page.

Master pages provide code reuse; public properties and methods defined in the master page are accessible to content pages through the *Master* property (type MasterPage). To access custom properties and methods, cast the Master object as follows:

((MyMaster)Master).CustomPropertyGet = “value”;



### Personalization: Profiles and Themes

Personalization is not used in the Carton Tracking application.

## Object Model

The following drawing shows a model of the application classes and their relationships. The following classes are of interest:

TrackingServices.cs- provides carton tracking data services.

MembershipServices.cs- provides membership services including users, roles, and user profiles; this class provides a simplified interface of the System.Web.Security and System.Web.Profile classes.

EMailServices.cs- provides email services using SMTP via the System.Net.Mail classes.



## Object State

### MembershipUser State

Tracking non-members access the Tracking application as anonymous users. After submitting a registration from Register.aspx,



### Session State

The application relies on Session state for messaging and data caching. The following drawing shows where session state is created, read, updated, and deleted by the classes of the application.



## Configuration

Application Settings in Web.config:

* ConnectionStringName=MembershipDB- used by Membership system to connect to the membership database.
* defaultDatabase=TsortDB- used by the Enterprise Library for data access.
* Both connection strings need LAN IP address of the database server, since there is no LAN name resolution within the DMZ.

# Web Service

## Overview

The Carton Tracking web service is a publicly accessible web service that provides the ability to track a single carton. A request for a single carton number returns information about the carton, pickup, client, vendor, PO, TL, BOL, departure, arrival, and POD.

Barnes & Noble is using the service with TsortEnterpriseMembership account bnwsuser\bn2004.

## SOAP Envelope

The web service has one public method- TrackCarton() accepts a carton number and returns a dataset containing tracking details. The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

POST /Tracking/carton.asmx HTTP/1.1

Host: extranet.argixdirect.com

Content-Type: application/soap+xml; charset=utf-8

Content-Length: **length**

<?xml version="1.0" encoding="utf-8"?>

<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">

<soap12:Header>

<SoapCredential xmlns="https://extranet.argixdirect.com/Tracking/">

<UserName>**string**</UserName>

<Password>**string**</Password>

</SoapCredential>

</soap12:Header>

<soap12:Body>

<TrackCarton xmlns="https://extranet.argixdirect.com/Tracking/">

<cartonNumber>**string**</cartonNumber>

</TrackCarton>

</soap12:Body>

</soap12:Envelope>

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: **length**

<?xml version="1.0" encoding="utf-8"?>

<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">

<soap12:Body>

<TrackCartonResponse xmlns="https://extranet.argixdirect.com/Tracking/">

<TrackCartonResult>**dataset**</TrackCartonResult>

</TrackCartonResponse>

</soap12:Body>

</soap12:Envelope>

## Object Model

The drawing below shows the object model for the web service. The following classes are of interest:

Carton.asmx- provides a web service front-end to the Carton Tracking object model; inherits from System.Web.Services.WebService;

SoapCredential.cs- SOAP headers offer a method for passing data to and from an XML Web service method if the data is not directly related to the XML Web service method's primary functionality; this class defines what to expect in the SOAP request header for credentials (i.e. username, password) so that they don’t need to be included as parameters for each public method. An instance is created by the Framework at runtime by adding the [SoapHeader("<memberName>")] attribute to the TrackCarton() method.

Once the credentials of the requestor are validated (i.e. administrators, wsmembers), the web service returns tracking details for the carton as described by the xml schema in the CartonWSDetail dataset. Only the classes required for the web service are shown in this drawing.



## Configuration

Currently there is no administrative functionality for creating web-service users. Users can be created using the Register.aspx page and approved by mapping them to a client id. Then using .NET Admin Configuration from VS2005, web-service user can be moved to wsMembers role. All web-service users should belong to wsMembers role.

Consumers of this web-service can get web-service description and XML Schema from <https://extranet.argixdirect.com/tracking/carton.asmx> URL.

Users will to pass valid credentials programmatically using SoapCredential class before they can call TrackCarton() method.

# Security

## Overview

The drawing below illustrates the considerations for user authentication and authorization. Users access the web application anonymously, but must login to access the web pages or web service; all users can access the Login page in the root directory. Once authenticated by the Membership services, users can access web content when authorized by the role-based security. Data is accessed by the application on behalf of the users using integrated security (web server local user account Tracker (password=) is passed to SQL Server databases and verified against SQL database account Tracker).



## Authentication

Users are authenticated via Form-based authentication using non-persisted cookie with a session timeout of 10 minutes as specified by the authentication entry in the applications web.config file:

<authentication mode="Forms">

<forms name=".ASPXFORMSAUTH" loginUrl="Login.aspx" defaultUrl="Default.aspx" timeout="10" />

</authentication>

* In IIS, the virtual directory Tracking is setup to use Anonymous user access over SSL; users are authenticated to the local IIS anonymous account (WWW1\IUSR\_WWW1, WWW2\IUSR\_WWW2)
* The application accesses the Tsort database on RGXSQL using integrated security by passing the local server account WWW1\Tracker (WWW2\Tracker) in its’ connection requests
* The application accesses the TsortEnterpriseMembership database on RGXSQL by passing the local server account WWW1\Tracker (WWW2\Tracker) in its’ connection requests

## Authorization

### Web Application and Web Service

The Tracking web application and web service uses role-based security for authorization to application resources. Web application users login and are authorized by the ASP.NET 2.0 Membership system; web service connecting applications pass user credentials in the SOAP header and are also authorized by the ASP.NET 2.0 Membership system. External (clients, vendors) users are in the role of Member. Internal (Argix) users are in the role of Member, but can also have Administrator, RSMember, and WSMember roles. Web service users must have the WSMember role to access the methods of the web service.

Authorization to web pages is granted when the users’ roles match the roles specified in the web.config file of the folder containing the web page. A typical entry looks like the following:

<system.web>

<authorization>

<allow roles="administrators, rsmembers, members" />

<deny users="\*"/>

</authorization>

</system.web>

**Note**: It is important that the allow roles should be specified first.

Application users, roles, and access rights can be managed by using the ASP.NET Web Site Administration Tool. This utility is accessible from the Website\ASP.NET Configuration menu item in Visual Studio 2005. The tool will open the membership database pointed to by the membership database connection string specified in the applications web.config file:

<add name="MembershipDB" connectionString="Data Source=RGXDEVDB; Initial Catalog=TsortEnterpriseMembership; Persist Security Info=True; User ID=; Password=;" providerName="System.Data.SqlClient" />

### Tsort Database

The Tsort database uses integrated security. Users access the database as database user account Tracker. Tracker account has login permissions and following authorization;

* Carton Tracking stored procedures- execute permissions

### Membership Database

The TsortMembershipEnterprise membership database uses role-based security. Users access the database as user account Tracker. Tracker account has login permissions and is added to the following database roles:

* Aspnet\_Membership\_FullAccess
* Aspnet\_Profile\_FullAccess
* Aspnet\_Roles\_FullAccess

These roles have preconfigured access to the membership stored procedures with access rights as specified by each role.

External (clients, vendors) and internal (Argix) users belong to Members membership role. The system determines internal users if they belong to Argix Logistics (profile: company code=00).

### Password Expiration

Users are forced to change their passwords after every 60 days. This value is configurable in the web.config. If user’s password is manually reset by an admin person, they are forced to change the password on their first login attempt. This is controlled through the PasswordReset flag property in user’s profile.

# Deployment

## Overview

The application is deployed per the drawing below. Users access the web application from a browser the same as any other web application; the ArgixDirect public servers are contained in a DMZ outside of the Argix LAN. The web application and web service access the LAN database using OLEDB; ODBC ports must be open in the Argix LAN firewall.



## Environment

### Web Servers

* Cluster of two Windows Server 2003 servers- WWW1 (administrator/jamesburg100), WWW2 (administrator/jamesburg100)
* IIS 6.0, ASP.NET 2.0, .NET Framework 2.0, Infragistics 2003 V3 IIS Scripts, Extranet.Argixdirect.com SSL Certificate
* Application pool- ASPNET2AppPool: ASP.NET 2.0 pool; runs under process identity of local user account Tracker
* User accounts- WWW1\Tracker (password=tracker\_rgx): local account; member of IIS\_WPG (worker process group)
* IIS virtual directory- Tracking virtual directory points to CartonTrackingV2 physical directory under wwwroot; configured to use SSL and anonymous access only.

### Database Servers

* Databases- RGXSQL server; Tsort: data database; TsortMembershipEnterprise: membership database
* User accounts- RGXSQL\Tracker (password= tracker\_rgx); password must match WWW1\tracker password
* Stored procedures: uspTracking…, some uspRpt…; user account **Tracker requires Execute permissions** on all stored procedures used by the application

## Configuration

Application

<!-- APPLICATION SPECIFIC -->

<appSettings>

<add key="maxCartonSearchLimit" value="10" />

<add key="maxCartonSearchCharLimit" value="400" />

<add key="minCartonLen" value="6" />

<add key="maxCartonLen" value="30" />

<add key="LabelSequenceLen" value="13" />

<add key="FromDateMaxAge" value="30" />

<add key="DeliverySearchMaxInFuture" value="7" />

<add key="SearchRangeMaxDays" value="7" />

<add key="PasswordExpiration" value="60" />

<add key="TrackingRole" value="members" />

<add key="TrackingWSRole" value="wsmembers" />

<add key="AdminEmail" value="jheary@argixdirect.com" />

<add key="SMTPServer" value="192.168.151.5" />

<add key="SendMailFrom" value="jheary@argixdirect.com" />

<add key="UseWebSvc" value="false" />

</appSettings>

<!-- DATA ACCESS -->

<dataConfiguration defaultDatabase="TsortDB" />

<connectionStrings>

<add name="MembershipDB" connectionString="Data Source=192.168.151.33;Initial Catalog=TsortEnterpriseMembership;Integrated Security=SSPI;Connect Timeout=30;" providerName="System.Data.SqlClient" />

<add name="TsortDB" connectionString="Data Source=192.168.151.33;Initial Catalog=TSORT;Integrated Security=SSPI;Connect Timeout=30;" providerName="System.Data.SqlClient" />

</connectionStrings>

Security

Membership

<!-- MEMBERSHIP CONFIGURATION - expression "^[^.\^{[(|)]}\* > + ? / \\:;%!=]\*$" -->

<membership defaultProvider="TrackingMembershipProvider">

<providers>

<add name="TrackingMembershipProvider" type="System.Web.Security.SqlMembershipProvider, System.Web, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" connectionStringName="MembershipDB" applicationName="Tracking" enablePasswordRetrieval="true" maxInvalidPasswordAttempts="5" passwordAttemptWindow="10" requiresQuestionAndAnswer="false" requiresUniqueEmail="false" passwordFormat="Encrypted" minRequiredNonalphanumericCharacters="0" minRequiredPasswordLength="6" passwordStrengthRegularExpression="" />

</providers>

</membership>

<!-- ROLE COMFIGURATION -->

<roleManager defaultProvider="TrackingMembershipProvider" enabled="true" cacheRolesInCookie="true" cookieName=".ASPROLES">

<providers>

<add name="TrackingMembershipProvider" type="System.Web.Security.SqlRoleProvider" connectionStringName="MembershipDB" applicationName="Tracking" />

</providers>

</roleManager>

<!-- PROFILE COMFIGURATION -->

<profile defaultProvider="TrackingMembershipProvider" enabled="true">

<providers>

<clear/>

<add name="TrackingMembershipProvider" type="System.Web.Profile.SqlProfileProvider" connectionStringName="MembershipDB" applicationName="Tracking" description="SqlProfileProvider for Carton Tracking application"/>

</providers>

<properties>

<add name="Company" type="string" allowAnonymous="false" />

<add name="UserFullName" type="string" />

<add name="Type" type="string" />

<add name="ClientVendorID" type="string" />

<add name="PasswordReset" type="boolean" />

<add name="WebServiceUser" type="boolean" />

</properties>

</profile>

<!-- Required for storing Profile for anonymous users -->

<!-- <anonymousIdentification enabled="true" /> -->