Dues, Events, Orders, Oh My… Webservices for All!

Tim Sloan

Randy Richter

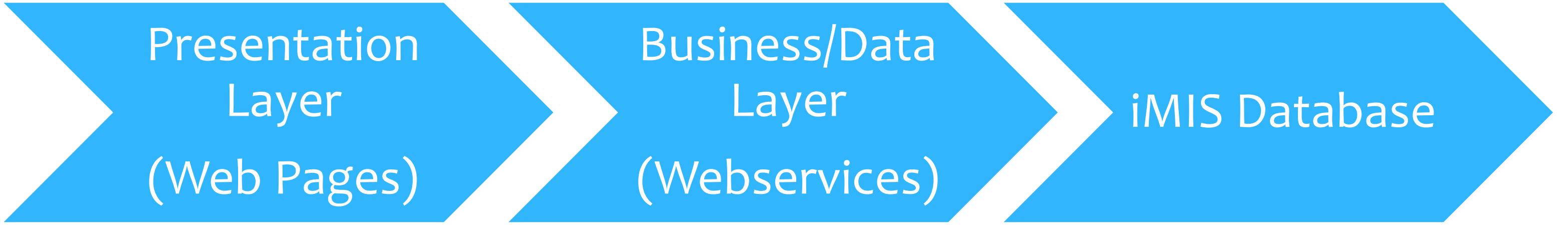
we can solve it.

Getting Data In and Out of iMIS

- SQL Reads and Writes
  - Not sustainable
  - May break applications with upgrades
  - May violate license agreement
- iMIS API’s
  - Upgradable
  - Scalable

Application Layer

- Separate Presentation Layer from Business and Data Layer
  - No direct SQL calls or API calls in your website pages
  - Works with single or multiple server topologies



Presentation Layer

(Web Pages)

Business/Data Layer

(Webservices)

iMIS Database

History of iMIS APIs

- COM ibo’s
  - Worked with iMIS version 10.x
  - Late 1990’s
- .NET ibo’s
  - Work with all versions of iMIS from 10.x to 20.x
- SOA Webservices
  - Generally based on .NET ibo’s
  - Still have some limited functionality

ats

we can solve it.

History of ATS Webservices

- Similar code was being written for similar projects (e.g., member directory)
- Needed something upgradable that worked when clients upgraded iMIS and the underlying data structure changes.
- Began building them in 2007
- Used in every ATS project
- Based on the iMIS Business Objects (ibo’s)
- Thousands of hours of development time

ats

we can solve it.

Goals

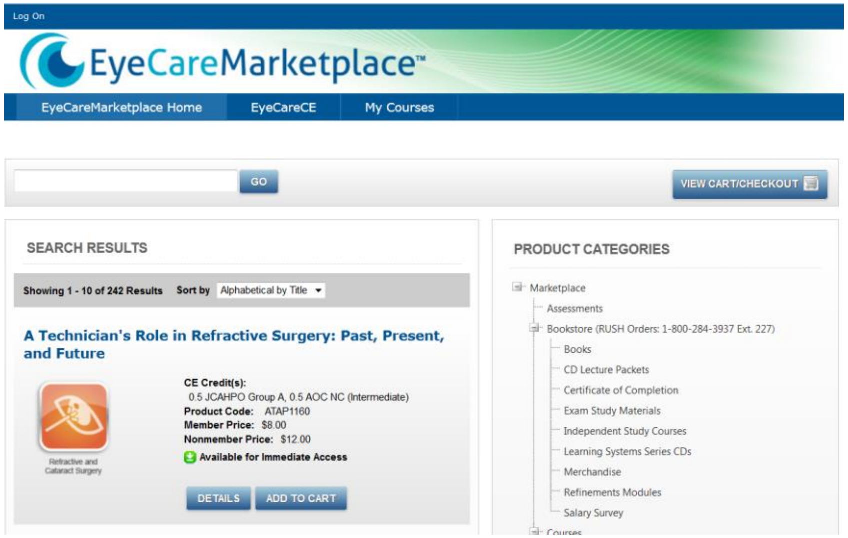
- Provide a consistent way for developers to call the underlying IBO
- Allowed us to address bugs in IBO’s and compensate for them by making the appropriate changes in the code
- Allowed us to account for differences in versions of iMIS (e.g. some iMIS versions require you to set a regclass BEFORE overriding the price and other versions require you to override the price BEFORE setting the regclass)
- One place for all batch creation, credit card charging calls.

ats

we can solve it.

Types of Projects

- ATS Online Marketplace
- Profile Management
- Bridges to third party applications
- Highly customized Forms



SEARCH RESULTS

Showing 1 - 10 of 242 Results Sort by Alphabetical by Title

A Technician's Role in Refractive Surgery: Past, Present, and Future

CE Credit(s):
0.5 JCAHPO Group A, 0.5 AOC NC (Intermediate)
Product Code: ATAP1160
Member Price: $8.00
Nonmember Price: $12.00
Available for Immediate Access

PRODUCT CATEGORIES

- Marketplace
  - Assessments
  - Bookstore (RUSH Orders: 1-800-284-3937 Ext. 227)
    - Books
    - CD Lecture Packets
    - Certificate of Completion
    - Exam Study Materials
    - Independent Study Courses
    - Learning Systems Series CDs
    - Merchandise
    - Refinements Modules
    - Salary Survey
  - Courses

ats

we can solve it.

What the Webservices Are

- A set of .NET SOAP based webservices
- Can be called from any programming language with POST/GET calls
- Provide success/failure responses with error messages
- Can be locked down through IIS and/or through the use of an Authentication Token for SQL calls
- Are separately licensed by Suite

ats

we can solve it.

ATS Webservice Suites

- Contacts – includes the ability to get and put contact information including names, addresses, user defined fields, activities, relationships, etc. as well as Login
- Dues – includes the ability to get information about subscriptions and dues products and pay dues
- Orders – Includes the ability to get information about products, freight, shipping, and place orders
- Events – Includes the ability to get information about an event and register people for an event
- Donations – Includes the ability to get fundraising information and process donations

ats

we can solve it.

ATS Webservices

Individual Pre-Built Webservices for Third Party Platforms

- Authorize.NET
- Beacon Live
- Blue Sky Broadcast
- Boxwood
- CadmiumCD
- CDS
- CECity
- CommPartners
- CQRC Engage
- Digital Ignite
- Digitec
- Drupal
- Elogic
- Eshow
- Experient
- GoToTraining
- GoToWebinar
- MapYourShow
- Moodle
- Omnovia
- Peach New Media
- Powered By Dub
- Ready Talk
- Real Magnet
- Refined Training
- Scholar Lab
- Sitefinity
- Small World Labs
- Socious
- Walsworth
- WebCourseWorks
- Many More in Development!

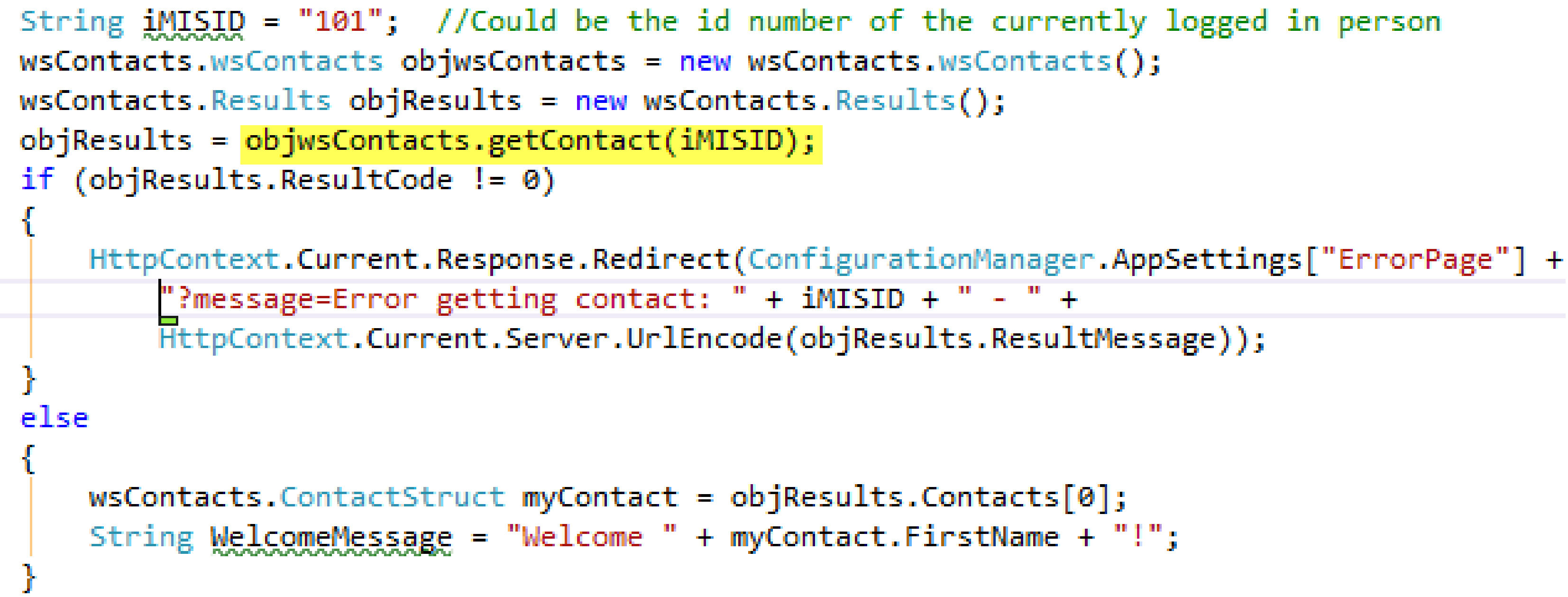
we can solve it.

Example Application Development

- Create a custom membership directory
  - Create a .NET application in Visual Studio Application
  - Add a reference to the webservices
  - Create object for the webservice
  - Create object to hold webservice response
  - Call the webservice
  - Analyze the results

we can solve it.

Sample Webservice Call

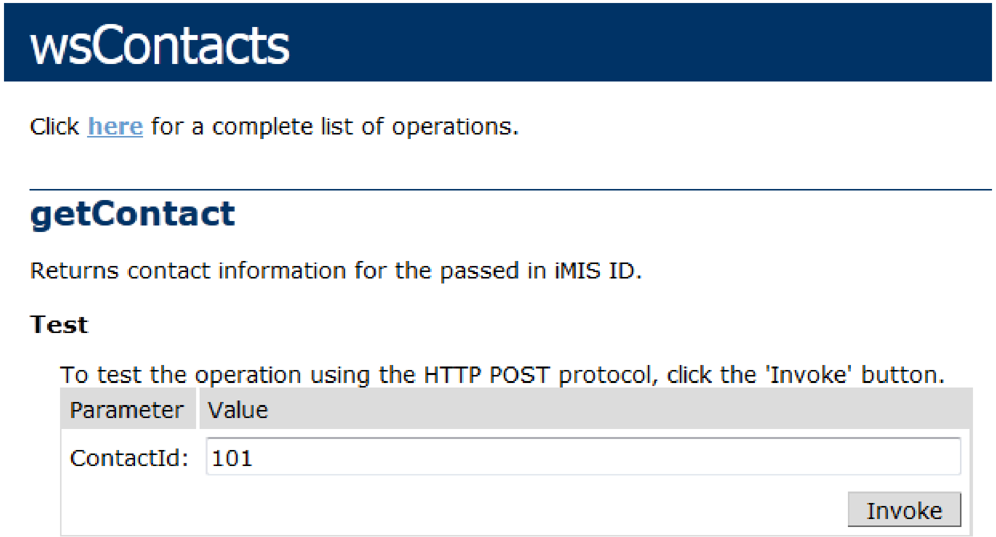


```
String iMISID = "101";  //Could be the id number of the currently logged in person
wsContacts.wsContacts objwsContacts = new wsContacts.wsContacts();
wsContacts.Results objResults = new wsContacts.Results();
objResults = objwsContacts.getContact(iMISID);
if (objResults.ResultCode != 0)
{
    HttpContext.Current.Response.Redirect(ConfigurationManager.AppSettings["ErrorPage"] +
        "?message=Error getting contact: " + iMISID + " - " +
        HttpContext.Current.Server.UrlEncode(objResults.ResultMessage));
}
else
{
    wsContacts.ContactStruct myContact = objResults.Contacts[0];
    String WelcomeMessage = "Welcome " + myContact.FirstName + "!";
}
```

we can solve it.

Sample Methods

- Information on the ATS Website at http://www.atsol.org/
- Demo calls at http://demo.atsol.org/WebServices.aspx



wsContacts

Click here for a complete list of operations.

getContact

Returns contact information for the passed in iMIS ID.

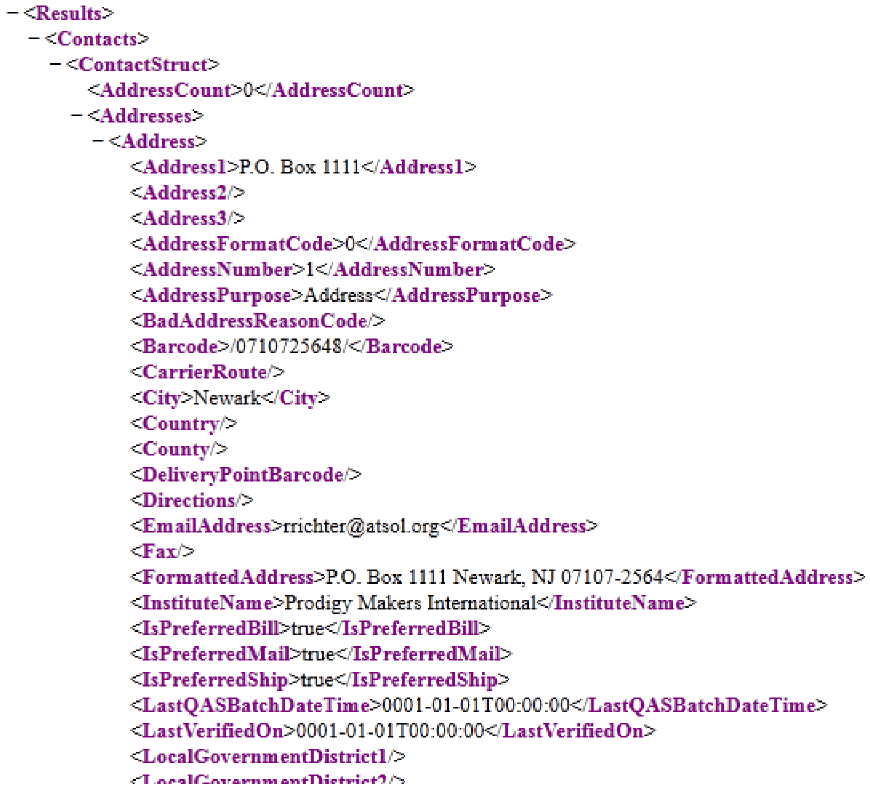
Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
ContactId:	101

we can solve it.

Sample Results



```
– <Results>
  – <Contacts>
    – <ContactStruct>
        <AddressCount>0</AddressCount>
      – <Addresses>
        – <Address>
            <Address1>P.O. Box 1111</Address1>
            <Address2/>
            <Address3/>
            <AddressFormatCode>0</AddressFormatCode>
            <AddressNumber>1</AddressNumber>
            <AddressPurpose>Address</AddressPurpose>
            <BadAddressReasonCode/>
            <Barcode>/0710725648/</Barcode>
            <CarrierRoute/>
            <City>Newark</City>
            <Country/>
            <County/>
            <DeliveryPointBarcode/>
            <Directions/>
            <EmailAddress>rrichter@atsol.org</EmailAddress>
            <Fax/>
            <FormattedAddress>P.O. Box 1111 Newark, NJ 07107-2564</FormattedAddress>
            <InstituteName>Prodigy Makers International</InstituteName>
            <IsPreferredBill>true</IsPreferredBill>
            <IsPreferredMail>true</IsPreferredMail>
            <IsPreferredShip>true</IsPreferredShip>
            <LastQASBatchDateTime>0001-01-01T00:00:00</LastQASBatchDateTime>
            <LastVerifiedOn>0001-01-01T00:00:00</LastVerifiedOn>
            <LocalGovernmentDistrict1/>
            <LocalGovernmentDistrict2/>
```

we can solve it.

Licensing

- Licensed per SUITE, Contacts, Dues, Orders, Events, Fundraising ($2500 each)
- Individual bridges to third party platforms are licensed separately (Price based on # of iMIS users)
- If you are licensed for the ATS Online Marketplace you are automatically licensed for the corresponding webservices suites.
- Full support is included for first year and is optional year 2 and beyond for additional SU fee.
  - Includes any bug fixes
  - New functionality/methods
  - Versions to match iMIS upgrades

ats

we can solve it.

Installation and Support

- Typically scheduled in 1-2 weeks
- WSDL definitions are very complete and include descriptions
- Clients often reserve 4 hours for application support if needed to help with asking what methods to call in what order (for example)
  - Create an order and let iMIS calc Freight, Tax, and Handling
  - Collect and authorize a credit card
  - Create the Batch
  - Place the Order
  - Delay Capture the Card

ats

we can solve it.

Thank You

Tim Sloan – tsloan@atsol.org
Randy Richter – rrichter@atsol.org

http://www.atsol.org/

we can solve it.