



ISHOP

Functional Implementation Guide

Version 3



Revision History

Revision	Revision Description	Revision Date
3.0	Initial Release	July, 2006
3.0.1	<ul style="list-style-type: none"> • All Information Server basic operation request messages now contain the following elements in sequence after message-specific elements: Vehicle, UserArea. Basic operations do not include Line Filtering operations (i.e. operations beginning with “LF”). Changes made from 3.0 are as follows: <ul style="list-style-type: none"> ○ Vehicle element resequenced for messages: OpenRequest ○ Optional Vehicle elements added to messages: GetNavigationPathRequest, PopNavigationRequest, GetInformationTypesRequest, SetLineFilteringRequest, SetLineFilteringConfigurationRequest. • All Information Server basic operation response messages now contain the following elements in sequence after message specific elements: Vehicle, Response, UserArea. Basic operations do not include Line Filtering operations (i.e. operations beginning with “LF”). Changes made from 3.0 are as follows: <ul style="list-style-type: none"> ○ Vehicle, Response elements resequenced for messages: GetNavigationPathResponse, SetInformationTypesResponse. ○ Optional Vehicle elements added to messages: GetInformationTypesResponse, SetLineFilteringResponse, SetLineFilteringConfigurationResponse. • New element NavigationResultsURL added to the NavigationResults element of the Information Server NavigateResponse message. This new element is therefore also available in the SearchResponse and PopNavigationResponse messages. 	July, 2008

	<ul style="list-style-type: none"> • Documentation of URL elements in Information Server response messages updated to reflect new NavigationResultsURL element and to include other clarifications. • New elements “Action” and “Note” added to the MaintenanceInformation element of the Information Server NavigateResponse message. This new element is therefore also available in the SearchResponse and PopNavigationResponse messages. 	
--	---	--

Prepared by: Dan Harrel, Prescient Technologies Inc.

Contributors: Dave Callingham, Prescient Technologies Inc.; Michael Lucas, Prescient Technologies Inc. and Russ Bailey, Autovoice Consulting

Applicable content has been taken from:

- Design documents from Mark Erdrich, ALLDATA/AutoZone
- The iShop Version 2.0 specification document by Ben Johnson and Jim Mikesell
- Auto Care Association documentation and web resources

Notice

The Auto Care Association, formally Automotive Aftermarket Industry Association (AAIA), makes no warranty of any kind regarding this material, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. Auto Care Association shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

©Copyright 2004 - 2006 Auto Care Association

This document contains proprietary information, which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of the Auto Care Association. The information contained in this document is subject to change without notice.

Table of Contents

Revision History.....	1
Introduction.....	5
History.....	5
Objectives.....	6
How this document was developed.....	8
Maintaining the Standards.....	8
Contact Information.....	10
Auto Care Association iShop Web Site.....	10
Certification Requirements.....	11
Document Outline.....	11
Related Documents.....	12
Shop Management Server.....	12
Introduction.....	12
Operations.....	18
XML Structures Reused in SMS.....	43
Information Server.....	56
Introduction.....	56
Operations.....	69
Information Server Reused Structures.....	121
Global XML Structures.....	123
UserArea.....	123
Response.....	123

Vehicle	125
Appendices	132
Appendix A – Standard Messages	132
Appendix B – Order Search Fields.....	133
Appendix C – Note Fields	136

Introduction

Welcome to Auto Care Association's iShop integrated shop standards! Shop Integration Standards enable computer based diagnostic and repair equipment and information servers to share data about the customer, vehicle, work order, service and repair history, as well as parts and labor information seamlessly throughout the service environment.

This document describes iShop Version 3 which is based on XML Web services technology.

The iShop standard covers the following systems used within a service facility:

- Information Server serving parts, labor, service and repair information. These servers may operate within the facility or be accessed over the internet. Access may be subject to license and subscription.
- Shop Management Server serving customer, vehicle, and task information, and consuming data from Information Servers.
- Back shop equipment communicating with a Shop Management Server, and also consuming data from Information Servers.

This document was written to assist companies wanting to write software that operates within an iShop conformant environment, whether such software consumes or provides iShop services. It addresses functional considerations, and should be read by any party requiring an understanding of iShop interfaces and processes. Technical issues are described in a separate document: "Auto Care Association iShop Technical Implementation Guide".

History

Shop integration emerged and evolved as an important tool to improve shop operations throughout the 90's. In September 1999 the major providers of integrated shop systems joined forces to define an open standard of communication between front and back shop applications. Auto Care Association was chosen as the facilitative group for these standards, and was appointed the responsibility of managing and governing the project. iShop was chosen as the name for this standard.

A Steering Committee was established, with members of both the Enterprise Alliance and SnapOn, to provide governance and strategic management of the iShop Standard.

A Shop Integration Task Force was created with technical representation of the companies involved, to develop the specifications that make up the standard and implement the standard into their company's products.

In the summer of 2004, the iShop Technical Team held a meeting to discuss the future of the iShop specifications relative to the current strengths and weaknesses of Version 2. The Technical Team developed a recommend strategy to rewrite the current Version 2 specifications that would address the current weaknesses of the specifications plus improve the documentation and provide sample code. In the fall of 2004 the iShop Management Team held a meeting to review the Technical Team recommendations and subsequently approved the strategy to develop a new version of the specification designated as Version 3.

The goals and objectives of the Version 3 specification were to:

- Replace COM/DCOM with a non-proprietary, simpler, and more current technology: XML Web services.
- Move the specification forward to embrace non PC based products typically described as handheld units with the implementation of XML.
- Address communication and information sharing beyond the walls of the independent automotive service provider.
- Significantly improve the specification documentation.
- Provide sample code and examples to assist new iShop adopters.
- Provide a Version 2 to Version 3 bridge for ALLDATA SMS (Shop Management Systems) Partners that have adopted the Version 2 iShop Standards.

In February of 2005 the Technical Team held a meeting to initiate the development of the Version 3 iShop specification. The Team held conference calls and selective meetings through 2005 to develop and finalize the specification.

Objectives

In short, the objectives of iShop is to develop a single, open, worldwide standard for the integration and interconnectivity of computer-based shop equipment utilizing standards-based XML Web service technology.

iShop – Functional Implementation Guide

The results of these objectives are the guidelines that developers from disparate companies can use to create a standard communication architecture that enables an integrated vehicle service facility.

Therefore, it is appropriate that we define here what an integrated service facility is:

- An integrated service facility can share information about the customer, the vehicle, the work order, the work performed, and test results between the shop management system and all diagnostic test equipment and information servers.
- An integrated service facility can send and receive existing data without the need to re-key from any work station.

iShop focuses both on the internal network within the service facility, such as interfacing back shop equipment with Shop Management Systems, and on interfaces external to the service facility via the Internet, such as interfacing back shop equipment and Shop Management Systems with external Information Servers. The diagram below depicts the integrated shop that iShop envisions.



Figure 1 - Integrated Shop

While iShop currently is concentrating on North American companies, a future implementation will include an information gateway to the European connectivity standard, Asanetwork. This will further iShop's connectivity in the global marketplace.

The iShop standards are intended to address the widest variety of developers. Now that iShop 3 is based on standard XML Web service technology, the iShop 2 constraint requiring Microsoft Windows operating system platforms has been eliminated.

How this document was developed

This document represents the third iteration of the iShop standard. The iShop 2 general design served as a starting point, and in fact much of the material in this document was taken or adapted directly from the iShop Version 2.0 Specifications document. Changes from iShop 2 occurred either out of necessity to migrate from a Microsoft COM/DCOM/ActiveX architecture to XML Web services architecture, or as a result of lessons learned by iShop 2 implementers.

A working group of interested representatives from Auto Care Association member organizations steered the evolution from iShop 2 to iShop 3 through a series of prototypes and conference calls. This culminated in a face-to-face meeting to work out and agree to a design. Draft versions of this document were produced from the design, and were subsequently reviewed and iterated upon by the working group until a final version was accepted.

This document restricts itself to functional documentation. It is accompanied by a Technical Implementation Guide documenting technical issues. Code samples are provided with the iShop 3 standard to demonstrate material from both functional and technical documents.

Maintaining the Standards

Auto Care Association provides Facilitation and Project Management of the iShop Standards as directed by the iShop Management Committee.

Auto Care Association's facilitative duties include:

- Handling of requests to modify the current Standards documents as well as requests to further enhance the Standards.
- Coordination of other industry standard initiatives with all committees reporting to the Auto Care Association Standards and Technology Committee. This helps insure that standards being developed across a very diverse industry are complimentary where appropriate and do not conflict with one another.

- Marketing and promotion of the iShop brand and standards to the Automotive Service Industry.

Auto Care Association's Project Management responsibilities include:

- Create and maintain a project plan per the scope of work and resources identified to perform the scope of work. Depending on the situation, the resources may be volunteered by iShop Participant Companies, or may be contracted directly by Auto Care Association.
- Facilitate development and management of the Project Life Cycle incorporating future releases of enhancements to the Specifications.
- Facilitate regular Project Meetings and Conference Calls as required, and compile and report progress and outstanding issues relative to the project underway.
- Manage the development and distribution of the iShop Software Developer's Kit.
- Manage the iShop Product Approval Process to validate that proposed products meet minimum iShop requirements.

See section 1.5 or check the iShop web site for the assigned Project Manager's contact information.

Shop Integration Task Force

The Shop Integration Task Force (SITF) consists of iShop participants who are assisting in the definition, development, prototyping and testing of the specifications. The SITF are the architects of the various functions within iShop. Therefore, the group participants change as the requirements for development of new features changes. Any iShop participant who is interested in working in a team to determine appropriate ways to implement new features or debug problems as they are identified with the standard is welcome to join the SITF.

iShop Meetings

The Shop Integration Task Force (SITF) has meetings that are scheduled in accordance to the requirement for the meeting or the status of the project(s) currently being undertaken. Conference calls are held weekly during times of specification development. In these meetings proposals are considered for changes to the specification (i.e. bugs) as well as new features to enhance the Standard. The meeting information and topics for discussion are posted on the iShop website in advance of the meeting. The SITF maintains a project page on the Auto Care Association GroupHub website as the iShop Technical Subcommittee at <http://aftermarket.grouphub.com>, where meeting minutes and technical documents are maintained. A face-to-face meeting is often scheduled during development of the specification to accelerate the completion of the standard. The iShop 3 developers held a face-to-face

meeting in Toronto in September of 2005. At that meeting, a detailed comparison to iShop 2 was undertaken for the SMS and IS Web services.

Document Change Process

The iShop Website (see page 13) contains instructions for reporting any issues with the specification that you may feel are inaccurate or need to be modified. You will receive a response to your request within a reasonable amount of time from date of submission.

iShop Standard Enhancements

The iShop Standard is designed to be an evolutionary standard, realized ultimately through the completion of multiple projects. As technology or industry requirements are identified, either by Auto Care Association or its membership, that are deemed by the iShop Management Committee to be beneficial to the Standard, the new requirements will be compiled in a future version of the specification document to enhance the Standard.

The iShop Website (see page 13) contains instructions for requesting iShop Specification enhancements. Any participant company will be able to submit a request for consideration of a feature not currently included in the specification.

Backwards Compatibility Statement

As noted in section 1.4.4, the iShop project is an evolving standard. The current released version of the specifications are the specifications that products will be tested against to certify them as iShop conformant. Reasonable effort will be made when considering enhancements to the standard to allow backwards compatibility. However, Auto Care Association makes no guarantee that products certified conformant to a previous version of iShop will continue to operate as intended if they are not updated to support the current specifications. Specifically, iShop 3 is not backward compatible with iShop 2. The most current released specification is available on the iShop website for member download.

Contact Information

Questions regarding this document or any iShop technical specification should be directed to:

technology@autocare.org

Auto Care Association iShop Web Site

The Auto Care Association iShop web site is found at www.ishopportal.org. Auto Care Association maintains password protected content at this web site exclusively for iShop participants. A “participant” is defined as any representative of a company that has paid the annual dues required. A participant is

entitled to any and all documentation, software development tools and contributed source code assuming they have signed all appropriate intellectual property agreements. A participant is welcome and invited to attend the regular Shop Integration Task Force Meetings and is encouraged to participate and contribute to the continuing development of the iShop Specifications.

This website contains information regarding upcoming meetings, electronic copies of the current specification, and a Discussion area where participants can browse issues that have been raised and resolutions to those issues by developers implementing the standard. Participants will also have access to meeting minutes generated at the regular iShop meetings, and to the contact numbers for participants of the Shop Integration Task Force.

Access the website using the login information given to you when you became in iShop Participant.

Certification Requirements

All products will be validated to insure they meet defined minimum requirements to be iShop conformant. Refer to the Certification document available on the The iShop Website for details of the process for validating a product.

Display of iShop Certification Status

Applications that display a user interface (typically client applications) will display as part of their “startup” screen, the iShop logo. This will indicate that the product has been certified iShop compliant. Further, they will include in their “information” area, along with their version number and other pertinent information, which version of the iShop standard they are certified as conformant.

Server applications return the iShop version number in their response to the GetAdmin operation (iSHOPVersion element).

Document Outline

This document is organized in two main sections:

1. Shop management Web service - an iShop conformant shop management system exposes a Web service with operations such as CreatePhoto and GetCustomer. These operations are typically invoked by backshop equipment in order to provide standardized access to the shop management functionality. This section describes these operations in detail in terms of the expected xml request and response messages.
2. Information Server Web service - an iShop conformant information server exposes a Web service with operations such as Navigate and SetLineFiltering that allows Shop Management and Back

Shop equipment to interact with the information server. This section describes these operations in detail in terms of the expected xml request and response messages.

Related Documents

In addition to this document, readers may wish to consult the following:

- Auto Care Association iShop Technical Implementation Guide. This document contains technical details on implementing an iShop solution.
- ACES Delivery Specifications, version 1.07. This document describes the Auto Care Association Enhanced Vehicle Database and its XML conventions that are used within iShop to identify and specify vehicles.
- Auto Care Association Imaging Best Practices, October 2005. This document may be useful with respect to photos stored into and retrieved from an iShop SMS. It is available online at: http://www.aftermarket.org/eCommerce/AAIA_Imaging_Best_Practices.pdf

Shop Management Server

Introduction

A key component of the iShop standard is sharing of information between networked front shop and back shop equipment. To support this requirement, an iShop conformant shop management system must provide data storage and retrieval methods for other equipment. It must also be capable of accepting information delivered from back shop equipment on request of the back shop equipment.

The iShop shop management server interface is exposed as a stateless XML Web service. The Web service exposes a number of operations such as GetCustomer, GetOrder and CreatePhoto. Each of these operations has a request message and response message. These messages are described by way of XML Schema Definitions (XSDs). The Web service, operations and their use of messages are formally described using Web Services Definition Language (WSDL).

Process Overview

This section contains examples to illustrate some typical interactions between a back shop client application and a shop management server. The examples refer to shop management server operations summarized in the table below. Each operation is further described in detail in the section Operations.

Table 1 - Shop Management Web service operations

Operation	Summary
CreatePhoto	<p>Adds a new photo to either a customer or a vehicle.</p> <p>This operation may be called by a back shop system when a photo is created as part of the repair process.</p>
GetAdmin	<p>Returns information about the iShop server application such as vendor information, iShop version supported, licensing information, security requirements, etc.</p>
GetCustomer	<p>Returns customer information given a customer id value.</p> <p>This operation is typically called by a back shop system after an order has been selected from an order list so that the back shop equipment can display detailed information about the customer.</p>
GetOrderList	<p>Returns summaries of orders matching a search expression.</p> <p>This operation is typically called by a back shop system in order to present the technician with a list of pending orders.</p>
GetOrder	<p>Returns order information given an order id value.</p> <p>This operation is typically called by a back shop system after the technician has selected an order from a list returned by the <i>GetOrderList</i> operation.</p>
GetPhotoList	<p>Returns a list of all photos associated with either a vehicle or customer. The resulting photos may be displayed in the user interface of the back shop application.</p> <p><i>GetPhotoList</i> is more convenient to use than <i>GetPhoto</i> if you wish to retrieve all photos for a customer or vehicle.</p>

GetPhoto	Returns a photo given a photo id value. The resulting photo may be displayed in the user interface of the back shop application. <i>GetPhoto</i> is more convenient to use than <i>GetPhotoList</i> if you wish to retrieve one or a subset of photos for a customer or vehicle.
GetTechnicianList	Returns the names of technicians known by the SMS.
GetVehicle	Returns full information on a customer's vehicle given a vehicle id value. This operation may be called by a back shop system after the technician has selected an order from a list returned by the <i>GetOrderList</i> operation.
UpdateCustomer	Updates customer information in the SMS.
UpdateOrder	Updates order information in the SMS. Back shop equipment would use this operation to change the contents of an item – for example, adding notes or diagnostic information.
UpdateVehicle	Updates information in the SMS stored for a customer's vehicle.

Example – Executing a Repair Order

The technician is ready to do a repair on a customer vehicle. The following example dialog takes place between the iShop conformant back shop equipment application and the iShop conformant SMS application. The scenario is presented first as a sequence diagram, then in tabular format. The table also presents some alternate paths through the process.

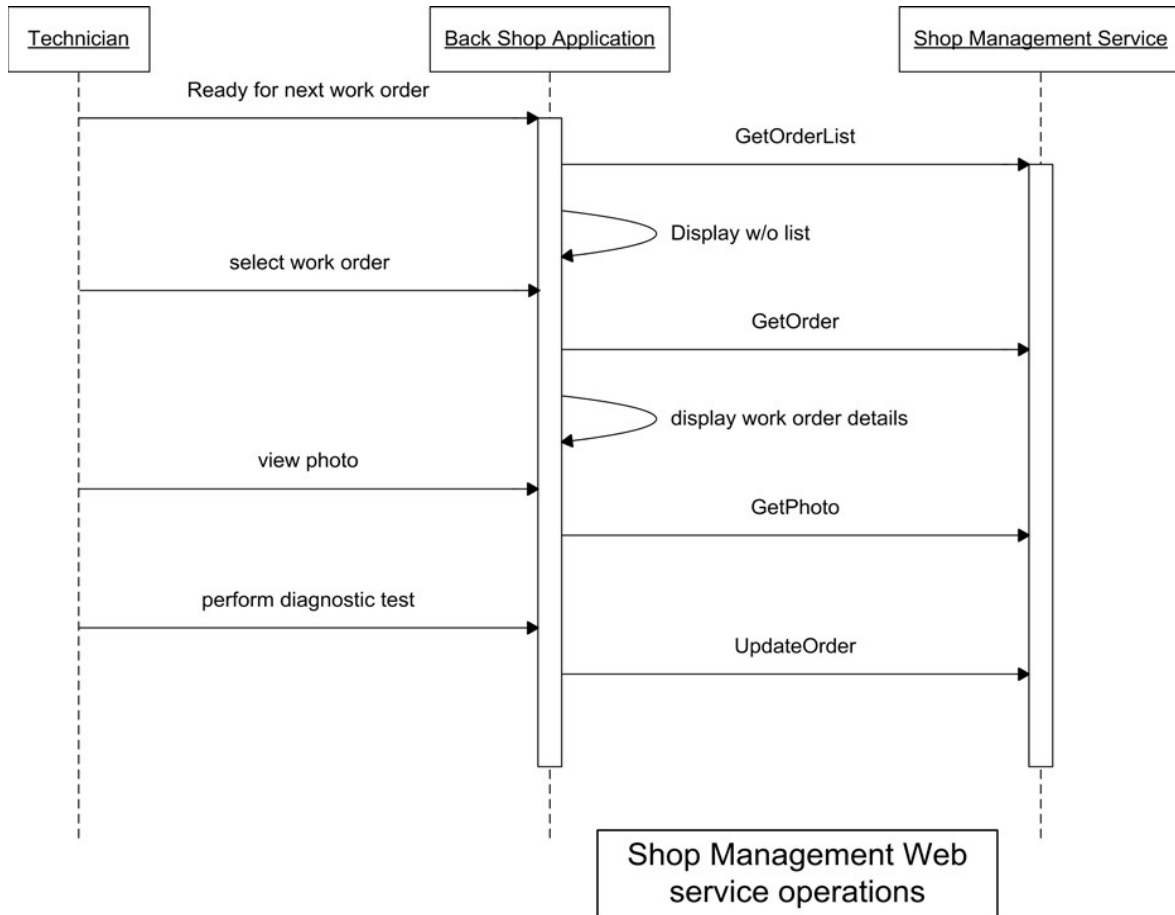


Figure 2 - Repair order sequence diagram

The sequence diagram shows three lifelines; technician, back shop application and the shop management service (SMS). The Web service operations are labeled on the calls depicted from the backshop application to the SMS.

The table below provides a text description of the scenario with some additional information about interaction with other applications for vehicle information lookup.

Table 2 - Repair Order Scenario

Client Action	Server Action
Invokes <i>GetOrderList</i> operation to get a list of open orders with an item assigned to the technician.	Returns a list of orders.

Displays the orders to the technician and allows the technician to make a selection.	
Invokes <i>GetOrder</i> operation for the selected order.	Returns full information for the selected order.
Displays the order to the technician.	
<p>The technician browses the order on the back shop equipment display to determine what work is required.</p> <p>The technician decides he wants to see a particular photo linked to the customer.</p>	
Invokes <i>GetPhoto</i> operation passing a photo ID taken from the previously returned order.	Returns photo jpeg and summary information.
Displays the photo to the technician.	
<p>The technician asks to see information on the customer's vehicle.</p> <p>If the <i>GetOrder</i> response contains a "Vehicle" element identifying the customer's vehicle using an "AcesVehicle" child element, then the information required may be embedded within that "AcesVehicle" element. If the information request cannot be satisfied this way, the back shop application may obtain vehicle information one of the following ways.</p> <ul style="list-style-type: none"> • Look up to a local copy of the ACES vehicle catalog. • Look up to a legacy data store using the Auto Care Association legacy ID 	

<ul style="list-style-type: none"> Engage in a conversation with an information server as documented in the Information Server section of this document. <p>Any of these look-ups can be done by the back shop application independently of the SMS, or alternatively by the SMS on the back shop equipment's behalf using non-iShop interfaces.</p>	
<p>At this point the technician performs the repair. Diagnostic information is created that should be can back to the SMS. Technician notes can be saved back to the SMS as well.</p>	

Security

iShop 3 does not specifying security conventions for SMS communication. The OASIS WS-Security standard for securing Web services was considered for SMS security, however at the time of iShop release, implementations of this technology were not sufficiently mature and platform/technology interoperability was not assured.

SMS security is likely to be needed in the future, and should be added to iShop in a subsequent release as: (1) WS-Security implementations and interoperability stabilize, and (2) the iShop community calls for this addition.

Further information on WS-Security is available at <http://www.oasis-open.org>.

Deployment

The following principles apply to client iShop client software finding and accessing an iShop server:

- No server software should have to be installed on the client computer in order to access the server.
- The client computer should not have to be configured with the server's location or URL.
- iShop servers should be able to be found as soon as they are made available without any intervention on the client computer.

iShop client programs find SMS servers on a local network using UDP (User Datagram Protocol) or over the internet using UDDI (Universal Description, Discovery, and Integration). SMS servers are within the shop in most cases, and therefore UDP applies in most cases.

Although UDP is an unreliable internet protocol, it is an efficient and suitable method of sending short messages. It is used simply to find one or more iShop servers, although more than one should not typically be found.

UDDI is a platform-independent, XML-based registry for businesses worldwide to list themselves and find each other on the Internet. UDDI listings can also define how the listed Web service should be used. UDDI is an OASIS standard. As mentioned above, UDDI is not typically used to find SMS Servers.

UDP/UDDI are used only to find an iShop server. Once found, Web service communication uses standard HTTP over TCP/IP.

The iShop 3 Technical Implementation Guide contains detailed technical information on this process.

Operations

This section provides the details on each of the Shop Management Server Web service. The name of the service is iShopSMSServer. The operations supported by the Web service are detailed in this section. Each operation has a request message and a response message. The naming convention for these messages is:

operationnameRequest

operationnameResponse

The namespace for all XML elements, unless otherwise noticed, is:

<http://www.aftermarket.org/iShop/V3/XMLSchema>

The backshop application is typically the “consumer” of the Web service and the SMS the “provider” of the Web service. It is the responsibility of the consumer to prepare and populate the request message and the responsibility of the service to provide the response message.

CreatePhoto

This operation is called to add a new photo to the SMS photo database, also identifying the object that the photo should be associated with (e.g. customer, vehicle). The response will return a photo ID if successful.

The *CreatePhoto* operation may be called by a back shop system when a photo is created as part of the repair process. Document Auto Care Association’s “Digital Assets Best Practices” may be useful with respect to creating and storing photos – see section *Related Document*.

CreatePhotoRequest

Photo: This element is required. It contains the photo that should be created.

Photoid: This element is required. Any value provided will be ignored. The SMS will assign a photo Id and return the value in the operation response.

This element is required solely because it and its siblings are based on a reusable type declaration. Standards require the Id element of such type declarations to be required.

DateCreated: This element is optional and its value must be a valid date and time. It contains the date and time the photo was created.

The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

Description: This element is optional. It contains a description of the photo.

Descriptions may include multi-line text using word wrap and/or CRLF end of line markers. No other control characters may be included.

JPEGBytes: This element is required and its value must be type “hexbinary”. It contains a hexadecimal encoding of a JPEG image representing the photo.

User Area: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

CustomerId: Exactly one of the VehicleId and CustomerId elements are required. This element identifies the customer to whom you want to associate the new photo. A customer's photos are listed in the *Customer* reused structure.

VehicleId: Exactly one of the VehicleId and CustomerId elements are required. This element identifies the physical vehicle to which you want to associate the new photo. A vehicle's photos are listed in the *Vehicle* reused structure.

User Area: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

In the following example, the JPEGBytes element value is abbreviated for clarity and does not represent an actual JPEG.

```
<?xml version="1.0" encoding="UTF-8"?>
<CreatePhotoRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Photo>
    <PhotoId/>
    <DateCreated>2005-11-23T16:43:00</DateCreated>
    <Description>This is a sample photo.</Description>
    <JPEGBytes>0FB7</JPEGBytes>
  </Photo>
  <VehicleId>6005</VehicleId>
</CreatePhotoRequest>
```

CreatePhotoResponse

PhotoId: This element is required. If the operation succeeded, it contains the photo id created by the SMS. The element should be empty if the operation failed.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<CreatePhotoResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <PhotoId>123</PhotoId>
  <Response>
    <Status>OK</Status>
  </Response>
```

</CreatePhotoResponse>

GetAdmin

This operation returns information about an iShop server application such as vendor, iShop version supported, licensing, etc. This information may be presented to the user of the client application.

GetAdmin is the only iShop operation that must not be secured. The response message, *GetAdminResponse*, optionally describes server security requirements applicable to other server operations (see element *SecurityRequirement*).

Element *SecurityRequirement* should be omitted or indicate “no security” when returned from an SMS server since security is not part of the iShop SMS specification. It is included in the specification, however, to support future security implementation, and because the same *GetAdminResponse* message is returned by an iShop Information Server that does support security.

GetAdminRequest

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetAdminRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
</GetAdminRequest>
```

GetAdminResponse

Admin: This element is required.

CompanyName: This element is optional. It contains the name of the company supplying the server software.

DataCopyright: This element is optional. It contains the copyright text from the server’s database, and must be displayed by the client.

DataExpiration: This element is optional and its value must be a valid date and time. It contains the expiration date of the server's databases.

The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

DataExpirationEnforced: This element is optional. It contains a flag to indicate whether the server enforces the expiration date. Values permitted are 'true', 'false', '1', or '0'.

DataVersion: This element is optional. It contains the server's database version number.

iShopVersion: This element is optional. It contains the iShop version number that the server has been certified to support.

LicenseCount: This element is optional and its value must be an integer. It contains the number of server licenses currently in use.

LicenseCountLimit: This element is optional and its value must be an integer. It contains the number of server licenses allowed.

SoftwareCopyright: This element is optional. It contains the copyright text from the server application, and must be displayed by the client.

SoftwareVersion: This element is optional. It contains the server software's version number.

SecurityRequirement: This element is optional. It describes the security requirements of all other server operations.

None: Exactly one of the None and UserPassword elements are required. This element indicates that iShop server's other Web service operations are not secured. This should be the cases for SMS servers since security is not part of the iShop SMS specification.

UserPassword: Exactly one of the None and UserPassword elements are required. This element indicates that a client must authenticate itself to the iShop server using a user ID and password. This option is only supported for iShop Information Servers, in which case the user ID and password are passed to the Open operation that starts the session.

HTTPS: This element is optional. It indicates that secure socket layer (SSL) transport security should be used when communicating with the iShop server.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetAdminResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Admin>
    <CompanyName>Acme Software Ltd.</CompanyName>
    <SecurityRequirement>
      <None />
    </SecurityRequirement>
  </Admin>
</Response>
  <Status>OK</Status>
</Response>
</GetAdminResponse>
```

GetCustomer

This operation is passed a customer id value and returns a customer structure. The standard response element will indicate whether the customer was found.

The *GetCustomer* operation is typically called by a back shop system after an order has been selected from an order list so that the back shop equipment can display detailed information about the customer.

GetCustomerRequest

CustomerId: This element is required. It contains a unique string identifying the customer that you want to retrieve. Typically this is the database primary key for the customer.

This value is usually obtained from the response messages of operations *GetOrderList* and *GetOrder*.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCustomerRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <CustomerId>004429</CustomerId> </GetCustomerRequest>
```

GetCustomerResponse

Customer: This element is required. It contains the standard Customer elements. This element is fully described in the Customer reused structure section.

Response: This element is required. It contains the standard response structure. See section Global XML Structures for full details.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCustomerResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Customer>
```

(See example of customer structure in Customer reused structure section on page 38.)

```
  </Customer>
  <Response>
    <Status>OK</Status>
  </Response>
</GetCustomerResponse>
```

GetOrderList

This operation is passed a search expression and returns summaries of the matching orders. The standard response element will indicate whether any orders were found.

The *GetOrderList* operation is typically called by a back shop system in order to present the technician with a list of pending orders. The back shop equipment should use the search criteria to list only the orders that are relevant to the equipment and appropriate for selection by the technician.

GetOrderListRequest

SearchExpression: This element is required. It contains a set of logically OR-ed together search expressions. An order will be returned if it matches any of these search expressions. Only one such search expression is provided in the majority of cases.

SearchExpressionOr: This element is optional and may be repeated any number of times. It contains a search expression comprising a set of logically AND-ed together expression components. An order will be returned if it matches all components of the search expression. All orders are returned if this element is missing.

SearchExpressionAnd: This element is required and may be repeated more than one time. It contains a field/value comparison component of a search expression.

Field: This element is required. It contains the name of the search field for this search expression component. See documentation section Appendix B – Order Search Fields for the list of supported search fields.

Operation: This element is required. It contains a logic operator for comparing the Field to the Value. Values permitted are 'LE', 'LT', 'EQ', 'GT', 'GE', and 'LIKE'. The first of these are the standard relational operators and are abbreviations for 'less than or equal', 'less than', 'equal', 'greater than', 'greater than or equal'. The 'LIKE' operation allows the client to search by a substring with wildcard character '%' which matches any number of characters and '_' which matches any single character.

Value: This element is required. It contains the comparison value for this search expression component.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<GetOrderListRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <SearchExpression>
    <SearchExpressionOr>
      <SearchExpressionAnd>
        <Field>FirstName</Field>
        <Operation>EQ</Operation>
        <Value>George</Value>
      </SearchExpressionAnd>
      <SearchExpressionAnd>
        <Field>LastName</Field>
        <Operation>EQ</Operation>
        <Value>Carpenter</Value>
      </SearchExpressionAnd>
    </SearchExpressionOr>
  </SearchExpression>
</GetOrderListRequest>

```

GetOrderListResponse

OrderList: This element is required.

Order: This element is optional and may be repeated any number of times. It contains a summary of an order satisfying the request search criteria.

CustomerId: This element is optional. It contains the unique customer ID string associated with the order.

VehicleId: This element is optional. It contains a unique ID identifying the physical vehicle associated with the order. The identified physical vehicle can be retrieved using the *GetVehicle* operation.

OrderID: This element is required. It contains the unique ID identifying the customer associated with the order. The identified order can be retrieved using the *GetOrder* operation. The *OrderID* value is unique across all orders in the SMS, including both current and historical estimates, work orders, and invoices. The *OrderID* may not be the number printed on the work order. Typically this is the database primary key for the order.

OrderNumber: This element is optional. It contains the printed order number as it appears on the paper document for this order.

Status: This element is optional. It contains the order status code. If included, exactly one of its child elements must be included.

Standard: This element contains a standard status value associated with the order. Permitted values are: 'WorkNotStarted', 'WorkInProgress', 'WorkComplete', 'WaitingForPart', 'WaitingForEquipment', 'WaitingForAuthorization', and 'WaitingForTechnician'.

Custom: This element contains a custom status value associated with the line item. Any value is permitted.

FullName: This element is optional. It contains the concatenated customer name associated with the order for display. The recommended implementation is as follows: If the Customer *FirstName* and *LastName* are both non-blank then the *FullName* is formed by concatenating the *FirstName* to the *LastName* with a comma and space separator between the first and last names. If the *FirstName* is nonblank but the *LastName* is blank the *FirstName* is also returned as the *FullName*. If the *LastName* is non-blank but the *FirstName* is blank the *LastName* is returned as the *FullName*. If the *FirstName* and *LastName* are both blank then the *Company* is returned as the *FullName*. Examples:

Table 3 – Full Name Illustrations

FirstName	LastName	Company	Fullname
Bill	Smith	Fresh Food Deliveries	Smith, Bill
Bill		Fresh Food Deliveries	Bill
	Smith	Fresh Food Deliveries	Smith
		Fresh Food Deliveries	Fresh Food Deliveries

Hat: This element is optional. It contains the identification tag or hat assigned to the vehicle by the shop to identify the vehicle associated with this order. Typically this information is displayed to the user to assist in identifying the vehicle.

Location: This element is optional. It contains the location of the vehicle associated with the order. Typically this information is displayed to the user to assist in locating the vehicle.

LicensePlate: This element is optional. It contains the license of the vehicle associated with the order for display.

YearMakeModel: This element is optional. It contains the concatenated year, make and model of the vehicle associated with the order for display. Rigorous classification as per ACES or any other standard is not important since the purpose of this information is to present pending orders to the back shop equipment operator for selection. Any values that are meaningful and that facilitate identification are sufficient.

OrderType: This element is optional. It contains the order type associated with the order. Permitted values are 'Estimate', 'Repair Order', 'Invoice', 'Quote', and 'History'.

Color: This element is optional. It contains the color of the vehicle associated with the order. Typically this information is displayed to the user to assist in identifying or locating the vehicle.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetOrderListResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
<OrderList>
    <Order>
```

```

    <CustomerId>004429</CustomerId>
    <VehicleId>6005</VehicleId>
    <OrderId>00022868</OrderId>
    <OrderNumber>024951</OrderNumber>
    <FullName>Carpenter, George</FullName>
    <Hat>042</Hat>
    <Location>side lot</Location>
    <LicensePlate>005TMZ</LicensePlate>
    <YearMakeModel>2005 Chevrolet Aveor</YearMakeModel>
    <OrderType>Repair Order</OrderType>
    <Color>Blue</Color>
  </Order>
</OrderList>
<Response>
  <Status>OK</Status>
</Response>
</GetOrderListResponse>

```

GetOrder

This operation is passed an order id value and returns an order structure. The standard response element will indicate whether the order was found.

The *GetOrder* operation is typically called by a back shop system after the technician has selected an order from the order list – see *GetOrderList* operation.

2.2.5.1 GetOrderRequest

OrderId: Exactly one of the *OrderId* and *OrderNumber* elements are required. This element contains the unique order ID string for the order you want to retrieve.

The *OrderID* value is unique across all orders in the SMS, including both current and historical estimates, work orders, and invoices. The *OrderID* may not be the number printed on the work order. Typically this is the database primary key for the order.

OrderNumber: Exactly one of the *OrderId* and *OrderNumber* elements are required. This element contains the printed order number for the order you want to retrieve.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetOrderRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
    <OrderId>00022868</OrderId>
</GetOrderRequest>
```

GetOrderResponse

Order: This element is required. It contains the standard Order elements. This element is fully described in the Order reused structure section (page 43).

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetOrderResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
    <Order>
```

(See example of order structure in Order reused structure section on page 40.)

```
    </Order>
    <Response>
        <Status>OK</Status>
    </Response>
```

</GetOrderResponse>

GetPhotoList

This operation returns a list of all photos associated with either a vehicle or customer. The resulting photos may be displayed in the user interface of the back shop application.

Operation *GetPhotoList* is more convenient to use than *GetPhoto* if you wish to retrieve all photos for a customer or vehicle. Operation *GetPhoto* is more convenient to use than *GetPhotoList* if you wish to retrieve a single photo given its photo Id, or a subset of photos referenced within a customer or vehicle element.

GetPhotoListRequest

VehicleId: Exactly one of the *VehicleId* and *CustomerId* elements are required. This element contains the unique vehicle ID string for the vehicle for which you want to retrieve all photos.

CustomerId: Exactly one of the *VehicleId* and *CustomerId* elements are required. This element contains the unique customer ID string for the customer for which you want to retrieve all photos.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetPhotoListRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <VehicleId>6005</VehicleId>
</GetPhotoListRequest>
```

GetPhotoListResponse

PhotoList: This element is optional. It contains any number of photos found using the request search criteria.

Photo: This element is optional and may be repeated any number of times. It contains a photo found using the request search criteria.

Photoid: This element is required. It contains a unique string identifying the photo. Typically this is the database primary key for the photo.

DateCreated: This element is optional and its value must be a valid date and time. It contains the date and time the photo was created. The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

Description: This element is optional. It contains a description of the photo. Descriptions may include multi-line text using word wrap and/or CRLF end of line markers. No other control characters may be included.

JPEGBytes: This element is required and its value must be type “hexbinary”. It contains a hexadecimal encoding of a JPEG image representing the photo.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

In the following example, the JPEGBytes element value is abbreviated for clarity and does not represent an actual JPEG.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetPhotoListResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
<PhotoList>
    <Photo>
        <Photoid>1234</Photoid>
```

```

        <DateCreated>2005-11-23T16:43:00</DateCreated>
        <Description>This is a sample photo.</Description>
        <JPEGBytes>0FB7</JPEGBytes>
    </Photo>
</PhotoList>
<Response>
    <Status>OK</Status>
</Response>
</GetPhotoListResponse>

```

GetPhoto

This operation is passed a photo id value and returns a photo structure. The standard response element will indicate whether the photo was found.

The *GetPhoto* operation may be called by the back shop equipment after obtaining a *PhotoId* from a customer or vehicle structure. The resulting photo may be displayed in the user interface of the back shop application.

Operation *GetPhoto* is more convenient to use than *GetPhotoList* if you wish to retrieve a single photo given its photo Id, or a subset of photos referenced within a customer or vehicle element. Operation *GetPhotoList* is more convenient to use than *GetPhoto* if you wish to retrieve all photos for a customer or vehicle.

GetPhotoRequest

PhotoId: This element is required. It contains a unique string identifying the photo that you want to retrieve. Typically this is the database primary key for the photo.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<GetPhotoRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <PhotoId>1234</PhotoId>
</GetPhotoRequest>
```

GetPhotoResponse

Photo: This element is optional. It contains the requested photo, if found.

PhotoId: This element is required. It contains a unique string identifying the photo. Typically this is the database primary key for the photo.

DateCreated: This element is optional and its value must be a valid date and time. It contains the date and time the photo was created. The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

Description: This element is optional. It contains a description of the photo. Descriptions may include multi-line text using word wrap and/or CRLF end of line markers. No other control characters may be included.

JPEGBytes: This element is required and its value must be type "hexbinary". It contains a hexadecimal encoding of a JPEG image representing the photo.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures for full details*.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures for full details*.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures for full details*.

Example

In the following example, the JPEGBytes element value is abbreviated for clarity and does not represent an actual JPEG.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetPhotoResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
```

```
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Photo>
    <PhotoId>1234</PhotoId>
    <DateCreated>2005-11-23T16:43:00</DateCreated>
    <Description>This is a sample photo.</Description>
    <JPEGBytes>0FB7</JPEGBytes>
  </Photo>
  <Response>
    <Status>OK</Status>
  </Response>
</GetPhotoResponse>
```

GetTechnicianList

This operation returns the names of technicians known by the SMS.

GetTechnicianListRequest

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetTechnicianListRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
</GetTechnicianListRequest>
```

GetTechnicianListResponse

TechnicianList: This element is required.

Technician: This element is optional and may be repeated any number of times.

Code: This element is optional. It contains the code for a technician.

Name: This element is optional. It contains the name of a technician. There is no required named format.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetTechnicianListResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <TechnicianList>
    <Technician>
      <Code>bbrown</Code>
      <Name>Bob Brown</Name>
    </Technician>
    <Technician>
      <Code>jjones</Code>
      <Name>John Jones</Name>
    </Technician>
  </TechnicianList>
  <Response>
    <Status>OK</Status>
  </Response>
</GetTechnicianListResponse>
```

GetVehicle

This operation is passed a vehicle id value and returns a vehicle structure representing a physical vehicle. A physical vehicle is a vehicle that belongs to a customer, and can be the subject of a repair order. The standard response element will indicate whether the vehicle was found.

This operation may be called by a back shop system after the technician has selected an order from a list returned by the *GetOrderList* operation. Note that the vehicle structure returned by this operation is also embedded in the response to the *GetOrder* operation.

GetVehicleRequest

VehicleId: This element is required. It contains a unique string identifying the physical vehicle that you want to retrieve. Typically this is the database primary key for the vehicle.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetVehicleRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
    <VehicleId>6005</VehicleId>
</GetVehicleRequest>
```

GetVehicleResponse

Vehicle: This element is required. It contains the standard Vehicle elements. This element is fully described in the Vehicle reused structure section (page 104).

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetVehicleResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
<Vehicle>
```

(See example of vehicle structure in Vehicle reused structure section on page 98.)

```
</Vehicle>
<Response>
  <Status>OK</Status>
</Response>
</GetVehicleResponse>
```

UpdateCustomer

This operation updates customer information in the SMS.

UpdateCustomerRequest

Customer: This element is required. It contains the standard Customer elements as retrieved by operation *GetCustomer*. This element is fully described in the Customer reused structure section.

The customer to update is identified by the *CustomerId* element. Changes to update back to the SMS should be reflected in the structure's elements at the time of this operation. The set of updatable elements is defined by the SMS implementation.

Elements that should not be updated should be left out of this structure. An empty element should be included in this structure if the saved value of the element should be erased. Note that the *CustomerId* value must always be specified solely for the purpose of identifying the record to update.

IMPORTANT: Special considerations need to be taken into account when updating the Note element – see Appendix C – Note Fields.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<UpdateCustomerRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Customer>
    <CustomerId>004429</CustomerId>
    <Address>29384 Langaunet Ln NE</Address>
    <Note>Fixing George's street address.</Note>
  </Customer>
</UpdateCustomerRequest>
```

UpdateCustomerResponse

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<UpdateCustomerResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Response>
    <Status>OK</Status>
  </Response>
</UpdateCustomerResponse>
```


UpdateOrder

This operation updates order information in the SMS.

Back shop equipment would rarely change order information, but would use this operation to change the contents of an item - for example, adding notes or diagnostic information. More detail is provided in the request message documentation below.

UpdateOrderRequest

Order: This element is required. It contains the standard Order elements as retrieved by operation *GetOrder*. This element is fully described in the Order reused structure section.

The order to update is identified by the *OrderId* element. Changes to update back to the SMS should be reflected in the structure's elements at the time of this operation. The set of updatable elements is defined by the SMS implementation.

Elements that should not be updated should be left out of this structure. This convention makes it easy for back shop equipment to update an item's note value – which is a typical course of action. All the back shop equipment needs to include in the Order structure is the order ID, item element with item ID, and a note element within the item.

An empty element should be included in this structure if the saved value of the element should be erased. Note that the *OrderId* value must always be specified solely for the purpose of identifying the record to update.

IMPORTANT: Special considerations need to be taken into account when updating the Note element – see Appendix C – Note Fields.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<UpdateOrderRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Order>
```

```

    <OrderId> 00022868</OrderId>
    <Note>Front brake discs need replacement.</Note>
    <OdometerOut Unit="Miles">5971</OdometerOut>
    <ItemList>
      <Item>
        <ItemID> 00022868030001</ItemID>
        <Note>Alignment completed successfully</Note>
      </Item>
    </ItemList>
  </Order>
</UpdateOrderRequest>

```

UpdateOrderResponse

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<UpdateOrderResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
  xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
  xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Response>
    <Status>OK</Status>
  </Response>
</UpdateOrderResponse>

```

UpdateVehicle

This operation updates information on a physical vehicle in the SMS. A physical vehicle is a vehicle that belongs to a customer, and can be the subject of a repair order.

UpdateVehicleRequest

Vehicle: This element is required. It contains the standard Vehicle elements as retrieved by operation *GetVehicle*. This element is fully described in the Vehicle reused structure section (page 104).

The physical vehicle to update is identified by the “Vehicle / PhysicalVehicle / VehicleId” element. Changes to update back to the SMS should be reflected in the structure’s elements at the time of this operation. The set of updatable elements is defined by the SMS implementation.

Elements that should not be updated should be left out of this structure. An empty element should be included in this structure if the saved value of the element should be erased. Note that the VehicleId value must always be specified solely for the purpose of identifying the record to update.

An SMS would save to its database information from the “Vehicle / PhysicalVehicle” element since this information represents the customer’s vehicle. The SMS would typically only save identifying information from the other “vehicle type” elements – i.e. elements Auto Care Association LegacyId or AcesVehicle. For example, if the vehicle is identified using the AcesVehicle structure, then the SMS would typically only save values for elements BaseVehicle and EngineBase. Information from other elements could be restored through subsequent lookups to the source database – e.g. the ACES VCDB (vehicle configuration database).

IMPORTANT: Special considerations need to be taken into account when updating the Note element – see Appendix C – Note Fields.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<UpdateVehicleRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
<Vehicle>
    <AcesVehicle>
        <vehicle:BaseVehicle id="18296"></vehicle:BaseVehicle>
        <vehicle:EngineBase>1460</vehicle:EngineBase>
    </AcesVehicle>
</Vehicle>
```

```

    <PhysicalVehicle>
      <VehicleId>6005</VehicleId>
      <LicensePlate>006TMZ</LicensePlate>
      <Note>Updating license plate information.</Note>
    </PhysicalVehicle>
  </Vehicle>
</UpdateVehicleRequest>

```

UpdateVehicleResponse

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<UpdateVehicleResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Response>
    <Status>OK</Status>
  </Response>
</UpdateVehicleResponse>

```

XML Structures Reused in SMS

The XML structures described in this section are used in more than one SMS request or response message. Since they are shared components, they are described here once for consistency. The namespace for all XML elements, unless otherwise identified is:

<http://www.aftermarket.org/iShop/V3/XMLSchema>

Customer

This structure identifies a customer who may own a vehicle that is the subject of a repair order. Child elements are as follows.

CustomerId: This element is required. It contains a unique string identifying the customer. Typically this is the database primary key for the customer.

FirstName: This element is optional. It contains the customer's first name.

LastName: This element is optional. It contains the customer's last name.

Company: This element is optional. It contains the customer's company name. This element is applicable to vehicles that are owned by a business rather than an individual.

Address: This element is optional. It contains the street address for the customer.

City: This element is optional. It contains the city for the customer.

State: This element is optional. It contains the state or province for the customer. Standard codes appropriate to the country are used where applicable.

Zip: This element is optional. It contains the zip or postal code for the customer.

Country: This element is optional. It contains the country for the customer. The country is specified using the ISO-3166-1 two-letter country code.

See <http://www.din.de/gremien/nas/nabd/iso3166ma/> for information on ISO-3166-1 and <http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html> for an English list.

Note: This element is optional. It contains the note associated with the customer.

See important information on note elements in section Appendix C – Note Fields. **ContactList:** This element is optional. It contains any number of contact descriptors for the customer – e.g. phone number, email address.

Phone: This element is optional and may be repeated any number of times. It contains a phone number for the customer.

Type: This element is required. It contains the type of phone number represented – E.g. 'home', 'work', etc. The supported values are defined by the implementer.

Comment: This element is optional. It contains general notes about the phone number.

Number: This element is optional. It contains the phone number except for the extension. The phone number can be formatted any way that is acceptable to the server system.

Ext: This element is optional. It contains the phone number extension.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Email: This element is optional and may be repeated any number of times. It contains an email address for the customer.

PhotoList: This element is optional. It contains any number of photo references for the customer. Photo references are included rather than actual photos to save message space. Photos are retrieved using the GetPhoto and GetPhotoList operations.

Photo: This element is optional and may be repeated any number of times. It contains a photo reference for the customer.

Photoid: This element is required. It contains a unique string identifying the photo. Typically this is the database primary key for the photo.

DateCreated: This element is optional and its value must be a valid date and time. It contains the date and time the photo was created. The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

Description: This element is optional. It contains a description of the photo. Descriptions may include multi-line text using word wrap and/or CRLF end of line markers. No other control characters may be included.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

2.3.1.1 Example

```
<Customer>  
  <CustomerId>004429</CustomerId>  
  <FirstName>George</FirstName>
```

```

<LastName>Carpenter</LastName>
<Address>29864 Launganet Ln NE</Address>
<City>Poulsbo</City>
<State>WA</State>
<Zip>98370</Zip>
<Country>USA</Country>
<Note>George is a long-standing customer</Note>
<ContactList>
  <Phone>
    <Type>Work</Type>
    <Comment>Call before 4:30pm</Comment>
    <Number>360-456-7890</Number>
    <Ext>12</Ext>
  </Phone>
  <Phone>
    <Type>Home</Type>
    <Number>360-890-4536</Number>
  </Phone>
  <Email>george_carpenter01@hotmail.com</Email>
</ContactList>
<PhotoList>
  <Photo>
    <Photold>3456</Photold>
    <DateCreated>2005-11-24T15:20:00</DateCreated>
    <Description>George's Car</Description>
  </Photo>
</PhotoList>
</Customer>

```

Order

This structure describes a repair order. Child elements are as follows.

OrderId: This element is required. It contains the unique order ID string associated with the order. The OrderID value is unique across all orders in the SMS, including both current and historical estimates, work

orders, and invoices. The OrderID may not be the number printed on the work order. Typically this is the database primary key for the order.

OrderNumber: This element is optional. It contains the printed order number as it appears on the paper document for this order.

Status: This element is optional. It contains the order status code. If included, exactly one of its child elements must be included. This element is read-only, and thus may not be updated by the UpdateOrder operation (back shop equipment should only be concerned with its order item).

Standard: This element contains a standard status value associated with the order. Permitted values are: 'WorkNotStarted', 'WorkInProgress', 'WorkComplete', 'WaitingForPart', 'WaitingForEquipment', 'WaitingForAuthorization', and 'WaitingForTechnician'.

Custom: This element contains a custom status value associated with the line item. Any value is permitted.

Customer: This element is optional. It contains the standard Customer elements. This element is fully described in the Customer reused structure section (page 40).

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the Vehicle reused structure section (page 104).

Hat: This element is optional. It contains the identification tag or hat assigned to the vehicle by the shop to identify the vehicle associated with this order. Typically this information is displayed to the user to assist in identifying the vehicle.

Location: This element is optional. It contains the location of the vehicle associated with the order. Typically this information is displayed to the user to assist in locating the vehicle.

Manager: This element is optional. It contains the manager associated with the order. Child element values are independent of the Technician List returned by operation *GetTechnicianList*.

Code: This element is optional. It contains the code for the manager.

Name: This element is optional. It contains the name of the manager. There is no required named format.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Note: This element is optional. It contains the note associated with the order. See important information on note elements in section Appendix C – Note Fields.

OdometerIn: This element is optional and its value must be numeric. It contains the odometer reading IN of the vehicle associated with the order. This element supports an optional attribute 'Unit'. Permitted values are 'Miles' and 'KM'.

OdometerOut: This element is optional and its value must be numeric. It contains the odometer reading OUT of the vehicle associated with the order. This value is empty for Open Orders. This element supports an optional attribute 'Unit'. Permitted values are 'Miles' and 'KM'.

OrderType: This element is optional. It contains the order type associated with the order. Permitted values are 'Estimate', 'Repair Order', 'Invoice', 'Quote', and 'History'.

Promised: This element is optional and its value must be a valid date and time. It contains the date and time associated with the order the customer was promised the vehicle would be ready for pickup.

The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

Scheduled: This element is optional and its value must be a valid date and time. It contains the scheduled date and time associated with the order that work is/was scheduled to begin.

The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

ServiceWriter: This element is optional. It contains the service writer associated with the order. Child element values are independent of the Technician List returned by operation *GetTechnicianList*.

Code: This element is optional. It contains the code for the service writer.

Name: This element is optional. It contains the name of the service writer. There is no required named format.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Timeln: This element is optional and its value must be a valid date and time. It contains the date and time IN associated with the order. Typically this is the date and time that the vehicle was delivered to the shop and the initial work was authorized.

The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

TimeOut: This element is optional and its value must be a valid date and time. It contains the date and time OUT associated with the order. Typically this is the date and time that the vehicle was returned to the customer.

The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

ItemList: This element is optional. It contains any number of items for the order.

Item: This element is optional and may be repeated any number of times. It contains a item for the order.

ItemId: This element is optional. It contains the unique line item ID used by the shop management system. The ItemId value is unique across all orders, including both current and historical estimates, work orders, and invoices.

Description: This element is optional. It contains the full line item description.

This description describes the part or labor associated with this item. Part line items may contain textual part descriptions; labor line items may contain textual labor descriptions. Note that a line item is associated with either a Part or Labor, but not neither nor both.

Descriptions contain all text for the item. Descriptions may include multi-line text using word wrap and/or CRLF end of line markers. No other control characters may be included.

Status: This element is optional. It contains the line item status code. If included, exactly one of its child elements must be included.

Standard: This element contains a standard status value associated with the line item. Permitted values are: 'WorkNotStarted', 'WorkInProgress', 'WorkComplete', 'WaitingForPart', 'WaitingForEquipment', 'WaitingForAuthorization', and 'WaitingForTechnician'.

Custom: This element contains a custom status value associated with the line item. Any value is permitted.

Note: This element is optional. It contains the note associated with the order item.

See important information on note elements in section Appendix C – Note Fields.

Group: This element is optional. It contains the group associated with the item. Certain line items may be associated or grouped with other line items on the current order such as ‘packaged’ parts, multiple line descriptions and fees. This line item has no grouping if the element is missing or is empty.

Symptom: This element is optional. It contains a symptom associated with the line item. If included, exactly one of its child elements must be included.

Standard: This element contains a standard symptom associated with the line item. Permitted values are not yet defined.

Custom: This element contains a custom symptom associated with the line item. Any value is permitted.

EquipmentType: This element is optional. It contains an equipment type associated with the line item. If included, exactly one of its child elements must be included.

Standard: This element contains a standard equipment type associated with the line item. Permitted values are:

- Unknown
- None
- Automotive Wheel Aligner
- Automotive Brake Tester
- Automotive Suspension Tester
- Automotive Wheel Balancer
- Automotive Gas Analyzer
- Automotive Diagnostic Unit (Hand-Held)
- Automotive Diagnostic Unit (Workstation)
- Automotive Ignition Analyzer • Automotive Engine Analyzer
- Automotive Structural Repair

- Automotive Structural Estimation
- Automotive Structural Verification
- Automotive A/C Service Equipment
- Automotive Fluid Flush / Transfer Equipment
- Automotive Brake Lathe
- Automotive Scan Tool
- Automotive Lab Scope
- Automotive Government Mandated Emissions Analyzer
- Automotive Battery/Electrical Systems Tester

Custom: This element contains a custom equipment type associated with the line item. Any value is permitted.

Part: This element is optional. It contains part information associated with the line item. If included, the Labor element cannot be included. Therefore a line item represents a part, labor, neither, but not both.

PartNumber: This element is optional. It contains the part number for this part item. The part number used should be meaningful to the audience for the order – i.e. there is no requirement to indicate whether this is a manufacturer part number, vendor part number, etc.

Description: This element is optional. It contains description of the part.

Descriptions may include multi-line text using word wrap and/or CRLF end of line markers. No other control characters may be included.

Quantity: This element is optional and its value must be decimal. It contains the quantity of this part normally required. Value -1 indicates a default quantity is not available for this item.

This element supports an optional attribute 'UOM'. Any string value is permitted.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Labor: This element is optional. It contains labor information associated with the line item. If included, the Part element cannot be included. Therefore a line item represents a part, labor, neither, but not both.

LaborCode: This element is optional. It contains the labor code for this labor item.

Technician: This element is optional and may be repeated any number of times. It identifies a technician for the labor item.

Code: This element is optional. It contains the code for a technician.

Name: This element is optional. It contains the name of a technician. There is no required named format.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

Quantity: This element is optional and its value must be decimal. It contains the amount of labor expressed in hours.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

DiagnosticResultsList: This element is optional. It contains any number of diagnostic results for the order line.

DiagnosticResult: This element is optional and may be repeated any number of times. It contains any number of diagnostic results for the order line encoded using varying standards.

ASANETISHOP: This element is required as it is the only diagnostic result encoding standard presently supported. It may be repeated any number of times. It contains any number of diagnostic results for the order line encoded using the ASANET/iShop standard.

The ASANET/iShop standard is defined in document, "Definition for inspection results in asanetwork", edition 3.0 dated "14.11.2002". This document is presently available at: http://www.axonet.de/public_doc/xml_v30_en.pdf.

Results: This element is optional. It contains an XML document representing the diagnostic results formatted according to the ASANET/iShop document referenced above.

The namespace for all child XML elements is:

`http://www.axonet.de/4.0/awnres.xsd`

StyleSheet: This element is optional. It contains and describes an XSLT style sheet to properly render the diagnostic results in HTML. Child elements name and version should be leveraged by the SMS to save only one copy of a style sheet, rather than repeatedly saving the same style sheet each time it is used.

Name: This element is optional. It contains the name of the style sheet to transform the diagnostic results XML into viewable HTML.

Version: This element is optional. It contains the version of the named style sheet to transform the diagnostic results XML into viewable HTML.

ZipFile: This element is optional and its value must be type “hexbinary”. It contains a hexadecimal encoding of a zip file of all files making up the named style sheet of the specified version.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

Order Example

In the following example, the ZipFile element value is abbreviated for clarity and does not represent an actual Zip file.

```
<Order>  
  <OrderId>00022868</OrderId>  
  <OrderNumber>024951</OrderNumber>
```

```

<Customer>
  <CustomerId>004429</CustomerId>
</Customer>
<Vehicle>
  <AaiaLegacyId>1432644</AaiaLegacyId>
</Vehicle>
<Hat></Hat>
<Location></Location>
<Manager></Manager>
<Note></Note>
<OdometerIn Unit="Miles">5962</OdometerIn>
<OdometerOut Unit="Miles">5971</OdometerOut>
<OrderType>Repair Order</OrderType>
<Promised>2005-11-24T18:25:00</Promised>
<Scheduled>2005-11-24T16:00:00</Scheduled>
<ServiceWriter></ServiceWriter>
<TimeIn>2005-11-24T16:00:00</TimeIn>
<TimeOut>2005-11-24T15:30:00</TimeOut>
<ItemList>
  <Item>
    <ItemID></ItemID>
    <Description></Description>
    <Status>
      <Standard>WaitingForPart</Standard>
    </Status>
    <Note></Note>
    <Group></Group>
    <Symptom>
      <Standard></Standard>
    </Symptom>
    <EquipmentType>
      <Standard>Automotive Brake Tester</Standard>
    </EquipmentType>
    <Part>
      <PartNumber></PartNumber>
  </Item>
</ItemList>

```

```

        <Description></Description>
        <Quantity>2.0</Quantity>
    </Part>
</Item>
<Item>
    <ItemID></ItemID>
    <Description></Description>
    <Status>
        <Standard>WaitingForPart</Standard>
    </Status>
    <Note></Note>
    <Group></Group>
    <Symptom>
        <Standard></Standard>
    </Symptom>
    <EquipmentType>
        <Standard>Automotive Brake Tester</Standard>
    </EquipmentType>
    <Labor>
        <LaborCode></LaborCode>
        <Technician>
            <Code>jjones</Code>
            <Name>John Jones</Name>
        </Technician>
        <Quantity>1.0</Quantity>
    </Labor>
    <DiagnosticResultsList>
        <DiagnosticResult>
            <ASANETISHOP>
                <asanet:RESULTS></asanet:RESULTS>
                <StyleSheet>
                    <Name></Name>
                    <Version></Version>
                </StyleSheet>
            </ASANETISHOP>
        </DiagnosticResult>
    </DiagnosticResultsList>
    <ZipFile>0123456789abcdef</ZipFile>
    </StyleSheet>

```



```

    </ASANETISHOP>
      </DiagnosticResult>
    </DiagnosticResultsList>
  </Item>
</ItemList>
</Order>

```

Information Server

Introduction

The primary purpose of an iShop Information Server (IS) is to provide information to SMS and backshop clients. The IS provides diagnostic and repair procedures as well as parts and labor information.

“Navigation” is the process used by an information server client to find the information they wish to retrieve. In particular, navigation is the process by which a client application drills down from general to specific navigation levels through the information server’s information hierarchy towards the information the user is ultimately seeking.

Throughout this documentation, the term “information server” should be considered synonymous with “information server application” or “information server interface”.

The information server is a statefull server in that a client application opens a session, invokes operations to search for information, and then closes the session. The information server saves state information for the duration of the session – for example, the vehicle of interest and the current navigation position.

An iShop information server presents both a programmatic interface and an HTML interface.

The programmatic interface is an XML Web service called by client applications that interface to a user. The client application lets the user find information by iteratively accepting a user’s navigational choice, calling the information server, presenting the results, and letting the user navigate further by choosing again from these results.

The HTML interface lets the user perform the same type of navigation that could be performed within a client application that uses the programmatic interface. A navigation results returned by the programmatic interface (Web service) includes a link to a web page that presents the same results in the HTML interface. Therefore, a client application using the programmatic interface can let the user jump

over to the HTML interface. The HTML interface web page may permit navigation to other pages presenting other data, however this navigation is outside the scope of the iShop specification.

Process Overview

This section contains examples to illustrate some typical interactions between a client application and an information server. The examples refer to information server operations summarized in the table below. Each operation is further described in detail in section Operations.

Table 4 - Information Server Web service operations

Operation	Summary
<i>Session and System Operations</i>	
Open	Opens an information server session.
Close	Closes an information server session.
GetAdmin	Returns information about the iShop server application such as vendor information, iShop version supported, licensing information, security requirements, etc.
<i>Navigation Configuration Operations</i>	
GetInformationTypes	Returns a list of information types that the information server can serve or that the information server has been asked to serve. Examples: repair information, tire fitment information, buyers guides, etc.
SetInformationTypes	Limits the information types the server will return back to the client.
SetNavigationHierarchy	Selects the navigation hierarchy that will be used for navigation.
<i>Navigation Operations</i>	
Navigate	Navigates to a specific navigation item in the information server hierarchy and returns a list of child navigation items and/or child terminal information items.

Search	Searches the information server database and returns a list of navigation items and/or terminal information items that match user specified criteria.
GetNavigationPath	Returns a list of navigation items from the navigation root to the current navigation item in the information server hierarchy.
PopNavigation	Backs up a specified number of levels from the current navigation item in the information server hierarchy toward the navigation root. This operation returns the same list of child navigation items and/or child terminal information items that was returned when this navigation item was originally navigated to.
<i>Line Filtering Operations – see Line Filtering Section.</i>	
CatalogInformation	Retrieves information about the catalogs available from the information server, including the lines they contain.
ComponentInformation	Retrieves information about the components (a.k.a. part types) available from the information server. In particular, this operation returns information about components within a catalog and line.
LFConfigurationAdd	Adds a new line configuration to the information server's storage for the current user.
LFConfigurationDelete	Deletes a line configuration from the information server's storage for the current user.
LFConfigurationList	Retrieves a list of line configuration saved to the information server's storage for the current user.
LFConfigurationRetrieve	Retrieves the full details of a named line configuration from the information server's storage for the current user.
LFConfigurationUpdate	Updates a line configuration saved on the information server's storage for the current user.

SetLineFiltering	Enables or disables line filtering. Line filtering is enabled by passing the name of a line filtering configuration stored on the information server for the current user.
SetLineFilteringConfiguration	Enables or disables line filtering. Line filtering is enabled by passing an actual configuration element instead of the name of a stored configuration.

Example – Basic Navigation

The user wants to view brake system Technical Service Bulletins for a 2003 Infiniti G35. The client application does not identify the vehicle to the server.

Table 5 - Basic Navigation Example

Client Action	Server Action
Invokes <i>Open</i> operation without passing a user ID and password in the request message. The client is testing whether the information server is not secured. See section <i>Security</i> .	Returns 'Access Denied'. The information server is in fact secured.
Gets username and password from the user, and invokes the <i>Open</i> operation with that information.	Returns 'OK'. The information server session is now open.
Invokes <i>Navigate</i> operation passing neither a navigation item nor a vehicle.	Returns a <i>NavigationList</i> element containing a list of <i>NavigationItem</i> elements. Each item represents a different year.
Displays the navigation items to the user, and allows the user to make a choice.	
Invokes <i>Navigate</i> passing the navigation item representing the user's choice (2003 in this case).	Returns another <i>NavigationList</i> element containing a set of <i>NavigationItem</i> elements representing the valid makes (A.K.A. manufacturers).

Displays the choices to the user. User chooses 'Inifiniti' in this case. Invokes <i>Navigate</i> again	Returns another <i>NavigationList</i> , this time containing navigation items representing the possible models.
Displays the choices to the user. User chooses 'G35 Coupe'. Invokes <i>Navigate</i> again.	Returns the possible engine choices to the user in another <i>NavigationList</i> .
User chooses the V6 engine. Invokes <i>Navigate</i> again.	The server now knows the vehicle to use. It returns the top level navigation items for the selected vehicle in a <i>NavigationList</i> element.
	The information server may now return a <i>Vehicle</i> element representing the selected vehicle.
The user continues to navigate down the hierarchy towards 'brake systems' for the identified vehicle until reaching a location where he can go no further – i.e. the <i>Navigate</i> operation returns no more navigation items. At this point the server returns the terminal information elements found at this point in the hierarchy. These may be articles, technical service bulletins, part information, etc.	
User selects the Technical Service Bulletin item they are interested in and the client application deals with it appropriately.	

To further this example, the user now chooses to back up the navigation to a previous position, so they can go down a different path. This scenario builds upon the prior one.

Table 6 -Pop Navigation Sample

Client Action	Server Action
User presses the 'Back' button displayed by the client application.	

Invokes the <i>PopNavigation</i> operation, passing 1 as a parameter (i.e. pop up 1 level).	The server returns a <i>NavigationList</i> element containing a set of <i>NavigationItem</i> elements previously seen when navigating down the hierarchy.
Displays the choices to the user. The user chooses a different navigation item than their prior selection. The client then invokes the <i>Navigate</i> operation with the new selected <i>NavigationItem</i> .	The server returns information for the user's new choice.
User continues navigating and selecting terminal information.	
Invokes the <i>Close</i> operation to close the information server session.	Returns 'OK'. The information server session is now closed.

Example – Creating a Line Filtering Configuration

The user wants to create a line filtering configuration, to limit which catalogs are available for selected part types.

Table 7 - Line Filtering Example

Client Action	Server Action
Invokes the <i>Open</i> operation passing a valid user id and password. See section <i>Security</i> .	Returns 'OK'. The information server session is now open.
Invokes the <i>CatalogInformation</i> operation to get a list of catalogs and line codes.	Returns a list of the catalogs supported.
Displays the catalog and line information to the user. The user selects the catalogs and lines that they wish to see parts for.	

Invokes the <i>ComponentInformation</i> operation for each selected catalog and line to get a list of components.	Returns a list of supported components for each catalog and line.
Displays the components supported for each catalog and line selected by the user. The user selects which components they wish to see parts for by catalog and line.	
Builds the line filtering configuration structure from the catalogs / lines / components selected by the user.	
The client application asks the user for a name for the line filtering configuration. It then invokes the <i>LFCconfigurationAdd</i> operation to store the line filtering configuration on the server.	The server stores the line filtering configuration for the user.
Invokes the <i>Close</i> operation to close the information server session.	Returns 'OK'. The information server session is now closed.

Example – Viewing Parts filtered by a Line Filtering Configuration

The user wants to view part information (and only part information) using the line filtering configuration it just created. The client identifies the vehicle using an Auto Care Association legacy vehicle.

Table 8 - Viewing Filtered Parts Example

Client Action	Server Action
Invokes the Open operation passing a valid user id and password. See section <i>Security</i> .	Returns 'OK'. The information server session is now open.

<p>Invokes the <i>GetInformationTypes</i> operation to request a list of all the information types the server supports (i.e. attribute Type="All").</p>	<p>Returns a list of all of the information types it supports.</p>
<p>Presents the set of information types to the user. The user indicates they only want to see type 'AftermarketParts'.</p>	
<p>Invokes the <i>SetLineFiltering</i> operation, passing the name of a previously stored line filtering configuration (see previous example).</p>	<p>Remembers to use the line filtering configuration for the current session.</p>
<p>Invokes the <i>SetInformationTypes</i> operation, passing only 'AftermarketParts'.</p>	<p>Sets the active information type to only 'AftermarketParts' for the current session.</p>
<p>Invokes <i>Navigate</i> operation passing a vehicle element containing an Auto Care Association legacy id value. Navigation item is not passed in.</p>	<p>Validates the vehicle id resulting in one of the following possible actions:</p> <ul style="list-style-type: none"> • If the Auto Care Association legacy id is a direct match to the server's vehicle id system, the server will return the top level navigation information for the vehicle. It may also return a <i>Vehicle</i> element representing the matched vehicle. • If the Auto Care Association legacy id matches more than one server vehicle id, the server will return a <i>NavigationList</i> with navigation items for each possible choice. • If the Auto Care Association legacy id matches nothing in the server's vehicle database, the server will return a <i>NavigationList</i> with top level navigation items containing the vehicle years.

<p>Displays the navigation items to the user, and allows the user to make a choice.</p> <p>Invokes the <i>Navigate</i> operation again passing the selected navigation item.</p>	<p>Navigates to the selected navigation item and returns navigation items for the next level of information</p>
<p>The user continues to navigate down the hierarchy until <i>PartInformation</i> elements are found. The <i>PartInformation</i> elements received will be filtered by the line filtering configuration selected by the user. The client application displays the part information to the user.</p>	
<p>Invokes the <i>Close</i> operation to close the information server session.</p>	<p>Returns 'OK'. The information server session is now closed.</p>

Line Filtering

Line filtering is a mechanism whereby the client can restrict the parts that are returned by an Information Server by catalogs and lines. This helps keep the number of parts returned down to a manageable level, and also helps the client focus only on the parts they are interested in.

One or more line filters set up by a client can be saved for easy re-use in a Line Filtering Configuration. These configurations are stored by the Information Server on the client's behalf. For example, one configuration may be used for hot-shot orders, and a different configuration for replenishment (stocking) orders.

Requirements

The following requirements have been addressed by Line Filtering in iShop 3:

- Line filtering must work off an inclusion model, not an exclusion model. When catalogs which carry thousands of lines try to implement iShop 3, the client must be able to specify which lines to include, rather than which lines to exclude (block).
- Line filtering must support a hierarchy of catalogs and lines. A given catalog might have many different line codes. The client must be able to view just the catalogs themselves, and then expand given catalogs to see the actual lines. The client should be able to include (or exclude) lines at the catalog level, and not need to descend to the line level to perform this operation.

- The client needs to be able to create multiple line filtering configurations (one per distributor, for example).
- The Information Server should be able to persist the client’s line filtering configurations.
- The line filtering process must be efficient. I.e. it must not require a Web services request to validate each individual part.
- The client should have an option to access parts that were filtered out without any additional Web service calls.

Definitions

The following definitions apply to the subject of line filtering:

Table 9 - Line Filtering Subjects

Term	Definition
Catalog:	Line Filtering catalogs are equivalent to the paper catalogs of the past. The catalog has a name and a set of lines. A given distributor or supplier will typically have many catalogs, each having one or more lines.
Line:	A Line is a category of components from a specific brand, supplier, or manufacturer. For our purposes, information about a line includes the name, the code, and a set of components.
Component:	A component is a specific part type, such as alternator or water pump. The terms “Components” and “Part Types” are often used interchangeably. The ACES standard uses the term “Component”, and since iShop 3 supports the ACES standard, iShop 3 will use this term as well.
Part:	A part is an individual record which describes the attributes of a component that goes on a vehicle.

Implementation

The three distinct parts to the Line Filtering functionality are described below:

Catalog and Component Information Retrieval

Catalog and component information retrieval is implemented by the *CatalogInformation* and *ComponentInformation* operations respectively.

Configuration Management

Configuration management is implemented by operations: *LFConfigurationList*, *LFConfigurationRetrieve*, *LFConfigurationAdd*, *LFConfigurationUpdate*, and *LFConfigurationDelete*.

Applying Line Filters

Operations *SetLineFiltering* and *SetLineFilteringConfiguration* operations set the line filtering that should be in effect when navigation results are returned. The former operation names a line filtering configuration stored on the information server for the current user. The latter operation specifies the filter explicitly rather than referencing a configuration.

Each of the above operations specifies whether filtered-out parts should be returned within navigation results. If filtered-out parts are returned, their filtered-out status is indicated by element “NavigationResults / PartInformation / PartsFilteredOut” which may be returned by any operation that returns navigation results – see section *Process Overview*. If filtered-out parts are NOT returned, they simply will not appear in a NavigationResults structure.

Process Overview – using line filtering configurations

The client application would typically get a list of existing line filtering configurations and allow the user to select one. It would then call the *SetLineFiltering* operation, passing in the selected configuration and identifying whether filtered-out parts should or should not be returned. If filtered-out parts are returned, the client will check the *PartsFilteredOut* flag in the return results to determine if it should display the part or not.

If the line filtering configuration doesn’t exist, or the user wants to create a new one, the client application will first call the *CatalogInformation* operation to get back a list of catalogs and line codes. Then for each line code the user is interested in, the client will request the component information for that line code. This is a list of the components represented by that line code. The client application will then let the user decide to include or exclude each (or all) components for the line code.

When all done, the client will then call the *ConfigurationAdd* operation to store the configuration.

Examples of line filtering are included in section *Process Overview*.

Security and Licensing

Security and Licensing are related subjects for iShop information servers since both control access to the server and server functionality. In particular, licensing may determine who may be authenticated by security, and what their authorization levels are once authenticated.

Security

iShop 3 Information Server security is implemented by including user ID and password values in an *OpenRequest* message sent to the *Open* operation that starts an information server session. The user ID and password may be protected during transmission using transport level security – i.e. HTTPS (this is recommended).

The *GetAdmin* operation response message, *GetAdminResponse*, optionally describes server security requirements for the information server (see element *SecurityRequirement*).

The OASIS WS-Security standard for securing Web services was considered for IS security, however at the time of iShop 3.0 release, implementations of this technology were not sufficiently mature and platform/technology interoperability was not assured.

WS-Security may be adopted by iShop in a subsequent release if: (1) WS-Security implementations and interoperability stabilize, and (2) the iShop community calls for this addition.

Further information on WS-Security is available at <http://www.oasis-open.org>.

Licensing

Information servers may limit access to only licensed users, and for licensed users, only to a designated set of workstations. The iShop 3 standard supports this objective through means described below.

The input message to the information server *Open* operation provides two elements to support licensing (see section *OpenRequest: MachineKey* and *MachineName*).

The *MachineKey* element should contain a hardware key that will distinguish one computer from another. The *MachineName* element should contain a description allowing the client to easily identify the licensed computer.

Computing a MachineKey Value

In an ideal world, each computer would have its own unique identifier, but this is not true. So, we need to create something which should be relatively unique to each computer and easy to determine. iShop 3 will

use the *Media Access Control* (MAC) address built into every network card. MAC addresses are guaranteed to be unique – i.e. no two network cards will have the same value.

It must be considered, however, that computers may have multiple network cards. For example, a laptop may have an Ethernet card built in for a wired network, and also may have a wireless card. Depending on where in the shop it is used, either network card might be in use at any one time.

A MAC address is of the form aa.bb.cc.dd.ee.ff where each pair of letters represents a hexadecimal digit. In other words, a MAC address is 6 bytes long, with each byte separated by a period. The MachineKey value for a computer should be constructed by concatenating together all of the computer's MAC addresses, in ascending order alphabetically, and separating each address with a pipe symbol (|). All letter hexadecimal characters (A-F) must be uppercase.

See the iShop 3 Technical Implementation Guide for sample code for extracting MAC address values, and for other implementation-level issues.

Obtaining Secured Access

A user may need to obtain access to an information server if either the user or the particular workstation is not licensed to use the information server.

In both cases, authentication will fail on the user's first attempted Web service call. The standard response structure (page 103) should indicate an "authentication failed" error. The information server can help the client obtain access by returning a web link in the "Response / MessageList / Message / URL" element alongside the other "authentication failed" elements. The client application should launch this web page for the user, who can then have an interactive conversation with the information server to negotiate access. If the negotiation outcome obtains access for the user, the user can close the web browser and then use the client application to attempt access to the information server again.

Direct Communication between a Client and Information Server

Even if the user has secured access to an information server, and a session has been successfully established, the information server may want to convey some information to the user or solicit information. In this case the information server can return a URL in the response to the Web service call that starts a session, even if no error is reported. The URL is returned in the standard response structure encapsulated in an "Information" or "Warning" severity message – i.e. the "Response / MessageList / Message / URL" element. The client application should launch this web page for the user, who can then view the information server's message or provide the feedback requested.

Deployment

The following principles apply to client iShop client software finding and accessing an iShop server:

- No server software should have to be installed on the client computer in order to access the server.
- The client computer should not have to be configured with the server's location or URL.
- iShop servers should be able to be found as soon as they are made available without any intervention on the client computer.

iShop client programs find Information Servers over the internet using UDDI (Universal Description, Discovery, and Integration) or on a local network using UDP (User Datagram Protocol). Information servers are external to the shop in most cases, and therefore UDDI applies in most cases.

UDDI is a platform-independent, XML-based registry for businesses worldwide to list themselves and find each other on the Internet. UDDI listings can also define how the listed Web service should be used. UDDI is an OASIS standard.

Although UDP is an unreliable internet protocol, it is an efficient and suitable method of sending short messages. It is used simply to find one or more iShop servers. As mentioned above, UDP is not typically used to find Information Servers.

UDP/UDDI are used only to find an iShop server. Once found, Web service communication uses standard HTTP over TCP/IP.

The iShop 3 Technical Implementation Guide contains detailed technical information on this process.

Operations

The namespace for all XML elements, unless otherwise noticed, is:

<http://www.aftermarket.org/iShop/V3/XMLSchema>

CatalogInformation

This operation retrieves information about the catalogs available from the information server, including the lines they contain. It is used when creating or updating line filtering configurations.

CatalogInformationRequest

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<CatalogInformationRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
</CatalogInformationRequest>
```

CatalogInformationResponse

CatalogList: This element is optional.

Catalog: This element is optional and may be repeated any number of times. It contains information on a catalog.

Name: This element is required. It contains the name of a catalog. The value of this field is specific to the information server.

Code: This element is required. It contains the code for a catalog, which is an abbreviated version of the name. The value of this field is specific to the information server.

LineList: This element is required.

Line: This element is optional and may be repeated any number of times. It contains information on a line within a catalog.

Name: This element is required. It contains the name of a line. The value of this field is specific to the information server.

Code: This element is required. It contains the code for a line, which is an abbreviated version of the name. The value of this field is specific to the information server.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<CatalogInformationResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
<CatalogList>
    <Catalog>
        <Name>The Widget Company</Name>
        <Code>TWC</Code>
        <LineList>
            <Line>
                <Name>Automotive Widget 1</Name>
                <Code>AW1</Code>
            </Line>
        </LineList>
    </Catalog>
</CatalogList>
<Response>
    <Status>OK</Status>
</Response>
</CatalogInformationResponse>
```

Close

This operation is called when you are done with the information server. It closes your session and allows the server to free up any resources used internally. The session should have been opened by the Open operation.

CloseRequest

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<CloseRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
</CloseRequest>
```

CloseResponse

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<CloseResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Response>
    <Status>OK</Status>
  </Response>
</CloseResponse>
```

ComponentInformation

This operation retrieves information about the components (a.k.a. part types) available from the information server. In particular, it returns information about components within a catalog and line. Like the *CatalogInformation* operation, this operation is used in conjunction with the line filtering functionality.

ComponentInformationRequest

CatalogCode: This element is required. It contains the catalog code containing the line for which components are requested.

LineCode: This element is required. It contains the line code within the specified catalog for which components are requested.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ComponentInformationRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <CatalogCode>TWC</CatalogCode>
  <LineCode>AW1</LineCode>
</ComponentInformationRequest>
```

ComponentInformationResponse

ComponentList: This element is optional.

Component: This element is optional and may be repeated any number of times. It contains information on a component.

Name: This element is required. It contains the name of a component.

Id: This element is required. It contains the code, or identifier, of a component.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ComponentInformationResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <ComponentList>
    <Component>
      <Name>Brake Pads</Name>
      <Id>453232</Id>
    </Component>
  </ComponentList>
  <Response>
    <Status>OK</Status>
  </Response>
</ComponentInformationResponse>
```

GetAdmin

The *GetAdmin* operation is identical to its counterpart in the Shop Management Server. See section *GetAdmin* for full details.

GetInformationTypes

This operation returns a list of information types that the information server can serve or that the information server has been asked to serve. The set of possible information types are documented below (examples: repair information, tire fitment information, buyers guides, etc.).

Use the *SetInformationTypes* operation to restrict information types to a subset that the user is interested in.

GetInformationTypesRequest

This element requires an attribute 'Type'. Permitted values are 'all' and 'active'. When type is 'all', the information server will respond with the list of information types that it supports. When type is 'active', the information server will respond with the list of information types that it has been requested to serve. The set of active information types will be a subset of the set of all information types.

Vehicle: This element is optional. If specified, this element should represent the current navigation vehicle. The requested operation should not change the current navigation vehicle, however the server may want the current navigation vehicle passed in as a way to preserve state information.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetInformationTypesRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd" Type="all">
</GetInformationTypesRequest>
```

GetInformationTypesResponse

InformationTypeList: This element is required.

InformationType: This element is optional and may be repeated any number of times. It contains the name of an information type. Permitted values are:

- RepairInformation
- TireFitment
- ComponentLocations
- Diagrams

iShop – Functional Implementation Guide

- TechnicalServiceBulletins
- FluidCapacities
- BatteryInformation
- Specifications
- OEParts
- LaborInformation
- AftermarketParts
- AftermarketAccessories
- ShopSupplies
- Tires
- Tools
- BuyersGuide

Vehicle: This element is optional. The values returned represent the current navigation vehicle.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example – Success

```
<?xml version="1.0" encoding="UTF-8"?>
<GetInformationTypesResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <InformationTypeList>
    <InformationType>LaborInformation</InformationType>
    <InformationType>OEParts</InformationType>
```

```

    <InformationType>RepairInformation</InformationType>
  </InformationTypeList>
</Response>
  <Status>OK</Status>
</Response>
</GetInformationTypesResponse>

```

Example – Error: Not Logged In

```

<?xml version="1.0" encoding="UTF-8"?>
<GetInformationTypesResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <InformationTypeList>
    <InformationType>LaborInformation</InformationType>
    <InformationType>OEParts</InformationType>
    <InformationType>RepairInformation</InformationType>
  </InformationTypeList>
</Response>
  <Status>OK</Status>
</Response>
</GetInformationTypesResponse>

```

GetNavigationPath

This operation returns a list of navigation items from the navigation root to the current navigation item in the information server hierarchy. This list represents a path in the tree, and not a history of all nodes navigated both up and down.

GetNavigationPathRequest

Vehicle: This element is optional. If specified, this element should represent the current navigation vehicle. The requested operation should not change the current navigation vehicle, however the server may want the current navigation vehicle passed in as a way to preserve state information.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetNavigationPathRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
</GetNavigationPathRequest>
```

GetNavigationPathResponse

NavigationPath: This element is required. It contains

NavigationPathItem: This element is optional and may be repeated any number of times. It contains a navigation item on the path from the start of navigation to the current navigation item, complete with its grouping information. The start-of-navigation item appears first.

NavigationGroup: This element is required. It represents the group belonged to by the navigation item listed below.

Note that other items may belong to the same group, and these items would have been selectable during navigation, however they are not on the navigation path to the current navigation item.

Type: This element is optional. It contains a generic value representing a type for the grouped items. This is mostly for the information server's benefit, however may be leveraged by a smart client application.

Title: This element is optional. It contains the title that was displayed to the user when the navigation item listed below was selected.

URL: This element is optional and its value must be a valid URI. It contains a link to a web page that presents the group and its navigation items to the user. This is the same navigation information as was presented when the user selected the navigation item listed below. The web page supports interactive navigation, for example drilling up and down through navigation items. The web

page may also contain hyperlinks to other information, either within the server or outside of the server.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

NavigationItem: This element is required. It contains a navigation item on the path from the start of navigation to the current navigation item. The group belonged to by this navigation item is listed above.

Text: This element is optional. It contains the description of the navigation item that the user selected during navigation.

ServerItemId: This element is optional. It contains a value meaningful to the information server application only. The information server application uses this value to identify the navigation item.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

The values returned represent the current navigation vehicle.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>  
<GetNavigationPathResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"  
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
```



```

xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <NavigationPath>
    <NavigationPathItem>
      <NavigationGroup>
        <Title>All Vehicle Components</Title>
      </NavigationGroup>
      <NavigationItem>
        <Text>Brakes and Traction Controls</Text>
        <ServerItemId>123</ServerItemId>
      </NavigationItem>
    </NavigationPathItem>
    <NavigationPathItem>
      <NavigationGroup>
        <Title>Brakes and Traction Controls Components</Title>
      </NavigationGroup>
      <NavigationItem>
        <Text>Brake Pads</Text>
        <ServerItemId>234</ServerItemId>
      </NavigationItem>
    </NavigationPathItem>
  </NavigationPath>
  <Response>
    <Status>OK</Status>
  </Response>
</GetNavigationPathResponse>

```

LFConfigurationAdd

This operation adds a new line configuration to the information server's storage for the current user. The configuration will be available for later retrieval. At most 32767 configurations may be stored for a user.

LFConfigurationAddRequest

Configuration: This element is required. It contains standard Line Configuration elements. This element is fully described in the *LineFilteringConfigurationType* reused structure section.

The configuration name as given by element Name should not already be saved on the information server for the current user.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<LFConfigurationAddRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Configuration>
    <Name>Sample</Name>
    <Description>This is a sample line configuration.</Description>
    <FilteringInformation>
      <CatalogCode>TWC</CatalogCode>
      <LineCode>AW1</LineCode>
      <ComponentList>
        <Id>123</Id>
        <Id>456</Id>
      </ComponentList>
    </FilteringInformation>
    <FilteringInformation>
      <CatalogCode>TWC</CatalogCode>
    <LineCode>AW2</LineCode>
      <AllComponents></AllComponents>
    </FilteringInformation>
    <FilteringInformation>
      <CatalogCode>TWC</CatalogCode>
      <LineCode>AW3</LineCode>
      <NoComponents></NoComponents>
    </FilteringInformation>
  </Configuration>
</LFConfigurationAddRequest>
```

```

        </FilteringInformation>
    </Configuration>
</LFConfigurationAddRequest>

```

LFConfigurationAddResponse

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<LFConfigurationAddResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
    <Response>
        <Status>OK</Status>
    </Response>
</LFConfigurationAddResponse>

```

LFConfigurationDelete

This operation deletes a line configuration from the information server's storage for the current user.

LFConfigurationDeleteRequest

Name: This element is required. It contains the name of the line configuration to delete.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```

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

```

```
<LFConfigurationDeleteRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
    <Name>Sample</Name>
</LFConfigurationDeleteRequest>
```

LFConfigurationDeleteResponse

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<LFConfigurationDeleteResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
    <Response>
        <Status>OK</Status>
    </Response>
</LFConfigurationDeleteResponse>
```

LFConfigurationList

This operation retrieves a list of line configuration saved to the information server's storage for the current user. Only the identifier is retrieved for each line configuration. Operation *LFConfigurationRetrieve* is used to retrieve the details.

LFConfigurationListRequest

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<LFConfigurationListRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
</LFConfigurationListRequest>
```

LFConfigurationListResponse

ConfigurationList: This element is optional.

Name: This element is optional and may be repeated any number of times. It contains the name of a line configuration saved to the information server's storage for the current user.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<LFConfigurationListResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <ConfigurationList>
    <Name>Sample 1</Name>
    <Name>Sample 2</Name>
  </ConfigurationList>
  <Response>
```

```

    <Status>OK</Status>
  </Response>
</LFConfigurationListResponse>

```

LFConfigurationRetrieve

This operation retrieves the full details of a named line configuration from the information server's storage for the current user. The line configuration name may have been retrieved by the *LFConfigurationList* operation.

LFConfigurationRetrieveRequest

Name: This element is required. It contains the name of the line configuration to retrieve.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<LFConfigurationRetrieveRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
  xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
  xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Name>Sample</Name>
</LFConfigurationRetrieveRequest>

```

LFConfigurationRetrieveResponse

Configuration: This element is optional. It contains standard Line Configuration elements. This element is fully described in the *LineFilteringConfigurationType* reused structure section.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

iShop – Functional Implementation Guide

```
<?xml version="1.0" encoding="UTF-8"?> <LFConfigurationRetrieveResponse
xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Configuration>
    <Name>Sample</Name>
    <Description>This is a sample line configuration.</Description>
    <FilteringInformation>
      <CatalogCode>TWC</CatalogCode>
      <LineCode>AW1</LineCode>
      <ComponentList>
        <Id>123</Id>
        <Id>456</Id>
      </ComponentList>
    </FilteringInformation>
    <FilteringInformation>
      <CatalogCode>TWC</CatalogCode>
      <LineCode>AW2</LineCode>
      <AllComponents></AllComponents>
    </FilteringInformation>
    <FilteringInformation>
      <CatalogCode>TWC</CatalogCode>
      <LineCode>AW3</LineCode>
      <NoComponents></NoComponents>
    </FilteringInformation>
  </Configuration>
  <Response>
    <Status>OK</Status>
  </Response>
</LFConfigurationRetrieveResponse>
```

LFConfigurationUpdate

This operation updates a line configuration saved on the information server's storage for the current user.

LFConfigurationUpdateRequest

Configuration: This element is required. It contains standard Line Configuration elements. This element is fully described in the *LineFilteringConfigurationType* reused structure section.

The configuration name as given by element Name must already be saved on the information server for the current user.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<LFConfigurationUpdateRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Configuration>
    <Name>Sample</Name>
    <Description>This is a sample line configuration.</Description>
    <FilteringInformation>
      <CatalogCode>TWC</CatalogCode>
      <LineCode>AW1</LineCode>
      <ComponentList>
        <Id>123</Id>
        <Id>456</Id>
      </ComponentList>
    </FilteringInformation>
    <FilteringInformation>
      <CatalogCode>TWC</CatalogCode>
    <LineCode>AW2</LineCode>
    <AllComponents></AllComponents>
  </FilteringInformation>
```



```

    <FilteringInformation>
      <CatalogCode>TWC</CatalogCode>
      <LineCode>AW3</LineCode>
      <NoComponents></NoComponents>
    </FilteringInformation>
  </Configuration>
</LFConfigurationUpdateRequest>

```

LFConfigurationUpdateResponse

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<LFConfigurationUpdateResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
  xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
  xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Response>
    <Status>OK</Status>
  </Response>
</LFConfigurationUpdateResponse>

```

Navigate

This operation is the primary operation of the information server.

This operation navigates to a specific navigation item in the information server hierarchy and returns a list of child navigation items and/or terminal information items found below the navigated-to item. The child navigation items are typically presented to a user who may select one to navigate to next (by invoking this operation again with the selected navigation item). The terminal information items are also presented for selection to the user, one or more of which may be the information they are ultimately seeking.

NavigateRequest

NavigationItem: This element is optional. It contains a navigation item in the information server hierarchy to navigate to. This item is usually obtained by the prior call to the Navigate operation, thus drilling down the information server hierarchy. It can also be obtained from any

other operation that return navigation items – E.g. Search, PopNavigation and GetNavigationPath.

Not specifying this element is equivalent to navigating to the current position in the information server hierarchy, thus returning the list of navigation items below the current navigation item. If there is no current position, the current position is set to the vehicle root if a vehicle is specified. Otherwise the current position is the information server root.

Text: This element is required. It contains the description of the navigation item to navigate to.

ServerItemId: This element is required. It contains a value meaningful to the information server application only. The information server application uses this value to identify the navigation item.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the Vehicle reused structure section. This element is only specified to update the current navigation vehicle in effect or update any of its attributes. An error will occur if the specified vehicle is incompatible with the navigated-to item.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

Example – Initial Navigation

```
<?xml version="1.0" encoding="UTF-8"?>
<NavigateRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
</NavigateRequest>
```

Example – Subsequent Navigation

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<NavigateRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
<NavigationItem>
  <Text>PROBE</Text>
  <ServerItemId>22</ServerItemId>
</NavigationItem>
<Vehicle>
  <AaiaLegacyId>23273</AaiaLegacyId>
</Vehicle>
</NavigateRequest>
  
```

NavigateResponse

NavigationResults: This element is required. It contains the items found by the navigation. These items can include child navigation items, which can be navigated to themselves, or terminal information that is the ultimate result of navigation.

The types of information returned is determined by the *SetInformationTypes* operation (page 99), the server's capabilities, and the *SetNavigationHierarchy* operation.

The server's capabilities are returned by the Open operation and are also available from the *GetInformationTypes* operation.

NavigationList: This element is optional. It will be omitted if the items found by the navigation include only terminal information. If included, this element contains a list of child navigation items below the navigated-to item in the navigation hierarchy. These items are usually presented to the end user, along with grouping information, to permit the end user to navigate further down the hierarchy to ultimately find the terminal information they are seeking.

NavigationGroup: This element is required. It represents the group belonged to by the navigation item listed below.

Type: This element is optional. It contains a generic value representing a type for the grouped items. This is mostly for the information server's benefit, however may be leveraged by a smart client application.

Title: This element is optional. It contains the title that should be displayed to the user when the navigation item listed below is presented selected.

URL: This element is optional and its value must be a valid URI. It contains a link to a web page that presents the group and its navigation items to the user. This is the same navigation information as represented by the navigation item listed below. The web page supports interactive navigation, for example drilling up and down through navigation items. The web page may also contain hyperlinks to other information, either within the server or outside of the server.

When a navigation response contains terminal information, the link to a corresponding web page should be represented by element *NavigationResultsURL* (child of element *NavigationResults*).

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

NavigationItem: This element is required and may be repeated more than one time. It contains a navigation item below the navigated-to item in the information server hierarchy.

Text: This element is optional. It contains the descriptive text that should be presented to the user as their option for navigating to this navigation item.

ServerItemid: This element is optional. It contains a value meaningful to the information server application only. The information server application uses this value to identify the navigation item.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

ArticleInformation: This element is optional and may be repeated any number of times. It contains a list of URLs to relevant articles.

URL: This element is optional and its value must be a valid URI. It contains a link to a web page that presents the relevant article. This element is deprecated in favor of element *NavigationResultsURL* (child of element *NavigationResults*).

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

BatteryInformation: This element is optional and may be repeated any number of times. It contains a list of battery specifications.

BatteryPerformance: This element is optional. It contains a string representing the primary performance parameter of the battery. Typically this will be the cold-cranking amps specification. Example: “625 CCA”.

BCICode: This element is optional. It contains the BCI Code for the battery.

Note: This element is optional. It contains any additional data about the battery specifications.

See important information on note content and display in section Appendix C – Note Fields. The appendix documentation on updating note values does not apply in this case.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

BuyersGuideInformation: This element is optional and may be repeated any number of times. It contains

ApplicationNote: This element is optional. It contains any application specific note about the part.

CatalogName: This element is optional. It contains the name of the catalog this part appears in.

LineCode: This element is optional. It contains the catalog’s line code for this part.

ComponentName: This element is optional. It contains the component (AKA Part Type) name for this part.

EngineName: This element is optional. It contains the vehicle engine identification string.

ManufacturerName: This element is optional. It contains the vehicle manufacturer name.

ModelName: This element is optional. It contains the vehicle model identification string.

PartNumber: This element is optional. It contains the part number of the part.

POPCode: This element is optional. It contains the popularity code for this part.

Price: This element is optional. It contains pricing information for the part.

ListPrice: This element is optional and its value must be decimal. It contains the suggested retail price for the part - I.e. price to the consumer. This element supports optional attributes 'Currency' and 'UOM'. Any string values are permitted.

DistributorPrice: This element is optional and its value must be decimal. It contains the price the distributor charges for the part.

This element supports optional attributes 'Currency' and 'UOM'. Any string values are permitted.

JobberPrice: This element is optional and its value must be decimal. It contains the price the jobber charges for the part. This element supports optional attributes 'Currency' and 'UOM'. Any string values are permitted.

CoreCharge: This element is optional and its value must be decimal. It contains the core charge for the part.

This element supports optional attributes 'Currency' and 'UOM'. Any string values are permitted.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

YearRange: This element is optional. It contains the year range of the vehicles the part applies to.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

FluidCapacityInformation: This element is optional and may be repeated any number of times. It contains a list of fluid capacity specifications.

Name: This element is optional. It contains the name or description of the fluid.
Example: "Engine Oil".

Specification: This element is optional. It contains the manufacturer's recommended fluid specifications. Examples: "SAE 10W/50 API SF", "ATF DEXRON II".

Note: This element is optional. It contains manufacturer’s notes or comments pertaining to fluid capacity information. Example: “Do not overfill”.

This may also be used to qualify the applicability of the data. Example: “applies to 5 liter engine only”.

See important information on note content and display in section Appendix C – Note Fields. The appendix documentation on updating note values does not apply in this case.

StandardHeavyCode: This element is optional. It contains a value indicating whether the fluid is standard or heavy duty. Some manufacturers have a code designation however few supply this information. Default values are: “S” for Standard, “H” for Heavy Duty.

Volume: This element is optional and its value must be decimal. It contains the manufacturer’s recommended fluid volume. Example: “4.5”. The element should be empty if no data is available.

This element requires an attribute ‘UOM’. Any string value is permitted.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

LaborInformation: This element is optional and may be repeated any number of times. It contains a list of labor operations.

MessageList: This element is optional. It contains a list of messages for the labor operation. These messages are to be displayed in conjunction with the labor operation description (and other related values) in order to clarify the information presented.

Message: This element is optional and may be repeated any number of times. It contains a message to display in conjunction with other labor operation values.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Description: This element is optional. It contains the description of the labor operation.

SkillLevelCode: This element is optional. It contains the skill level code for the labor operation. Typically this is a single letter. Examples: “A”, “B”, “C”, etc. There is no standard iShop code for skill level; this is dependent on the server provider.

SkillLevelDescription: This element is optional. It contains the skill level description for the labor operation. Typically this would be a string description. Example: “Highly Skilled”. There is no standard iShop description; this is dependent on the server provider.

StandardTime: This element is optional and its value must be decimal. It contains the standard time to perform the labor operation. The standard time should be specified as a decimal number without units indicating the number of hours and fractional hours. Typically this is specified in hours and tenths of hours. Example: 1.3 rather than 1.25.

WarrantyTime: This element is optional and its value must be decimal. It contains the warranty time to perform the labor operation. The standard time should be specified as a decimal number without units indicating the number of hours and fractional hours. Typically this is specified in hours and tenths of hours. Example: 1.3 rather than 1.25.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

PartInformation: This element is optional and may be repeated any number of times. It contains a list of parts.

Manufacturer: This element is optional. It contains the name of the part manufacturer.

MessageList: This element is optional. It contains a list of messages for the part. These messages are to be displayed in conjunction with the part description (and other related values) in order to clarify the information presented.

Message: This element is optional and may be repeated any number of times. It contains a message to display in conjunction with other part values.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Price: This element is optional. It contains pricing information for the part.

ListPrice: This element is optional. It contains the suggested retail price for the part - i.e. price to the consumer.

This element supports optional attributes ‘Currency’ and ‘UOM’. Any string values are permitted.

DistributorPrice: This element is optional. It contains the price the distributor charges for the part.

This element supports optional attributes 'Currency' and 'UOM'. Any string values are permitted.

JobberPrice: This element is optional. It contains the price the jobber charges for the part.

This element supports optional attributes 'Currency' and 'UOM'. Any string values are permitted.

CoreCharge: This element is optional. It contains the core charge for the part.

This element supports optional attributes 'Currency' and 'UOM'. Any string values are permitted.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Description: This element is optional. It contains the description of the part.

GraphicReference: This element is optional and its value must be a valid URI. It contains a link to a web page that presents the graphic or multimedia view of the part.

ManufacturerPartNumber: This element is optional. It contains the manufacturer's part number for the part.

Quantity: This element is optional and its value must be decimal. It contains the quantity of the part normally required. Value should be -1 if a default quantity is not available.

This element requires an attribute 'UOM'. Any string value is permitted.

ServerComponentId: This element is optional and its value must be a non-negative integer. It contains the component (part type) id for the part. This is used when configuring the line filtering configuration.

ServerCatalogId: This element is optional and its value must be a non-negative integer. It contains the catalog id value for the part. This is used when configuring the line filtering configuration.

Availability: This element is optional and may be repeated any number of times. It contains a value indicating where the part is available within the supply chain. Permitted values are:

Table 10 - Availability Codes

Values	Description
NotFound	The part was not found anyplace in the supply chain.
Store	The part is normally stocked in the store.
DC	The part is available at the distribution center.
Order	The part is only available via special order.

HotSpotIndex: This element is optional and its value must be a non-negative integer. It contains a value that should match a value in a “ImageInformation / HotSpotInformation / HotSpotIndex” entity. The value then matches the represented part to a particular image hot spot.

PartIsFilteredOut: This element is optional. It contains a flag to indicate if line filtering is active and the part has been filtered out. Values permitted are 'true', 'false', '1', or '0'. Absence of this element should be interpreted as 'false' or '0'.

A client can indicate when setting a line filter whether filtered out parts should be omitted or returned as 'filtered out'. See operations *SetLineFiltering* and *SetLineFilteringConfiguration*.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

MaintenanceInformation: This element is optional and may be repeated any number of times. It contains a list of maintenance data.

Component: This element is optional. It contains the name of a standard component or part. Example: “Engine oil filter”. A vehicle system can also be represented as a component. Example: “Cooling System”.

DrivingCondition: This element is optional. It contains manufacturer specific indication of any special driving conditions that are associated with this maintenance interval.

Example: “Normal”, “Severe”.

KilometerInterval: This element is optional and its value must be a positive integer. It contains the recommended maintenance interval between services in kilometers. The *KilometerInterval* and *MileageInterval* element values should correspond.

MileageInterval: This element is optional and its value must be a positive integer. It contains the recommended maintenance interval between services in miles. The *KilometerInterval* and *MileageInterval* element values should correspond.

Action: This element is optional. It indicates what action is required of the technician. Examples: Inspect, Replace, Service.

Note: This element is optional. It contains notes to display to the technician.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

ImageInformation: This element is optional and may be repeated any number of times. It contains

URL: This element is optional and its value must be a valid URI. It contains a link to a web page that presents the image.

This element is deprecated in favor of element *NavigationResultsURL* (child of element *NavigationResults*).

HotSpotInformation: This element is optional. It contains information on a hot spot within the image. The user should be able to click on the hot spot and be linked to related information.

X: This element is optional and its value must be a non-negative integer. It contains the X coordinate of the hot spot. The unit of measure is pixels.

Y: This element is optional and its value must be a non-negative integer. It contains the Y coordinate of the hot spot. The unit of measure is pixels.

Width: This element is optional and its value must be a positive integer. It contains the width of the hot spot. The unit of measure is pixels.

Height: This element is optional and its value must be a positive integer. It contains the X height of the hot spot. The unit of measure is pixels.

HotSpotIndex: This element is optional and its value must be a non-negative integer. It contains a numeric value identifying the hot spot. This value should match a *HotSpotIndex* value in another *NavigationResults* child element, thus identifying the hot spot related information. At present, only the *PartInformation* element supports this.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

SpecificationInformation: This element is optional and may be repeated any number of times. It contains specification data.

Note: This element is optional. It contains a note for this specification.

See important information on note content and display in section Appendix C – Note Fields. The appendix documentation on updating note values does not apply in this case.

Value: This element is optional and its value must be decimal. It contains the value of this specification.

This element supports an optional attribute 'UOM'. Any string value is permitted.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

TireFitmentInformation: This element is optional and may be repeated any number of times. It contains a list of tire fitment specifications.

LugnutTorque: This element is optional and its value must be decimal. It contains the vehicle manufacturer's recommended wheel lug nut torque for factory wheels.

This element requires an attribute 'UOM'. Any string value is permitted.

Material: This element is optional. It contains the material the wheel is constructed from. Examples: "Steel", "Aluminum".

Note: This element is optional. It contains manufacturer's notes or comments pertaining to wheel and tire information.

See important information on note content and display in section *Appendix C – Note Fields*. The appendix documentation on updating note values does not apply in this case.

Position: This element is optional. It contains the physical location or position of the wheel and tire on the vehicle. This is a vendor-defined string. An abbreviation is used. Examples: "F" for front, "R" for rear.

Pressure: This element is optional and its value must be decimal. It contains the vehicle manufacturer's recommended air pressure for factory tires under normal driving conditions. Example: "32".

This element requires an attribute 'UOM'. Any string value is permitted.

Group: This element is optional. It contains a server assigned number or identifier associating this tire with other tires in a vehicle set.

TireSize: This element is optional. It contains the vehicle manufacturer's factory original equipment tire size. Example: "185/70HR14".

WheelSize: This element is optional. It contains the vehicle manufacturer's factory original equipment wheel size. Example: "15x6".

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

TSBInformation: This element is optional and may be repeated any number of times. It contains a list of Technical Service Bulletins.

Name: This element is optional. It contains the name of a Technical Service Bulletin.

Date: This element is optional and its value must be a valid date. It contains the date the Technical Service Bulletin was issued.

The date value must be formatted as per ISO 8601, 'CCYY-MM-DD'. A time zone indicator is permitted - see the ISO 8601 specification for details.

Number: This element is optional. It contains the number of a Technical Service Bulletin.

Synopsis: This element is optional. It contains the synopsis, or summary, of a Technical Service Bulletin.

This data follows the same rules as notes. See important information on note content and display in section Appendix C – Note Fields (page 112). The appendix documentation on updating note values does not apply in this case.

URL: This element is optional and its value must be a valid URI. It contains a link to a web page that presents the Technical Service Bulletin.

This element is deprecated in favor of element `NavigationResultsURL` (child of element `NavigationResults`).

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

NavigationResultsURL: This element is optional and its value must be a valid URI. A value should only be specified when the navigation response contains terminal information, and the value should be a link to a web page that presents this terminal information. This web page may contain hyperlinks to other information, either within the server or outside of the server.

When a navigation response contains other navigation items rather than terminal information, the link to a corresponding web page should be represented by element “`NavigationList / NavigationGroup / URL`”.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the Vehicle reused structure section (page 104).

The values returned represent the current navigation vehicle. This operation may have changed attributes of the current navigation vehicle, and if so, the changes will be reflected in the values returned.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example – Navigation Items Only

iShop – Functional Implementation Guide

```
<?xml version="1.0" encoding="UTF-8"?>
<NavigateResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
<NavigationResults>
  <NavigationList>
    <NavigationGroup>
      <Type>NavigationPartGroups</Type>
      <Title>AZ Parts(SM)</Title>
      <URL>http://localhost/iShopServer/12345</URL>
    </NavigationGroup>
    <NavigationItem>
      <Text>Brakes & Traction Control</Text>
      <ServerItemId>1236</ServerItemId>
    </NavigationItem>
    <NavigationItem>
      <Text>Climate Control</Text>
      <ServerItemId>997</ServerItemId>
    </NavigationItem>
    <NavigationItem>
      <Text>Collision, Body Parts & Hardware</Text>
      <ServerItemId>1114</ServerItemId>
    </NavigationItem>
    <NavigationItem>
      <Text>Drivetrain</Text>
      <ServerItemId>1056</ServerItemId>
    </NavigationItem>
    <NavigationItem>
      <Text>Electrical</Text>
      <ServerItemId>976</ServerItemId>
    </NavigationItem>
    <NavigationItem>
      <Text>External Engine</Text>
```

```
<ServerItemId>1189</ServerItemId>
</NavigationItem>
<NavigationItem>
  <Text>Fuel Delivery</Text>
  <ServerItemId>685</ServerItemId>
</NavigationItem>
<NavigationItem>
  <Text>Internal Engine</Text>
  <ServerItemId>947</ServerItemId>
</NavigationItem>
<NavigationItem>
  <Text>Powertrain & Engine Management</Text>
  <ServerItemId>1189</ServerItemId>
</NavigationItem>
<NavigationItem>
  <Text>Routine Maintenance</Text>
  <ServerItemId>787</ServerItemId>
</NavigationItem>
<NavigationItem>
  <Text>Starting, Charging & Tune-up</Text>
  <ServerItemId>976</ServerItemId>
</NavigationItem>
<NavigationItem>
  <Text>Suspension & Steering</Text>
  <ServerItemId>998</ServerItemId>
</NavigationItem>
<NavigationItem>
  <Text>Tools & Manuals</Text>
  <ServerItemId>1007</ServerItemId>
</NavigationItem>
<NavigationItem>
  <Text>Trim Accessories</Text>
  <ServerItemId>1016</ServerItemId>
</NavigationItem>
</NavigationList>
```



```

</NavigationResults>
<Vehicle>
  <AaiaLegacyId>23273</AaiaLegacyId>
</Vehicle>
<Response>
  <Status>OK</Status>
</Response>
</NavigateResponse>

```

Example – Navigation Items and Terminal Data

```

<?xml version="1.0" encoding="UTF-8"?>
<NavigateResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
<NavigationResults>
  <NavigationList>
    <NavigationGroup>
      <Type>NavigationParts</Type>
      <Title>BRAKE PADS - FRONT </Title>
      <URL>http://localhost/iShopServer/12345</URL>
    </NavigationGroup>
    <NavigationItem>
      <Text>Related Parts</Text>
      <ServerItemId>-1</ServerItemId>
    </NavigationItem>
    <NavigationItem>
      <Text>BRAKE ROTOR - FRONT </Text>
      <ServerItemId>854</ServerItemId>
    </NavigationItem>
    <NavigationItem>
      <Text>BRAKE DISC HARDWARE KIT </Text>
      <ServerItemId>582</ServerItemId>

```

```

    </NavigationItem>
  </NavigationList>
  <PartInformation>
    <Manufacturer>
      <Name>DURALAST</Name>
      <Code>EPA</Code>
    </Manufacturer>
    <MessageList>
      <Message>SEMI-METALLIC</Message>
      <Message>*** CONSISTENT AND EFFECTIVE BRAKING WITH SHIMS
INCLUDED WHERE APPLICABLE ***</Message>
    </MessageList>
    <Price>
      <ListPrice>62.48</ListPrice>
    </Price>
    <ManufacturerPartNumber>MKD637</ManufacturerPartNumber>
    <ServerComponentId>1233</ServerComponentId>
    <ServerCatalogId>137</ServerCatalogId>
  </PartInformation>
  <PartInformation>
    <Manufacturer>
      <Name>PFC_CARBON-METALLIC</Name>
      <Code>PFC</Code>
    </Manufacturer>
    <MessageList>
      <Message>SUPERIOR STOPPING PERFORMANCE.</Message>
    </MessageList>
    <Price>
      <ListPrice>0.0000</ListPrice>
    </Price>
    <ManufacturerPartNumber>6374</ManufacturerPartNumber>
    <ServerComponentId>1233</ServerComponentId>
    <ServerCatalogId>188</ServerCatalogId>
  </PartInformation>
  <PartInformation>

```

```

<Manufacturer>
  <Name>MORSE-CERAMIC</Name>
  <Code>PFC</Code>
</Manufacturer>
<MessageList>
  <Message>*** CERAMIC FRICTION FOR QUIETER PERFORMANCE,
VIRTUALLY DUST - FREE BRAKING ***</Message>
</MessageList>
<Price>
  <ListPrice>103.48</ListPrice>
</Price>
<ManufacturerPartNumber>637C</ManufacturerPartNumber>
<ServerComponentId>1233</ServerComponentId>
<ServerCatalogId>1061</ServerCatalogId>
</PartInformation>
<PartInformation>
  <Manufacturer>
    <Name>DURALAST-GOLD</Name>
    <Code>EPA</Code>
  </Manufacturer>
  <MessageList>
    <Message>DESIGNED FOR MAXIMUM DURABILITY AND STOPPING
POWER.</Message>
  </MessageList>
  <Price>
    <ListPrice>83.23</ListPrice>
  </Price>
  <ManufacturerPartNumber>DG637</ManufacturerPartNumber>
  <ServerComponentId>1233</ServerComponentId>
  <ServerCatalogId>2521</ServerCatalogId>
</PartInformation>
<PartInformation>
  <Manufacturer>
    <Name>VALUCRAFT</Name>
    <Code>EPV</Code>
  </Manufacturer>

```

```

</Manufacturer>
<MessageList>
  <Message>SEMI-METALLIC</Message>
  <Message>SMOOTH, QUIET STOPPING.</Message>
</MessageList>
<Price>
  <ListPrice>34.98</ListPrice>
</Price>
<ManufacturerPartNumber>MKD637V</ManufacturerPartNumber>
<ServerComponentId>1233</ServerComponentId>
<ServerCatalogId>47</ServerCatalogId>
</PartInformation>
<PartInformation>
  <Manufacturer>
    <Name>PFC-Z-RATED</Name>
    <Code>PFZ</Code>
  </Manufacturer>
  <Price>
    <ListPrice>0.0000</ListPrice>
  </Price>
  <ManufacturerPartNumber>637Z</ManufacturerPartNumber>
  <ServerComponentId>1233</ServerComponentId>
  <ServerCatalogId>200</ServerCatalogId>
</PartInformation>
<PartInformation>
  <Manufacturer>
    <Name>PFC_CARBON-METALIC</Name>
    <Code>PFC</Code>
  </Manufacturer>
  <MessageList>
    <Message>SUPERIOR STOPPING PERFORMANCE.</Message>
  </MessageList>
  <Price>
    <ListPrice>103.48</ListPrice>
  </Price>

```

```

    <ManufacturerPartNumber>0637-20</ManufacturerPartNumber>
    <ServerComponentId>1233</ServerComponentId>
    <ServerCatalogId>2590</ServerCatalogId>
  </PartInformation> </NavigationResults>
<Vehicle>
  <AaiaLegacyId>23273</AaiaLegacyId>
</Vehicle>
<Response>
  <Status>OK</Status>
</Response>
</NavigateResponse>

```

Open

This operation opens an information server session. The information server will authenticate the user as described in section Security (page 58). User licensing may also be enforced as per section Licensing.

The standard response structure will indicate an error if a session could not be opened.

OpenRequest

UserName: User ID used by the information server for authentication and establishing a session.

Password: Password used by the information server for authentication and establishing a session.

MachineKey: This element is optional. It contains a unique physical client computer identifier that is used by the information server for license enforcement. Information server licensing is documented in section *Licensing*.

MachineName: This element is optional. It contains the descriptive name for the client computer. This is used by the information server for license enforcement. Information server licensing is documented in section *Licensing*.

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

If specified, the information server will consider it the default vehicle for the session and navigation will begin at the vehicle's root in the information server hierarchy. If not specified, navigation will begin at the information server root.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <MachineKey>00.0D.56.34.AD.FE|00.90.4B.19.8C.25|44.45.53.54.42.00</MachineKey>
  <MachineName>mystere</MachineName>
</OpenRequest>
```

OpenResponse

InformationTypeList: This element is required. It contains a list of information types that the information server can serve.

InformationType: This element is optional and may be repeated any number of times. It contains the name of an information type. See the *GetInformationTypesResponse* message for a list of permitted values.

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

The values returned represent the current navigation vehicle. This operation may have changed attributes of the current navigation vehicle, and if so, the changes will be reflected in the values returned.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<OpenResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <InformationTypeList>
    <InformationType>AftermarketParts</InformationType>
    <InformationType>AftermarketAccessories</InformationType>
  </InformationTypeList>
  <Response>
    <Status>OK</Status>
  </Response>
</OpenResponse>
```

PopNavigation

This operation backs up a specified number of levels from the current navigation item in the information server hierarchy toward the navigation root. Note that the navigation root is determined by the current navigation vehicle, if any. The server backs up to the navigation root if the number of levels specified is greater than the current navigation depth.

This operation returns the same list of child navigation items and/or child terminal information items that was returned when this navigation item was originally navigated to.

3.2.14.1 PopNavigationRequest

Count: This element is required and its value must be a non-negative integer. It contains the number of navigation levels to back-out. All levels will be backed-out if the value is 0.

Vehicle: This element is optional. If specified, this element should represent the current navigation vehicle. The requested operation should not change the current navigation vehicle, however the server may want the current navigation vehicle passed in as a way to preserve state information.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<PopNavigationRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
    <Count>1</Count>
</PopNavigationRequest>
```

PopNavigationResponse

NavigationResults: This element is the same as documented for the *NavigationResults* element of message *NavigateResponse*. It contains the same results as would be obtained by navigating to the new current node.

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

The values returned represent the current navigation vehicle. This operation may have changed attributes of the current navigation vehicle, and if so, the changes will be reflected in the values returned.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures for full details*.

Example

See example for *NavigateResponse*.

Search

This operation searches through the information server database and returns a list of navigation items and/or terminal information items that match user specified criteria. Navigation items found are presented to a user who may select one to navigate to next. Terminal information items found are also presented for selection to the user, one or more of which may be the information they are ultimately seeking.

A Search string and a vehicle must be specified when searching.

Information returned information from this operation is the same as the information returned from the *Navigate* operation.

This operation will reset the navigation hierarchy to the starting point (i.e. *GetNavigationPath* would return an empty path).

SearchRequest

SearchCriteria: This element is required. It contains a simple string for the purpose of text searching.

Vehicle: This element is required. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

The search results will only return results applicable to the vehicle specified, and this vehicle will become the current navigation vehicle for the session.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SearchRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <SearchCriteria>suspension</SearchCriteria>
  <Vehicle>
    <AcesVehicle>
      <vehicle:BaseVehicle id="18296"></vehicle:BaseVehicle>
      <vehicle:EngineBase>1460</vehicle:EngineBase>
    </AcesVehicle>
  </Vehicle>
</SearchRequest>
```

SearchResponse

SearchResults: This element is the same as documented for the *NavigationResults* element of message *NavigateResponse*. It contains all items that match the input search criteria filtered by the input vehicle.

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

The values returned represent the current navigation vehicle. This operation may have changed attributes of the current navigation vehicle, and if so, the changes will be reflected in the values returned.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

See example for *NavigateResponse*.

SetInformationTypes

This operation limits the types of information the information server will return back to the client. The set of possible information types are referenced in documentation below (examples: repair information, tire fitment information, buyers guides, etc.).

Use the *GetInformationTypes* operation to see which information types the server supports, and which are currently active.

SetInformationTypes is typically called before navigating.

This operation will reset the navigation hierarchy to the starting point (i.e. *GetNavigationPath* would return an empty path).

3.2.16.1 SetInformationTypesRequest

InformationTypeList: This element is required. It contains a list of information types that the information server is requested to serve.

InformationType: This element is optional and may be repeated any number of times. It contains the name of an information type. See the *GetInformationTypesResponse* message (page 65) for a list of permitted values.

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

If specified, the information server will consider it the default vehicle for the session and navigation will be reset to the vehicle's root in the information server hierarchy. If not specified, navigation will be reset to the information server root.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SetInformationTypesRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <InformationTypeList>
    <InformationType>LaborInformation</InformationType>
    <InformationType>OEParts</InformationType>
    <InformationType>RepairInformation</InformationType>
  </InformationTypeList>
</SetInformationTypesRequest>
```

SetInformationTypesResponse

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

The values returned represent the current navigation vehicle. This operation may have changed attributes of the current navigation vehicle, and if so, the changes will be reflected in the values returned.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<SetInformationTypesResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Response>
    <Status>OK</Status>
  </Response>
</SetInformationTypesResponse>

```

SetLineFiltering

This operation enables or disables line filtering. Line filtering is enabled by passing the name of a line filtering configuration stored on the information server for the current user. Line filtering is disabled by passing an empty configuration name.

Line filtering limits the part information returned to the client – see section *Line Filtering* for comprehensive documentation.

Line filtering can be set using either of the *SetLineFiltering* or *SetLineFilteringConfiguration* operations.

3.2.17.1 SetLineFilteringRequest

ConfigurationName: This element is optional. It contains the name of the line filtering configuration that is stored on the information server for the current user.

Line filtering is disabled if this element is missing.

ReturnFilteredOutParts: This element is optional. It contains a flag to indicate whether filtered out parts should be returned in the PartInformation element when results are returned. If filtered-out parts are returned, the PartIsFilteredOut element will mark its filtered-out status.

Values permitted are 'true', 'false', '1', or '0'.

Vehicle: This element is optional. If specified, this element should represent the current navigation vehicle. The requested operation should not change the current navigation vehicle, however the server may want the current navigation vehicle passed in as a way to preserve state information.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SetLineFilteringRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <ConfigurationName>Sample</ConfigurationName>
  <ReturnFilteredOutParts>>false</ReturnFilteredOutParts> </SetLineFilteringRequest>
```

SetLineFilteringResponse

Vehicle: This element is optional. The values returned represent the current navigation vehicle.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SetLineFilteringResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Response>
    <Status>OK</Status>
  </Response>
</SetLineFilteringResponse>
```

SetLineFilteringConfiguration

This operation enables or disables line filtering.

This operation is similar to the *SetLineFiltering* operation, except that it enables line filtering by passing an actual configuration element instead of the name of a configuration stored on the server for the current user. Clients might use this operation if the client stores the user's configurations rather than letting the server store them, or if the client builds a configuration dynamically.

Line filtering is disabled by omitting the configuration element in the request message.

Line filtering limits the part information returned to the client – see section *Line Filtering* for comprehensive documentation.

Line filtering can be set using either of the *SetLineFiltering* or *SetLineFilteringConfiguration* operations.

SetLineFilteringConfigurationRequest

Configuration: This element is required. It contains standard Line Configuration elements. This element is fully described in the *LineFilteringConfigurationType* reused structure section.

The element values specified will set the active line filter. Line filtering is disabled if this element is missing.

ReturnFilteredOutParts: This element is optional. It contains a flag to indicate whether filtered out parts should be returned in the *PartInformation* element when results are returned. If filtered-out parts are returned, the *PartsFilteredOut* element will mark its filtered-out status.

Values permitted are 'true', 'false', '1', or '0'.

Vehicle: This element is optional. If specified, this element should represent the current navigation vehicle. The requested operation should not change the current navigation vehicle, however the server may want the current navigation vehicle passed in as a way to preserve state information.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?> <SetLineFilteringConfigurationRequest
xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Configuration>
```

```

    <Name>Sample</Name>
    <Description>This is a sample line configuration.</Description>
    <FilteringInformation>
      <CatalogCode>A</CatalogCode>
      <LineCode>1</LineCode>
      <ComponentList>
        <Id>123</Id>
        <Id>456</Id>
      </ComponentList>
    </FilteringInformation>
    <FilteringInformation>
      <CatalogCode>A</CatalogCode>
      <LineCode>2</LineCode>
      <AllComponents></AllComponents>
    </FilteringInformation>
    <FilteringInformation>
      <CatalogCode>B</CatalogCode>
      <LineCode>1</LineCode>
      <NoComponents></NoComponents>
    </FilteringInformation>
  </Configuration>
  <ReturnFilteredOutParts>>false</ReturnFilteredOutParts>
</SetLineFilteringConfigurationRequest>

```

SetLineFilteringConfigurationResponse

Vehicle: This element is optional. The values returned represent the current navigation vehicle.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures for full details*.

Example

```
<?xml version="1.0" encoding="UTF-8"?> <SetLineFilteringConfigurationResponse
xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Response>
    <Status>OK</Status>
  </Response>
</SetLineFilteringConfigurationResponse>
```

SetNavigationHierarchy

This operation allows the user to specify an alternative navigation hierarchy. Value “Default” represents the server’s standard navigational hierarchy. Since the hierarchy also defines the data set available to the user, using any other value might limit the data available to the client. For example, if the user sets the navigation type to TSBNumber and calls the *Navigate* method, then only TSB information is returned, and only TSB information will be available without resetting the navigation hierarchy. This differs from the *SetInformationTypes* operation in that *SetInformationTypes* filters out data the user is not interested in. When using *SetNavigationHierarchy*, data outside of the selected hierarchy is just not available.

SetNavigationHierarchy is typically called before navigating.

This operation will reset the navigation hierarchy to the starting point (i.e. *GetNavigationPath* would return an empty path).

3.2.19.1 SetNavigationHierarchyRequest

NavigationHierarchy: This element is required. It contains a value specifying which navigational hierarchy the server should use. Permitted values are:

- Default
- ManufacturerComponent
- ManufacturerCatalog
- ManufacturerPartNumber
- AAIACategoryName

- AAIACategoryNumber
- AAIAServiceCategory
- TSBNumber
- ServerComponent
- ServerCategory

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

If specified, the information server will consider it the default vehicle for the session and navigation will be reset to the vehicle's root in the information server hierarchy. If not specified, navigation will be reset to the information server root.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SetNavigationHierarchyRequest xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <NavigationHierarchy>Default</NavigationHierarchy>
</SetNavigationHierarchyRequest>
```

SetNavigationHierarchyResponse

Vehicle: This element is optional. It contains the standard Vehicle elements. This element is fully described in the *Vehicle* reused structure section.

The values returned represent the current navigation vehicle. This operation may have changed attributes of the current navigation vehicle, and if so, the changes will be reflected in the values returned.

Response: This element is required. It contains the standard response structure. See section *Global XML Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SetNavigationHierarchyResponse xmlns="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:ishop="http://www.aftermarket.org/iShop/V3/XMLSchema"
xmlns:vehicle="http://www.aftermarket.org/iShop/V3/ACESVehicle"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aftermarket.org/iShop/V3/XMLSchema iShopMessages.xsd">
  <Response>
    <Status>OK</Status>
  </Response>
</SetNavigationHierarchyResponse>
```

Information Server Reused Structures

The XML structures described in this section are used in more than one Information Server message.

The namespace for all XML elements, unless otherwise noticed, is:

<http://www.aftermarket.org/iShop/V3/XMLSchema>

LineFilteringConfigurationType

All usages of this structure name the top-level element 'Configuration'. Child elements are as follows.

Name: This element is required. It contains the name of the line filtering configuration. This value must not contain more than 80 characters.

Description: This element is optional. It describes the line filter in a manner more descriptive than the name.

FilteringInformation: This element is optional and may be repeated any number of times. It contains a line filter that is part of this configuration.

CatalogCode: This element is required. It contains the code for a catalog.

LineCode: This element is required. It contains the code for a line within the identified catalog.

Begin Filter Choice 1 (ComponentList) – one choice must be selected

ComponentList: This element is required. It contains the Id of one or more components that should be included by this filter.

Id: This element is required and may be repeated any number of times. It contains a component that should be included by this filter.

Begin Filter Choice 2 (AllComponents) – one choice must be selected

AllComponents: This element is required and should not contain a value. Its presence indicates that ALL components for the catalog line should be included by this filter.

Begin Filter Choice 3 (NoComponents) – one choice must be selected

NoComponents: This element is required and should not contain a value. Its presence indicates that NO components for the catalog line should be included by this filter.

End Filter Choice 3 (NoComponents) – one choice must be selected

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

UserArea: This element is optional. It contains custom content. See section *Global Reused Structures* for full details.

Example

<Configuration>

<Name>Sample</Name>

<Description>This is a sample line configuration.</Description>

<FilteringInformation>

<CatalogCode>TWC</CatalogCode>

<LineCode>AW1</LineCode>

<ComponentList>

<Id>123</Id>

<Id>456</Id>

</ComponentList>

</FilteringInformation>

<FilteringInformation>

<CatalogCode>TWC</CatalogCode>

<LineCode>AW2</LineCode>

```

    <AllComponents></AllComponents>
  </FilteringInformation>
  <FilteringInformation>
    <CatalogCode>TWC</CatalogCode>
    <LineCode>AW3</LineCode>
    <NoComponents></NoComponents>
  </FilteringInformation>
</Configuration>

```

Global XML Structures

The XML structures described in this section are used in both SMS and Information Server messages.

The namespace for all XML elements, unless otherwise noted, is:

<http://www.aftermarket.org/iShop/V3/XMLSchema>

UserArea

The user area contains any XML content accepted by the iShop or custom schemas. Content from custom schemas should be contained within an element named after the company that defines the content.

A usage of this structure always names the top-level element 'UserArea'.

Example

```

<UserArea>
  <Acme>
    <acme:SpecialInstructions>abc</acme:SpecialInstructions>
  </Acme>
</UserArea>

```

Response

A usage of this structure always names the top-level element 'Response'. Child elements are as follows.

Status: This element is required. Values permitted are 'OK' and 'Fail'.

MessageList: This element is optional. It contains any number of message structures returned from the server to the client.

Message: This element is optional if 'Status' (above) is 'OK' and is required if 'Status' is 'Fail'. This element may be repeated any number of times. Each occurrence represents a message returned from the server to the client.

Severity: This element is required. It contains the severity of the condition that produced this message. Valid values are: 'Information', 'Warning' and 'Error'.

Code: This element is required and its value must be a non-negative integer. It contains a unique numeric message identifier. Values less than 10,000 represent standard errors, larger than 10,000 represent custom errors. *See Appendix A – Standard Messages.*

ShortDescription: This element is optional. It contains a brief version of the message. Message text is not standardized.

Description: This element is optional. It contains a full version of the message. Message text is not standardized.

Descriptions may include multi-line text using word wrap and/or CRLF end of line markers. No other control characters may be included.

URL: This element is optional and its value must be a valid URI. It contains a link to a web page that presents any type of information associated with this message. The consumer of the response may navigate to this link or give the user a choice whether to do so.

UserArea: This element is optional. It contains custom content. *See section Global Reused Structures for full details.*

UserArea: This element is optional. It contains custom content. *See section Global Reused Structures for full details.*

Example

```
<Response>
  <Status>Fail</Status>
  <MessageList>
    <Message>
      <Severity>Error</Severity>
```

```

    <Code>10</Code>
    <ShortDescription>Short message description.</ShortDescription>
    <Description>Full message description.</Description>
  </Message>
</MessageList> </Response>

```

Vehicle

The Vehicle structure identifies a physical vehicle that belongs to a customer and that can be the subject of a repair order. It also identifies the vehicle type using either an Auto Care Association legacy id or an ACES structure. ACES is the preferred vehicle identification mechanism going forward, however Auto Care Association legacy id is supported for compatibility with software that does not yet support ACES.

The ACES databases are available from the Auto Care Association on a subscription basis. At the time of this writing, the following excerpt from the subscription agreement governs embedding the databases in a subscriber's software product. Before doing so, however, the software vendor must read and consider the actual subscription agreement that they have signed.

“Subscribers are free to incorporate the Auto Care Association supporting tables in a software product or Internet application provided acknowledgement of the Auto Care Association copyright and reasonable precautions are taken to protect the tables from unauthorized redistribution from the application.”

Child elements are as follows.

Auto Care AssociationLegacyId: At least one of the AaiaLegacyId and AcesVehicle elements are required. Therefore a vehicle type must be identified by either a legacy Auto Care Association Id or an ACES structure. If both are specified, then AcesVehicle is assumed to be primary, and it is assumed that AaiaLegacyId identifies the same vehicle type.

This element contains a legacy Auto Care Association vehicle Identifier. It should only be used to retain compatibility with software that does not yet support ACES.

AcesVehicle: At least one of the AaiaLegacyId and AcesVehicle elements are required. Therefore a vehicle type must be identified by either a legacy Auto Care Association Id or an ACES structure. If both are specified, then AcesVehicle is assumed to be primary, and it is assumed that AaiaLegacyId identifies the same vehicle type.

This element identifies and describes a vehicle type using ACES elements and values.

Child element BaseVehicle, and optionally EngineBase, should be used to identify the vehicle type. Other elements contain lookup information and may be populated for information purposes. Accordingly, BaseVehicle and EngineBase should be saved in database structures that store a physical vehicle. Lookup attributes may be stored for convenience; however the lookup values in the ACES database should be considered the primary source of this information.

The structure for this element has been adapted from the “aces.xsd” schema distributed as part of the “ACES Delivery Specifications”, Version 1.07. The objective was to adapt as little as possible, however the following changes had to be made to make it usable for iShop:

- Elements “Header” and “Footer” were removed.
- Most child elements of element “App” were copied to become child elements of iShop element “AcesVehicle”. The child elements NOT copied are: Note, Qty, PartType, MfrLabel, Position, Part, DisplayOrder.
- The schema “choice” between BaseVehicle and Years/Make/etc. was modified such that BaseVehicle is required, and Years/Make/etc. can also be specified. This was desired because, although iShop intends to identify a vehicle type by BaseVehicle, the Years/Make/etc. information should be available, and iShop does not want to require an application to look up to an ACES database to get it.

Please refer to the above mentioned ACES documentation, section 4.10 Tags, for general descriptions of the child elements listed below. Descriptions are attached to the elements listed below to the extent that they describe the relevance of the element to iShop.

The namespace for all child XML elements is:

<http://www.aftermarket.org/iShop/V3/ACESVehicle>

This namespace is different than the iShop standard to accommodate the future possibility that the schema defining the ACES elements will be provided directly from the ACES group.

BaseVehicle: This element is required. This element is used within iShop to identify the vehicle type.

Years: This element is optional.

Make: This element is required if a value is specified for the Years element.

VehicleType: This element is optional. If specified, the Years and Make elements must be specified, and this element value applies to it.

Model: This element is optional. If specified, the Years and Make elements must be specified, and this element value applies to it.

SubModel: This element is optional. The original ACES structure had a separate SubModel element to apply to BaseVehicle and Model, however the iShop adaptation has only this one element.

MfrBodyCode: This element is optional.

BodyNumDoors: This element is optional.

BodyType: This element is optional.

DriveType: This element is optional.

EngineBase: This element is optional. This element is used within iShop to identify the vehicle type.

EngineDesignation: This element is optional.

EngineVIN: This element is optional.

EngineVersion: This element is optional.

EngineMfr: This element is optional.

FuelDeliveryType: This element is optional.

FuelDeliverySubType: This element is optional.

FuelSystemControlType: This element is optional.

FuelSystemDesign: This element is optional.

Aspiration: This element is optional.

CylinderHeadType: This element is optional.

FuelType: This element is optional.

IgnitionSystemType: This element is optional.

TransmissionMfrCode: This element is optional.

Begin Transmission Identification Choice 1 (TransmissionBase) – one choice or none should be selected

TransmissionBase: This element is optional.

Begin Transmission Identification Choice 2 (TransmissionType, ..., TransmissionNumSpeeds) – one choice or none should be selected

TransmissionType: This element is optional.

TransmissionControlType: This element is optional.

TransmissionNumSpeeds: This element is optional.

End Transmission Identification Choice 2 (TransmissionType, ..., TransmissionNumSpeeds) – one choice or none should be selected

TransmissionMfrCode: This element is optional.

TransmissionMfr: This element is optional.

TransferCaseBase: This element is optional.

TransferCase: This element is optional.

TransferCaseMfr: This element is optional.

BedLength: This element is optional.

BedType: This element is optional.

WheelBase: This element is optional.

BrakeSystem: This element is optional.

FrontBrakeType: This element is optional.

RearBrakeType: This element is optional.

BrakeABS: This element is optional.

FrontSpringType: This element is optional.

RearSpringType: This element is optional.

SteeringSystem: This element is optional.

SteeringType: This element is optional.

RestraintType: This element is optional.

Region: This element is optional.

PhysicalVehicle: This element contains properties of a physical vehicle as opposed to elements AaiaLegacyId and AcesVehicle that identify and describe a vehicle type (i.e. make/model).

VehicleId: This element is required. It contains a unique string identifying the physical vehicle. Typically this is the database primary key for the physical vehicle.

GovernmentId: This element is optional. It contains the government supplied vehicle type identifier.

InspectionDate: This element is optional and its value must be a valid date and time. It contains the inspection date for the vehicle. Normally this is the date that the vehicle was last inspected.

The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

LastInDate: This element is optional and its value must be a valid date and time. It contains the date the vehicle was last in the shop, if the shop management system maintains this information.

The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

LastOdometer: This element is optional and its value must be numeric. It contains the odometer reading from the last time this vehicle was in the shop or 0 if the vehicle has not been in the shop before.

This element supports an optional attribute 'Unit'. Permitted values are 'Miles' and 'KM'.

LicensePlate: This element is optional. It contains the license plate number for the vehicle.

LicenseState: This element is optional. It contains the state or province that issued the license plate for the vehicle. Standard codes appropriate to the country are used where applicable.

LicenseCountry: This element is optional. It contains the country that issued the license plate for the vehicle. The country is specified using the ISO-3166-1 two-letter country code.

See <http://www.din.de/gremien/nas/nabd/iso3166ma/> for information on ISO-3166-1 and <http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html> for an English list.

Note: This element is optional. It contains the note associated with the vehicle.

See important information on note elements in section Appendix C – Note Fields (page 112).

UnitNumber: This element is optional. It contains the customer's unit number for the vehicle. This is most commonly used on fleet vehicles.

VIN: This element is optional. It contains the Vehicle Identification Number for the vehicle if it has been previously set, otherwise the element should be either missing or empty.

This value is independent of vehicle type information provided or implied by either AaiaLegacyId or AcesVehicle.

Color: This element is optional. It contains the color of the vehicle paint.

PhotoList: This element is optional. It contains any number of photo references for the vehicle. Photo references are included rather than actual photos to save message space. Photos are retrieved using the GetPhoto and GetPhotoList operations.

Photo: This element is optional and may be repeated any number of times. It contains a photo reference for the vehicle.

Photoid: This element is required. It contains a unique string identifying the photo. Typically this is the database primary key for the photo.

DateCreated: This element is optional and its value must be a valid date and time. It contains the date and time the photo was created.

The date and time value must be formatted as per ISO 8601, 'CCYY-MM-DDThh:mm:ss'. Fractional seconds and time zone indicator are permitted - see the ISO 8601 specification for details.

Description: This element is optional. It contains a description of the photo.

Descriptions may include multi-line text using word wrap and/or CRLF end of line markers. No other control characters may be included.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

UserArea: This element is optional. It contains custom content. See section Global Reused Structures for full details.

Example – Vehicle identified by legacy Auto Care Association Id

```

<Vehicle>
  <AaiaLegacyId>1432644</AaiaLegacyId>
  <PhysicalVehicle>
    <VehicleId>6005</VehicleId>
    <LastInDate>2005-11-25T14:00:00</LastInDate>
    <LastOdometer Unit="Miles">5971</LastOdometer>
    <LicensePlate>005TMZ</LicensePlate>
  </PhysicalVehicle>
</Vehicle>

```

Example – Vehicle identified by ACES structure

```

<Vehicle>
  <AcesVehicle>
    <vehicle:BaseVehicle id="18296"></vehicle:BaseVehicle>
    <vehicle:EngineBase>1460</vehicle:EngineBase>
  </AcesVehicle>
  <PhysicalVehicle>
    <VehicleId>6005</VehicleId>
    <LastInDate>2005-11-25T14:00:00</LastInDate>
    <LastOdometer Unit="Miles">5971</LastOdometer>
    <LicensePlate>005TMZ</LicensePlate>
  </PhysicalVehicle> </Vehicle>

```

Appendices

Appendix A – Standard Messages

Common Messages

Table 11 - Common Messages

Code	Severity	General Description (actual message set by implementer)
00001	Error	Operation aborted.
00002	Error	Requester not authenticated.
00003	Error	Authentication failed.
00004	Error	Requester not authorized.
00005	Error	Request document invalid.
00006	Error	Response document invalid.
00007	Error	Operation not implemented.
00008	Error	License has expired.
00009	Warning	License requires attention.
00010	Error	License limit has been reached (E.g. maximum number of users).
00011	Error	Data not found.
00012	Error	Unexpected error.

Shop Management Server Specific Messages

No messages are identified at this time.

Information Server Specific Messages

Table 12 - IS Specific Messages

Code	Severity	General Description (actual message set by implementer)
07000	Error	Session not open.

Appendix B – Order Search Fields

Table 13 - Order Search Fields

Search Field Name	Corresponds to Element in Order structure (page 43)
<i>Customer Fields</i>	
Company	Order / Customer / Company
FirstName	Order / Customer / FirstName
LastName	Order / Customer / LastName
<i>Order Fields</i>	
Date	Not found in the Order structure. This is a vendor determined date/time value, usually equivalent to one of the specific date/time properties in the order.
Promised	Order / Promised
Scheduled	Order / Scheduled
TimeIn	Order / TimeIn
TimeOut	Order / TimeOut
OrderNumber	Order / OrderNumber

OrderType	Order / OrderType
<i>Vehicle Fields</i>	
AAIAID	Order / Vehicle / AaiaLegacyId
LicensePlate	Order / Vehicle / PhysicalVehicle / LicensePlate
VIN	Order / Vehicle / PhysicalVehicle / VIN
BaseVehicle	Order / Vehicle / AcesVehicle / BaseVehicle
Years	Order / Vehicle / AcesVehicle / Years
Make	Order / Vehicle / AcesVehicle / Make
VehicleType	Order / Vehicle / AcesVehicle / VehicleType
Model	Order / Vehicle / AcesVehicle / Model
SubModel	Order / Vehicle / AcesVehicle / SubModel <i>This search field may only be specified when ANDed with a search by search field BaseVehicle or Model.</i>
EngineBase	Order / Vehicle / AcesVehicle / EngineBase
<i>Order Item Search Fields</i>	
<i>An order satisfies the search criteria if any of its order items match a specified order item search expression.</i>	
Description	Order / ItemList / Item / Description
EquipmentType	Order / ItemList / Item / EquipmentType

Group	Order / ItemList / Item / Group
ItemType	<i>Does not correspond to an element value. Values permitted are “Part” and “Labor”. This search field is used to find order lines that identify parts or labor.</i>
ResultExists	<i>Does not correspond to an element value. Values permitted are 'true', 'false', '1', or '0'. This search field is used to find order lines that have at least one “Order / ItemList / Item / DiagnosticResultsList / DiagnosticResult” element.</i>
Status	Order / ItemList / Item / Status
Symptom	Order / ItemList / Item / Symptom
TechnicianCode	Order / ItemList / Item / Labor / Technician / Code
TechnicianName	Order / ItemList / Item / Labor / Technician / Name
<p><i>Diagnostic Results Search Fields</i></p> <p><i>An order satisfies the search criteria if any of its order items contains a DiagnosticResult element that matches a specified diagnostic results search expression.</i></p>	
Title	Applies to “DiagnosticResult / ASANETISHOP” only Corresponds to ASANET keyword “REF / TITLE”.
Equipment Type	Applies to “DiagnosticResult / ASANETISHOP” only Corresponds to ASANET keyword “EQUIPMENT”, attribute “TYPE”.
Model	Applies to “DiagnosticResult / ASANETISHOP” only Corresponds to ASANET keyword “EQUIPMENT / MODEL”.

Manufacturer	Applies to “DiagnosticResult / ASANETISHOP” only Corresponds to ASANET keyword “EQUIPMENT / MANUFACTURER”.
--------------	--

Appendix C – Note Fields

Note Value Content and Display

Note text may be displayed in either a fixed or proportional font. Word wrapping may occur while displaying or printing the note. The maximum length of a note is 32767 bytes. A CRLF (\r\n, 0x0d,0x0a) sequence may be included to indicate the end of a line. No other control characters may appear in the note.

Updating Note Values

All update operations append the value in Note elements to the end of any existing value. For example, consider a Vehicle document is retrieved using the *GetVehicleById* operation. The note element in the retrieved document will be populated with the current note value. The client updates the Vehicle document and requests the updates be applied using the *ChangeVehicleById* operation. Any value that is in the Note element at this time will be appended to the end of the note value stored in the database.

Important: Prior to invoking *ChangeVehicleById*, the client should either clear the retrieved note value, or insert new text which will be appended to the existing value. The note value in the database will be doubled if the user took no action with respect to the value of the retrieved note element.

A corollary of this is that note text cannot be edited or removed by the client.