XML Reference / Implementation

Version 4.0

Document Revision History			
Revision Date	Effective Date	Version	Revision Description
12/19/2002	1/01/2003	1.0	First published version
4/29/2003	7/04/2003	2.0	Updated Field Types, OAGI verb definitions, frequently asked questions and repository location section.
5/18/2004	7/04/2004	3.0	Updated frequently asked questions, added Application Area information to BOD Implementation section (7/16/2003) Added information on flattened BODs, updated the FAQ, modified the schema repository descriptions to match OAGI structure, removed the Create Verb, added information about Reliability in BOD Implementation section (9/30/2003) Changed name to "XML Reference/Implementation"; Updated verb definitions; Added OAGIS Application Area to the STAR schema strategy; Updated The ApplicationArea definitions; Moved verb definitions to the Implementation section and added scenarios to them; expanded definition of field population; updated flattened BOD information; removed BPSS as a deliverable (3/24/2004) Added Best Practices Recommendations from the Transport Guidelines (5/18/2004)
4/26/2005	7/05/2005	4.0	Updated FAQs (7/20/2004); Updated FAQs (8/03/2004/); Updated FAQs; Added trademarked logo; added new field definitions to Application Area, updated Implementation Guidelines section, clarified whitespace section (12/14/2004); Updated schema verb location information; removed the OAGIS schema as a requirement; updated FAQs; changed schema location in example to have relative addressing; updated Data Type definitions

Table of Contents

1 Introduction

The STAR BODs (STAROAGI) are based upon the Open Applications Group Incorporated (OAGI) development methodology. OAGI is a not-for-profit industry consortium whose purpose is to promote interoperability among Business Software Applications and to create and/or endorse one or more standards for easier business software interoperability. Their focus is on the concept of a Business Object Document (BOD) and its associated business processes used to exchange data between business components. Some of the advantages to STAR of using the OAGI proven methodology are:

- Architecture consistency
- Message Consistency
- Dictionary Use Consistency
- High Re-use
- Faster Development
- Smaller Learning Curve

1.1 Purpose

The purpose of this document is to define general information to assist the developers understanding of XML, STAR's schema design approach and also provide assistance for the implementation of BODs.

1.2 Objective

The objective is for all STAR members to use this document for the implementation of the approved STAROAGI BODs without modification. If any extensions or modifications are made to the STAROAGI BODs they would not be considered STAR standard.

1.3 Scope

This document applies to all STAR members that desire to be compliant with the STAR BODs.

2 Reference

2.1 Documents

- STAR approved XML BODs Repository: http:// www.STARStandard.org
- Subset of The OAGI XML BODs Repository: http:// www.STARStandard.org
- STAR approved XML Implementation Guidelines:
- http:// www.STARStandard.org
- Complete OAGI XML BODs Repository: http://www.openapplications.org
- The Open Applications Group Worldwide web site: http://www.openapplications.org
- Open Applications Group Integration Specification, Release 8.0, 05 April 2002 http://www.openapplications.org

2.2 Contacts

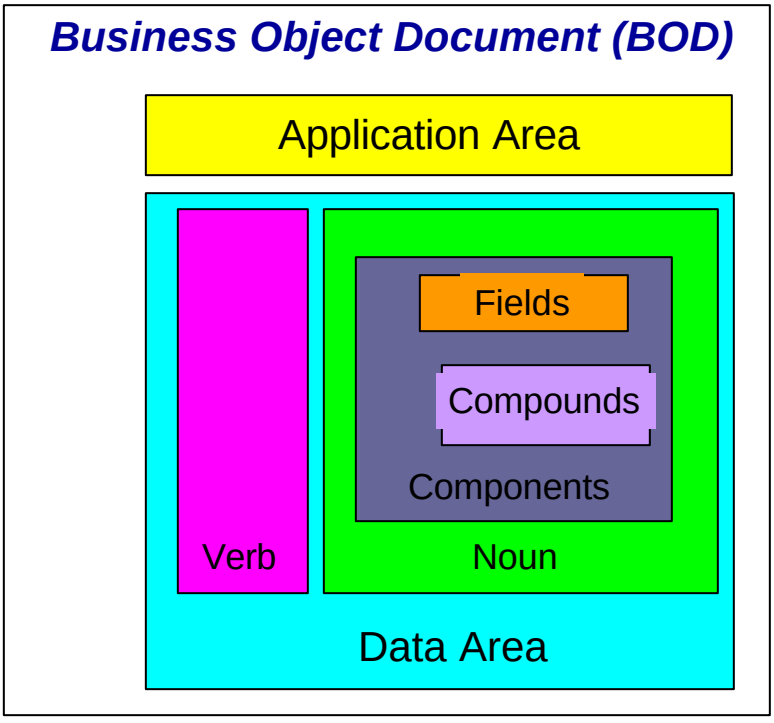
Please send any comments and suggestions pertaining to this document to Patricia Shih at PatriciaShih@aol.com or Michelle Vidanes at mvidanes@nada.org

3 BOD Basics

3.1 BOD Definition

BODs (Business Object Documents) are the business messages or business documents that exchange data between applications and components between companies. The BOD provides a common horizontal message architecture across multiple industries, with Automotive retail being just one such vertical industry. The BOD is a business message that is exchanged between application components and is independent of the communication mechanism. BODs can be used with various types of transport protocols such as SOAP and ebXML. However, the BOD Message Architecture is independent of the communication mechanism.

All BODs have two high-level parts: An Application Area and the Data Area.



3.2 Application Area

The BOD Application Area communicates information that can be used by the infrastructure to communicate the message. This includes information on the Sender of the BOD, date and time the BOD was created and information on the Destination of the BOD.

3.3 Data Area

The BODs Business Data Area includes a definition of the data, making it a self-describing message format. It can contain one or more occurrences of the data values. The data area is made up of the Verb and Noun.

3.3.1 Verb

The Verb identifies the action being performed on the specific Noun of the BOD. (i.e., Dealer initiates with a GET PartsOrder, OEM responds with an SHOW PartsOrder)

There is a predefined list of verbs (provided by OAGI) that STAR uses. This is information that is derived directly from OAGI and is applied to the STAR Nouns. Verbs constrain, add behavior and implied process to the Noun. Refer to *Verb Usage &Process Flows* in the BOD Implementation section of this document.

3.3.2 Noun

The Noun identifies the business specific data that is being communicated (i.e., Parts Order, Repair Order, Credit Application, etc). Noun behavior is affected by the associated Verb. The Noun is where the individual data requirements are defined. The Noun consists of Components, Compounds and Fields.

3.3.2.1 Components

Components are the building blocks of a Noun (i.e. PartsOrder Header, PartsOrder Line, Address, etc.). The components allow us to organize the data requirements in logical groupings. (i.e., Vehicle Component contains all information requirements associated with a vehicle; Demographics contains all demographics associated with an individual, etc.)

STAR's objective when designing components is to develop as many reusable components as possible.

```
<xs:complexType name="Demographics">
    <xs:sequence>
        <xs:element name="Gender" type="Gender" minOccurs="0"/>
        <xs:element name="Language" type="Language" minOccurs="0"/>
        <xs:element name="BirthDate" type="BirthDate" minOccurs="0"/>
        <xs:element name="FirstTimeBuyerInd" type="FirstTimeBuyerInd" minOccurs="0"/>
        <xs:element name="PreferredLanguage" type="PreferredLanguage" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

3.3.2.2 Compounds

Compounds are basic shared building blocks that are used by all BODs (i.e., Quantity, Amount, etc.) Compounds are a logical grouping of fields (low level elements) that are used across BODs. The contain complex content (element children)

```
<xs:complexType name="Quantity">
    <xs:simpleContent>
        <xs:extension base="xs:decimal">
            <xs:attribute name="uom" type="UOM" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

3.3.2.3 Fields

The lowest level elements defined in OAGIS. Fields are fundamental elements that are used to create Compounds and Components. (i.e. Description, Name, etc.). They contain simple content (no element children). All fields are defined as a specific “Type” which defines their characteristics. Refer to *STAR Field Type Definitions* in the Appendix of this document.

```
<xs:simpleType name="FamilyName">
    <xs:restriction base="Name"/>
</xs:simpleType>
```

4 XML

4.1 What is XML

XML (eXtensible Markup Language) is a text-based mark-up language for structuring data content into a self-describing wrapper. It was designed to create documents and data records that are fully portable and platform independent. XML is not a business language, but requires a business language to be defined within it, like a programming language.

4.2 Why XML

XML provides a much richer data capability than other approaches. It is designed for the web. XML tools provide more options for interoperability. It enables more advanced types of eBusiness connections and application integration such as web-based or process-based integration.

4.3 Schema

The XML Schema language is the Open Applications Group's Integration Specifications (OAGIS) recommended method for creating schemas. XML Schemas provide a rich syntax for expressing metadata. Some of its features include:

- Structures are defined that allow the definition of relational (keyed) data, and object-oriented (type inheritance) data.
- Schema uses elements and attributes for structural constraints
- Schema uses the same format as XML document.
- Schema uses XML Namespace support for extensibility. Namespaces enable developers to avoid naming collisions by assigning element and attribute names. An XML name is prefaced by characters defining the namespace it belongs to.

4.3.1 STAR Schema Strategy

STAR uses a global strategy to develop the schema:

- Schema are created in STAR namespace referencing OAGI and W3C namespaces for types, nouns and verbs.
- STAR uses Global field definitions instead of local. Occasionally, the local definition is enhanced with additions to the comments, but the core value of the fields are the same. (i.e., Item Number may be the Item number used on a repair or the Failed Part number on a claim). It should be noted that OAGI uses local field definitions. They are addressing a broader audience than STAR so they do not have the requirement that STAR has for an industry segment standard data dictionary.
- The STAR XML data dictionary is the Fields file.

- Most of the fields have been defined as Optional rather than Required. Only when all members of each specific workgroup agreed that a field should be Required could we define it as required.
- STAR does not support the use of the OAGIS "UserArea" which is added to all OAGIS components. Use of the "UserArea" would make each implementation unique and would be contrary to supporting common implementations across STAR members.
- Most references to namespaces are at a global level ("top level") in the Fields file, rather than local in components or noun files.
- STAR breaks data into reusable components wherever possible. STAR develops small components that can be combined as required. This is the same approach that OAGI uses.
- To keep the schema simple, STAR stays away from restriction and substitution as much as possible. Instead, STAR tries to keep components at a level where they are re-useable as they are or with an extension.
- OAGI uses XSL to apply constraints to schema. Star applies all constraints in the schema.
- As schemas are being developed, STAR has regular reviews with OAGI architects. Their feedback is incorporated into BODs before they are approved and posted on QuickPlace.
- It should be noted that while STAR follows OAGI as much as possible, there are differences which inhibit STAR from using some of their vanilla components. When this occurs, since restrictions do not work well across namespaces, STAR has to copy OAGI components into STAR namespace in order to redefine or restrict.
- Business Rules are applied to Implementation Guidelines to further constrain and clarify BODs. Examples:
    - Some fields are required for certain Verbs and not others
    - If a specific component is used, one or more of its subcomponents must be used
    - "INACTIVE" status of fields & components has been applied to Implementation Guidelines to reduce number of unused fields

4.3.1.1 OAGIS Application Area Exclusions

There are fields in the OAGIS Application Area that have not been included in the STAR namespace. The reasons they have not been included are explained as follows:

- **Sender Component**
    - **Confirmation** – OAGIS defines the ability to Confirm a BOD in two places, both in the Sender and in the verb. To avoid the confusion of which one to use, STAR has elected to specify whether to Confirm a BOD on the verb.

- **UserArea** – Since STAR does not support the OAGIS UserArea, this field has not been included in the STAR namespace

4.3.1.2 Advantages to STAR strategy:

- STAR Global Data Dictionary for definition of standard Automotive Retail terms.
- Consistency in definitions and design.
- Allows for re-use of standard fields.
- Allows for re-use of standard components.
- Allows for extension of standard components.
- Allows for easy maintenance. (i.e., only change one place)
- Allows for rapid development of new BODs through extensive re-use of Components and Nouns
- Smaller learning curve.

4.4 Naming Conventions

- Almost all names are unabbreviated.
- There is a maximum length of 30 characters.
- All element and type names are **UpperCamelCase**
  - ContactTime, Condition, DepartmentType
- All attribute names are **lowerCamelCase**
  - desc, length
- Exceptions:
  - If the abbreviation is widely used in the business world
    - VIN - An abbreviation of Vehicle Identification Number
  - Abbreviations are used if necessary to keep name within 30 character length restriction

4.5 Attribute Vs Element

One of the key schema design decisions was whether to represent a field as an XML element or attribute. Once a field has been made an attribute, it cannot be extended further; for this reason and to promote better uniformity within the Application, attributes have been used sparingly.

Following is an example showing the benefit of elements over attributes:

While yes I can define an element Car with attributes doors, engine, and color. Like so:

```
<Car doors="4" engine="4-cylinder" color="red">
```

I can not specify more details about the engine in the above. Say I want to know the manufacturer of the engine, the PowerRating of the engine and whether or not the engine is a turbo/diesel or gasoline engine. In the above example I have already defined the engine as an attribute. Sure I can add new attributes:

```
<Car doors="4" engine="4-cylinder" color="red" manufacturerName="XYZ"
type="gasoline" powerRating="200">
```

But these new attributes are not intended to be associated with the car but with the engine...I can add the Engine prefix or postfix to the attribute names (that's hokey). XML is supposed to provide the structure and doing this does not provide any structure. On another note my powerRating above says 200. 200 what? (HorsePower, DogPower, Joules, Watts, etc.) How can I indicate this above and not have it associated with the car.

The better approach is below.

```
<Car>
    <Doors>4</Doors>
    <Color>Red</Color>
    <Engine>
      <ManufacturerName>XYZ</ManufacturerName>
      <Type>gasoline</Type>
      <PowerRating uom="HorsePower">200</PoweRating>
    </Engine>
</Car>
```

5 STAR BOD Deliverables

There are three major deliverables for each STAR BOD:

- STAR Implementation Guidelines
- STAR Schema Repository
- BOD data mapping template

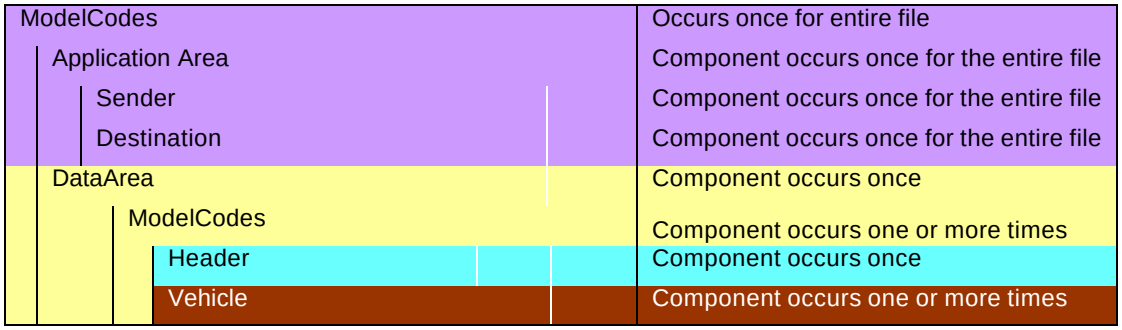
5.1 STAR Implementation Guidelines

This document is a Guideline on how to implement a BOD using the STAR schema. It explains the relationships, definitions, and valid values of the data. Users can review the guidelines to understand how their requirements are being addressed. Developers can use the guidelines to determine how to map STAR requirements to their own internal requirements.

The Implementation Guidelines are set up into three major sections of information – Relationship Diagram, Component Tables and DTS Mapping

5.1.1 Relationship Diagram

This diagram shows you the relationship of all the components within the schema. It will show you hierarchies and occurrences. It does not go down to the field or compound level (i.e., quantities, amounts, etc.). The purpose of the diagram is to help you put the word document in better perspective. Instead of just a series of tables you can see how the information in the tables relates to each other.



5.1.2 Columns in the Component Tables

This is where the detailed definition on the BOD data requirements is contained. There are several columns in each component table:

- **Field Name/Component** – This is the actual tag that corresponds to the schema. If a component rather than a field or compound is defined in this column, a notation will be in the "Comment" column saying "See xxxx Component"
- **Definition** - The definition of the field. If for a component, this column is usually left blank.

- **Required/ Optional** – STAR tries to define most fields as optional to keep the standard open and flexible. If a field is "Required, it must be populated when the XML is created. If a field is "Optional", but it is required for a specific organization, it will be their responsibility to make it "Required" within their application.
- **Valid Values or "Qualifiers"** – This is where any enumerators associated with a field are defined. It would be a qualifier if a component was being defined in the first column "Field Name/Component"
- **Business Rules/ Comments** – Comments relating to field, compound or component. When attributes are used on a compound, they will be listed here. This column can also contain examples of how data would be formatted and Business Rules that apply to the implementation of the field, compound or component.

Data Elements & Components				
Field Name/Component	**Definition**	**Required/ Optional**	**Valid Value or "Qualifier"**	**Business Rules/ Comments**
DocumentDateTime	Date and time document was created	Optional		EX:2003-11-05T13:15:30Z
star:SecondaryPassword	Secondary password used to validate access	Optional		

5.1.3 DTS Mapping Section

This is a cross reference spreadsheet from the existing DTS to the schema. This is only applicable for BODs that are based on an existing STAR DTS Interface. The purpose is to provide assistance to both OEMs and RSPs when converting to XML from batch.

Identification Record				
Position	**Data Element Name**	**Data Element Definition**	**DE #**	**Mapping to BOD**
1	Creator Name Code	DCS software creator code	100242	Application Area – Sender/CreatorNameCode
2	Creator Software Code	DCS software code name	100244	Application Area – Sender/Component
3	Interface Version	Software release version	100220	Application Area – Sender/SystemVersion
4	Transaction Create Date	Date the transaction was created	100430	Application Area - CreationDateTime
5	Transaction Create Time	Time the transaction was created	100440	Application Area - CreationDateTime
6	Dealer Number	Dealer code number	100210	Application Area - Sender or Destination/DealerNumber

5.2 STAR Schema Repository

The STAR Schema Repository contains the Schema and Sample XML for all the approved BODs.

5.2.1 Sample XML

Sample XML will provide the developers assistance in understanding how XML will look. It contains examples of the actual data tags and some sample data. It is populated with one example when valid values are specified.

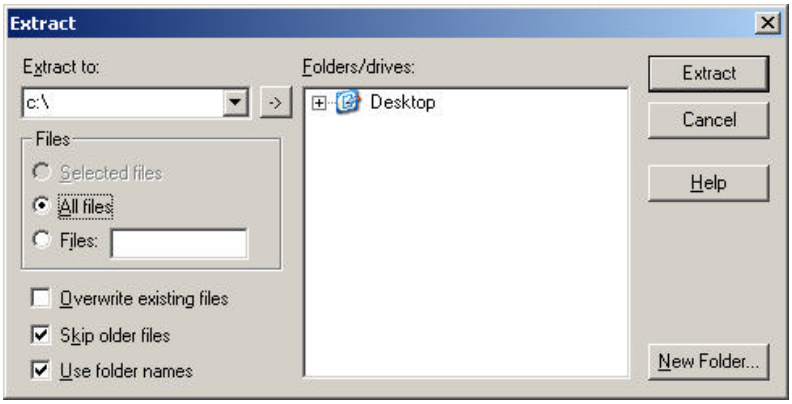
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- COPYRIGHT © 2005 STAR – STANDARDS FOR TECHNOLOGY IN AUTOMOTIVE RETAIL.  ALL RIGHTS RESERVED -->
<GetModelCodes xmlns="http://www.starstandards.org/STAR" xmlns:oa="http://www.openapplications.org/oagis" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=" ../BODs/Standalone/GetModelCodes.xsd" revision="2.0" release="8.1" environment="Production" lang="en-US">
    <ApplicationArea>
        <Sender>
            <LogicalId>String</LogicalId>
            <Component>String</Component>
            <Task>String</Task>
            <ReferenceId>String</ReferenceId>
            <AuthorizationId>String</AuthorizationId>
            <CreatorNameCode>String</CreatorNameCode>
            <SenderNameCode>AF</SenderNameCode>
            <SenderURI>http://www.altova.com</SenderURI>
            <DealerNumber>String</DealerNumber>
            <StoreNumber>String</StoreNumber>
            <AreaNumber>String</AreaNumber>
            <DealerCountry>US</DealerCountry>
            <Language>en-US</Language>
            <DeliverPendingMailInd>0</DeliverPendingMailInd>
            <Password>String</Password>
            <SystemVersion>String</SystemVersion>
        </Sender>
        <CreationDateTime>2003-11-05T13:15:30Z</CreationDateTime>
        <BODId>String</BODId>
        <Destination>
            <DestinationNameCode>TY</DestinationNameCode>
            <DestinationURI>http://www.altova.com</DestinationURI>
            <DestinationSoftwareCode>String</DestinationSoftwareCode>
            <DestinationSoftware>String</DestinationSoftware>
            <DealerNumber>String</DealerNumber>
            <StoreNumber>String</StoreNumber>
            <AreaNumber>String</AreaNumber>
            <DealerCountry>US</DealerCountry>
        </Destination>
    </ApplicationArea>
    <DataArea>
        <oa:Get confirm="Always" show="Always">
            <oa:ReturnCriteria>
                <oa:SelectExpression>String</oa:SelectExpression>
            </oa:ReturnCriteria>
        </oa:Get>
        <ModelCodes>
            <Header>
                <DocumentDateTime>2003-05-06T09:00:00Z</DocumentDateTime>
                <SecondaryPassword>String</SecondaryPassword>
                <SecondaryDealerNumber>String</SecondaryDealerNumber>
            </Header>
            <Vehicle>
```

```
            <Model>String</Model>
            <ModelYear>2002</ModelYear>
            <ModelDescription>String</ModelDescription>
            <Make>String</Make>
            <DoorsQuantity>0</DoorsQuantity>
            <BodyStyle>String</BodyStyle>
            <TransmissionType>3</TransmissionType>
            <VDSCode>String</VDSCode>
            <EngineType>String</EngineType>
            <DriveTrain>String</DriveTrain>
            <Grade>String</Grade>
            <DriveType>Front</DriveType>
        </Vehicle>
    </ModelCodes>
  </DataArea>
</GetModelCodes>
```

5.2.2 Schema Repository Location and Access

Schema is located on the STAR web site: www.STARStandard.org

- Download the schema zip file "STAR" and extract it making sure to take the option to "Use Folder Names":



- Once downloaded and extracted, open project "STAROAGI" using an IDE product for readability: **STAR/Rev2.0/STAROAGI.spp**

5.2.2.1 STAR Schema hierarchy:

STAR
- **Rev2.2**
  - **BODExamples**
  - **BODs**
    - **Standalone**
    - **Developer**
  - **Resources**
    - **Nouns**
    - **Components**
      - **Common**
        - **Components**
        - **DataTypes**
        - **Fields**

Meta
Verbs

- **STAR** is the top level folder name in the directory: **STAR**
- **Rev2.2** is the next folder down. It designates that this is the STAR 2.2 revision: **STAR/Rev2.2**
- The **Sample XML** files (Instances) are located in the BOD Example folder: **STAR/Rev2.2/BODExamples**
- The **BODs** files are located in the BODs folder: **STAR/Rev2.2/BODs**
- The **Standalone** folder contains the flattened BODs: **STAR/Rev2.2/BODs/Standalone**
- The **Developer** folder contains the structured BODs that are used for BOD Development: **STAR/Rev2.2/BODs/Developer**
- The **Resources** folder contains all the resources required for a BOD: **STAR/Rev2.2/Resources**
- The BOD **Nouns** are located in the Resources Nouns folder: **STAR/Rev2.2/Resources/Nouns**
- The **Components** folder in the Resources folder contains all the information that is referenced by the Noun schema: **STAR/Rev2.2/Resources/Components**
- The **Common** folder contains data that is common to all Nouns: **STAR/Rev2.2/Resources/Components/Common**
- The **Components** file in the Common folder contains the components that are used by the Noun schema. **STAR/Rev2.0/Resources/Components/Common/Components.xsd**
- The **DataTypes** file in the Common folder contains the data types that are referenced by the Fields and Enumerators files. **STAR/Rev2.2/Resources/Components/Common/DataTypes.xsd**
- The **Fields** file in the Common folder contains the data elements (fields) that are referenced by the components **STAR/Rev2.2/Resources/Components/Common/Fields.xsd**
- The **Meta** file in the Common folder contains the Business Object Document information that is standard for all STAR BODs **STAR/Rev2.2/Resources/Components/Common/Meta.xsd**
- The BOD **Verbs** are located in the Resources folder: **STAR/Rev2.2/Resources/Verbs**

6 BOD Implementation

6.1 BOD Implementation Recommendations

The following guidelines are suggested to assist the BOD developer in using STAR BODs.

- Understand XML and STAR BOD concepts.
- Use existing STAR BODs wherever possible.
- Use only STAR BODs (precisely as specified) that have been approved by the STAR and OAGI members.
- Work with STAR architects to include any additional specifications required to existing STARI BODs to ensure all enhancements become part of the STAR BOD standards.

6.2 Things to note about BODs

- Not all Verbs apply to all Nouns.
- BODs are designed according to documented interchange scenarios.
- BODs can be used in multiple scenarios.
- Wherever possible, STAR has reused OAGI BODs, components, compounds and fields.
- Only STAR information in the Implementation Guidelines will be included in the STAR Schema (i.e. no UserArea)

6.3 Preparing to use a BOD

- Select BOD(s) for interchange
- Develop applications that
  - Encode and Transmit BODs
    - conformant to their respective Schema Definitions
  - Receive and Process BODs
    - validating according to their respective Schema Definitions
- Exchange and process BODs according to STAR planned/agreed-upon protocol

6.4 The Life of a BOD Instance

- Creation/Generation by application,
  - according to the BOD's respective Schema
- Transmission via STAR accepted medium (Reference Infrastructure Transport recommendations)
- Receipt, validation, and checking of the BOD
  - against the BOD's respective Schema
- Use of the BOD content by the application

- Any BOD processing follow-up
  - via subsequent interchange of another BOD

6.5 STAR XML Schema Repository

6.5.1 XML Namespaces Used in BODs

There are three XML Namespaces that are used in the STAR BODs

- STAR Namespace = “star”
  NOTE: Since the STAR BODs are defined in the star: namespace, the fields and components are not prefaced by “star”
- OAGI Namespace = “oa”
- W3C Namespace = “xs”

6.5.2 Noun Version Numbering

Noun versions are numbered in format “nn.n” beginning with “1.0”. Interim versions of Nouns are identified by incrementing the published version of the modified Noun by “.01”. (i.e.; “1.21”). Noun version history is contained in the documentation at the beginning of each Noun in the STAR XML Repository

6.5.3 Schema Revision Number

Each approved publication of the STAR XML Repository will have a unique Revision number in format “nn.n” beginning with “1.0”. Interim Draft versions of Repository will be identified by incrementing the published Revision Number by “.01”. (i.e., “1.21”)

6.5.4 OAGI Release Number

STAR defines the OAGI release that the STAR XML schema is based on in the publication of the STAR BOD Specifications (STAR XML Repository and associated Implementation Guidelines)

6.5.5 Whitespace

Ignore the Whitespace that is only there for readability. It should not be in what goes over the wire.

6.5.6 Field Population

6.5.6.1 Inactive Fields

If there is a business rule on the field in the Implementation Guidelines that says a field is "Inactive", this means that there is no known usage of this field. Although this field appears as Optional in the schema, this filed is Inactive and should not be sent. If a member wants to use an Inactive field, they would need to complete the *STAR XML BOD Modification Request Form* to request that this field be activated.

6.5.6.2 Optional Fields

Optional fields do not need to be sent over the wire. Empty tags should not be sent. Sender must send all optional fields available in their system, and the Receiver will be responsible for only recognizing fields that they need. An out-of-band negotiation is required if a receiver only wants to:

1. Receive a subset of the optional fields
2. Request data not available in the sender's system
3. Request validation of non-enumerated fields

Note: an out-of-band negotiation would not be considered "non-compliant".

6.5.6.3 Required Fields

A required field must be populated with a non-null value. A required field on a component is only required to be sent when the component is sent. If an enumerated value has been defined for a required field, then the sender would need to populate the field with one of the defined enumerators.

6.5.7 Flattened BODs

All the xsd files that are used to create one BOD have been put into one file so that all the components, fields and enumerators for that BOD are found in one file, rather than multiples. Benefits of flattened BODs are 1) Better performance, 2) Easier on the tools, 3) Easier to read and 4) Stand alone to send to partners. The flattened BODs are in a folder called "Standalone".

../Rev1.12/BODs/Standalone

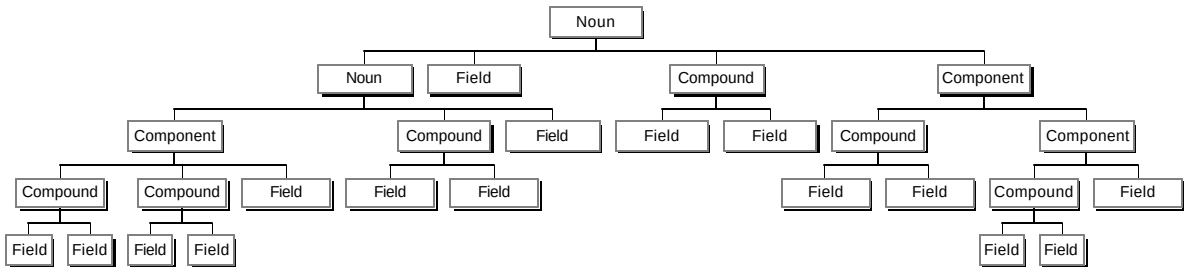
In order to take advantage of the flattened BODS - When creating your XML, make sure that the BOD location points to the flattened BOD. Example:

xsi:schemaLocation="../BODs/**Standalone**/AcknowledgeCreditContractResponse.xsd" revision="2.0" release="8.1" environment="Production" lang="en-US">

6.5.7.1 STAR Development Architecture Example

```
<xs:schema targetNamespace="http://www.starstandards.org/STAR" xmlns="http://www.starstandards.org/STAR"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:oa="http://www.openapplications.org/oagis"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:import namespace="http://www.openapplications.org/oagis" schemaLocation="../Resources/Verbs/Process.xsd"/>
    <xs:include schemaLocation="../Resources/Nouns/CreditApplication.xsd"/>                    } Resources
    <xs:element name="ProcessCreditApplication" type="ProcessCreditApplication"/>
    <xs:complexType name="ProcessCreditApplication">                                           } Which Content
        <xs:complexContent>
            <xs:extension base="BusinessObjectDocument">
                <xs:sequence>
                    <xs:element name="DataArea" type="ProcessCreditApplicationDataArea"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="ProcessCreditApplicationDataArea">
        <xs:sequence>
            <xs:element ref="oa:Process"/>
            <xs:element ref="CreditApplication" maxOccurs="unbounded"/>                        } Includes
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

6.5.7.2 Flattened BOD Assembly Example



6.5.8 Application Area

The Application Area provides the information that an application may need to know in order to communicate in an integration of two or more business applications. The Application Area is used at the application layer of communication. While the integration framework's web services and middleware provide the communication layer on top of which STAR operates.

The Application Area is a common component that is used in all STAR Business Object Documents (BOD). Therefore, the information is generic in nature and does not relate specifically to any one BOD. For this reason, very rarely are fields added to the Application Area.

There are five sections of information contained in the Application Area:

- **Sender**
- **CreationDateTime**
- **Signature**

- **BODId**
- **Destination**

6.5.8.1 Sender

The Sender component is a required component that identifies characteristics and control identifiers that relate to the application that created the BOD. The sender area can indicate the logical location of the application and/or database server, the application, and the task that was processing to create the BOD. The following identifies the type of information contained in the Sender component:

- **LogicalId –** This optional field provides the logical location of the server and applications from which the BOD originated. This would be the system or instance name of the server or application that is sending the BOD. For example, "SAP" or "Oracle Apps at Corporate Headquarters". It can be used to establish a logical to physical mapping, however its use is optional.

  Each system or combination of systems should maintain an external central reference table containing the logical names or logical addresses of the application systems in the integration configuration. This enables the logical names to be mapped to the physical network addresses of the resources needed on the network.

  Note: The technical implementation of this Domain Naming Service is not dictated by this specification.

  This logical to physical mapping may be done at execution time by the application itself or by a middleware transport mechanism, depending on the integration architecture used.

  This provides for a simple but effective directory access capability while maintaining application independence from the physical location of those resources on the network

- **Component –** This **required** field provides the DCS software code name. For example, **"Elite 2.0"**. This provides a finer level of control than Logical Identifier and represents the business application that issued the Business Object Document.

- **Task -** This **required** field describes the business event that initiated the need for the BOD to be created. Although the Task may differ depending on the specific implementation, it is important to enable drill back capability. Example Tasks may be Parts Order, Repair Order, etc.

- **ReferenceId -** This optional field enables the sending application to indicate the instance identifier of the event or task that caused the BOD to be created.

  The Sender of the originating application will populate the Reference Id with business level information.

  Reference ID does not have to be a GUID. It must be a locally unique application value for the transaction type, for example a database sequence number.

  The Reference Id will be an Optional field.

  The receiving application puts the value in Reference ID of the incoming message in the Reference ID of any acknowledgement messages.

  The Reference Id will not be required to tie two collaborations together such as Parts Order and Parts Shipment.

- **AuthorizatonId -** This optional field identifies the authorization level of the user or application that is sending the Business Object Document Message. For example, a user may only have read privileges or they may have administrator privileges. This authorization level being recognized on the receiving system indicates what can be done on the receiving system(s). An example of an AuthorizationId could be "jdoe", that being the user's ID to enter the application. (i.e., Your Id or System ID that relates to system you are integrating with)

- **CreatorNameCode –** This **required** field provides the DCS software creator code. This would be the name or code of the DCS software creator. An example of this is could be **"AD"** for ADP.

- **SenderNameCode -** This **required** field provides additional information about the sending application. There is a list STAR standard Short Manufacturer and Short DSP codes that should be used to populate this field. This list is defined in the Fields.xsd file in the STAR schema repository. It is also published yearly on the www.starstandard.org site. If a company does not have a standard code either a request should be submitted to the STAR organization or the company should use **"XX"**.

- **SenderURI -** URI is short for Uniform Resource Identifier, the generic term for all types of names and addresses that refer to objects on the World Wide Web. A URL is one kind of URI. In this case, this optional field is the physical address of the sender. An example or a SenderURI would be http://www.math.uio.no/faq/compression-faq/part1.html

- **DealerNumber -** This optional field is the Dealer Code of source of information. The Dealer Code is a manufacturer-assigned code given to all dealers. An example of a DealerNumber would be **"206501"**.

- **StoreNumber -** This optional field is the Dealer code store number. The Store Number is a DMS-assigned number. An example would be **"124"**.

- **AreaNumber -** This optional field is the Dealer code area number. The Area Number is a DMS vendor-assigned number. An example would be **"014"**.

- **DealerCountry –** This optional field describes the country location of the Dealer. There is an enumerated list of valid codes for "Country" in the DataTypes.xsd file in the STAR schema Repository

- **Language -** This optional code is used to define the language of the data used in this transaction. There is an enumerated list of valid language codes for "Language" in the DataTypes.xsd file in the STAR schema Repository

- **DeliverPendingMailInd -** This optional field indicates if the user requests to receive pending mail that has been stored and has yet not been delivered yet. By selecting 0, the user will only receive the response for the current transaction the user is performing. By selecting 1, the user will receive all pending mail.

- **Password -** This optional field is the token for application specific authentication. This password is used to authenticate dealership/users through application specific security.

- **SystemVersion -** This optional field is the version number of the sender's system.

- **PartyId -** This optional field uniquely identifies the Sender of the message. This element can be used for parties within the Automotive Community as well as external parties. Party Id is not intended as a replacement for the Dealer Number. Suggested formats for OEMs or other large institutions include: DUNs Number, ShortMfgCode + DUNs, or ShortMfgCode. The suggested format for Dealers is: ShortMfgCode+Dealer Number.

- **LocationId -** This optional field uniquely identifies the location of the Sender of a message. This Id may be aligned with a physical address or data centers. This field provides an additional level of granularity beyond the usage of the Party Id for additional routing and deliver of data.

- **ServiceId -** This optional field identifies the particular service to which a message is being sent from, e.g., an inventory service.

6.5.8.2 CreationDateTime

The CreationDateTime field is a required field that provides the date time stamp that the given instance of the Business Object Document was created. This date must not be modified during the life of the Business Object Document. Date and time conforms to ISO 8601format rules without offset EX: 2003-11-05T13:15:30Z

6.5.8.3 Signature

Signature refers to a digital signature. A digital signature is a digital code that can be attached to an electronically transmitted message that uniquely identifies the sender. Like a written signature, the purpose of a digital signature is to guarantee that the individual sending the message really is who he or she claims to be.

If the BOD is to be signed the signature element is included, otherwise it is not. The Signature supports any digital signature that maybe used by an implementation of STAR/OAGIS. The qualifyingAgency identifies the agency that provided the format for the signature.

This element supports any digital signature specification that is available today and in the future. This is accomplished by not actually defining the content but by allowing the implementation to specify the digital signature to be used via an external XML Schema namespace declaration. The Signature element is defined to have any content from any other namespace.

This allows the user to carry a digital signature in the xml instance of a BOD. The choice of which digital signature to use is left up to the user and their integration needs.

6.5.8.4 BODId

The BODId is an optional component that provides a place to carry a Globally Unique Identifier (GUID) that will make each Business Object Document instance uniquely identifiable. (i.e., If you have ten instances of a BOD, like on the same Parts Order, you could have the same Document Id and different BODIds)

This is a critical success factor to enable software developers to use the Globally Unique Identifier (GUID) to build the following services or capabilities:

1. Legally binding transactions,

2. Transaction logging,

3. Exception handling,
4. Re-sending,
5. Reporting,
6. Confirmations,
7. Security.

6.5.8.4.1 Best Practice

As a STAR recommended Best Practice, a unique BODId should be created every time you generate a BOD.

6.5.8.5 Destination

The Destination is a required component that identifies characteristics and control identifiers that relate to the application that **receives** the Business Object Document. The following identifies the type of information contained in the Destination component:

- **DestinationNameCode –** This **required** field provides the code for destination of file. There is a list STAR standard Short Manufacturer and Short DSP codes that should be used to populate this field. This list is defined in the Fields.xsd file in the STAR schema repository. It is also published yearly on the www.starstandard.org site. If a company does not have a standard code either a request should be submitted to the STAR organization or the company should use **"XX"**.

- **DestinationURI -** URI is short for Uniform Resource Identifier, the generic term for all types of names and addresses that refer to objects on the World Wide Web. A URL is one kind of URI. In this case, this optional field is the DestinationURI is the physical address of the destination.

- **DestinationSoftwareCode -** This optional field provides additional information about the destination application. This information may not be known. An example of DestinationSoftwareCode would be **"DD2.1"**.

- **DestinationSoftware –** This optional field is the software for which the software destination files are intended. This information may not be known. An example of DestinationSoftware would be **"DCS"**.

- **DealerNumber -** This optional field is the Dealer Code of receipt of information. The Dealer Code is a manufacturer-assigned code given to all dealers. An example of a DealerNumber would be **"206501"**.

- **StoreNumber -** This optional field is the dealer code store number. The Store Number is a DMS-assigned number. An example would be **"124"**.

- **AreaNumber -** This optional field is the dealer code area number. The Area Number is a DMS vendor-assigned number. An example would be **"014"**.

- **DealerCountry –** This optional field describes the country location of the Dealer. There is an enumerated list of valid codes for "Country" in the DataTypes.xsd file in the STAR schema Repository

- **PartyId -** This optional field uniquely identifies the Receiver of the message. This element can be used for parties within the Automotive Community as well as external parties. Party Id is not intended as a replacement for the Dealer Number. Suggested formats for OEMs or other large institutions include: DUNs Number, ShortMfgCode + DUNs, or ShortMfgCode. The suggested format for Dealers is: ShortMfgCode+Dealer Number.

- **LocationId -** This optional field uniquely identifies the location of the Receiver of a message. This Id may be aligned with a physical address or data centers. This field provides an additional level of granularity beyond the usage of the Party Id for additional routing and deliver of data.

- **ServiceId -** This optional field identifies the particular service to which a message is being sent to, e.g., an inventory service.

6.6 Verb Usage & Process Flows

Nouns identify the business specific data that is being communicated (i.e., PartOrder, SalesLead, VehicleServiceHistory, etc.). Noun is the object or document that is being acted upon. The Verb identities the action being performed on a Noun. One Noun may be used with one or many verbs.

The associated verb process flow diagrams are the standard default process flows when implementing STAR BODs. STAR recognizes that each implementation of a STAR BOD will be unique depending on the chosen transport, chosen software, etc. As a result, the associated sequence diagrams have been developed at a high level to allow room for individual implementations.

Note: If a member chooses to implement a process flow other than those identified by STAR, that implementation would be considered custom. If a member's process flow is not covered by the STAR standard process flow, they could submit a modification request to STAR to have their process flow reviewed for possible inclusion.

6.6.1 Acknowledge

The Acknowledge verb is used to acknowledge the application receipt of a PROCESS request. This function conveys the result of the original request. An example of this is ACKNOWLEDGE PO, where a PROCESS PO has been issued and the corresponding business application acknowledges the receipt of the PO and responds with an acceptance or a counter offer.

Attributes					
Attribute Name	**Attribute Type**	**Attribute Values**	**Attribute Presence**	**Attribute Default**	**Comment**
confirm	ConfirmType	“Always” "OnChange" “Never”	Required		

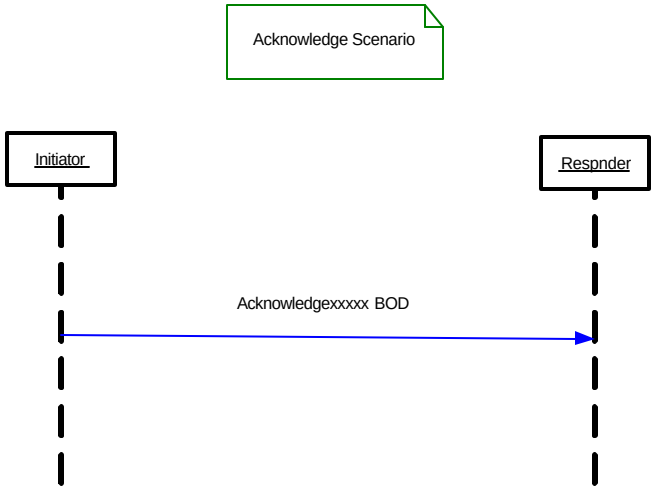
Elements					
Element Name	**Element Type**	**Element Values**	**Element Presence**	**Element Default Value**	**Comment**
OriginalBODId	xs:NMToken		Optional		Identification of BOD being acknowledged

6.6.1.1 Acknowledge Verb Attributes

STAR defines the Acknowledge verb attributes as follows

```
<Acknowledge  confirm="Never"/>
```

6.6.1.2 Acknowledge Scenario



6.6.1.2.1 Acknowledge BPSS Parameters

BPSS Parameters	
Binary Collaboration	
Parameter Name	Value
name	Acknowledgexxxxx
timetoPerform	1M
Business Transaction	
Parameter Name	Value
name	Acknowledgexxxxx
fromAuthorizedRole	initiator
toAuthorizedRole	responder
isConcurrent	true
isLegallyBinding	false
timeToPerform	30S
is GuaranteedDeliveryRequired	true
Initiator	
Parameter Name	Value
Is AuthorizationRequired	true
Initiator Document Envelope	
Parameter Name	Value
isAuthenticated	none
isConfidential	none
isTamperProof	none
Responder	
Parameter Name	Value
isAuthorizationRequired	true

6.6.2 Cancel

The CANCEL verb is used when the sender of the BOD is not the owner of the data, but is sending a request for the document to be canceled. An example is the CANCEL PO where the business implications must be calculated and a simple data processing term such as delete can not fully convey the business meaning and required processing associated with the meaning.

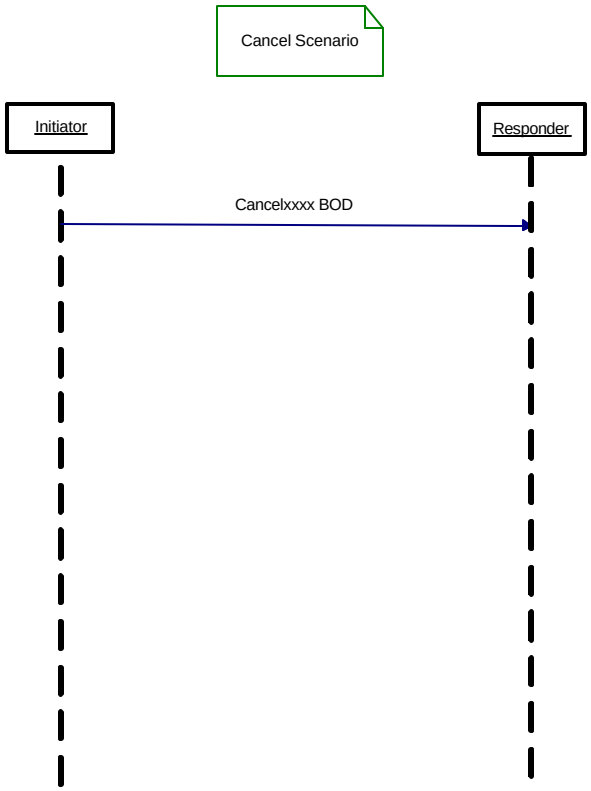
Attributes					
Attribute Name	**Attribute Type**	**Attribute Values**	**Attribute Presence**	**Attribute Default**	**Comment**
confirm	ConfirmType	“Always” "OnChange" “Never”	Required		

6.6.2.1 Cancel Verb Attributes

STAR defines the Cancel verb attributes as follows

<Cancel/>

6.6.2.2 Cancel Scenario



6.6.2.2.1 Cancel BPSS Parameters

BPSS Parameters	
Binary Collaboration	
Parameter Name	Value
name	Cancelxxxxx
timetoPerform	1M
Business Transaction	
Parameter Name	Value
name	Cancelxxxxx
fromAuthorizedRole	initiator
toAuthorizedRole	responder
isConcurrent	true
isLegallyBinding	false
timeToPerform	30S
is GuaranteedDeliveryRequired	true
Initiator	
Parameter Name	Value
Is AuthorizationRequired	true
Initiator Document Envelope	
Parameter Name	Value
isAuthenticated	none
isConfidential	none
isTamperProof	none
Responder	
Parameter Name	Value
isAuthorizationRequired	true
Responder Document Envelope	
Parameter Name	Value
isPositiveResponse	true
isAuthenticated	none
isConfidential	none
isTamperProof	none
businessDocument	ConfirmBOD

6.6.3 Change

The CHANGE verb is used when the sender of the BOD is not the owner of the data, but is sending a request for the document to be changed. An example of this is CHANGE FinancialStatement, where the original document needs to be changed based on a specific business event. The Change must refer to the original document and/or items. The change processing assumes replacement of fields sent, with the exception of the identifying fields of the Noun

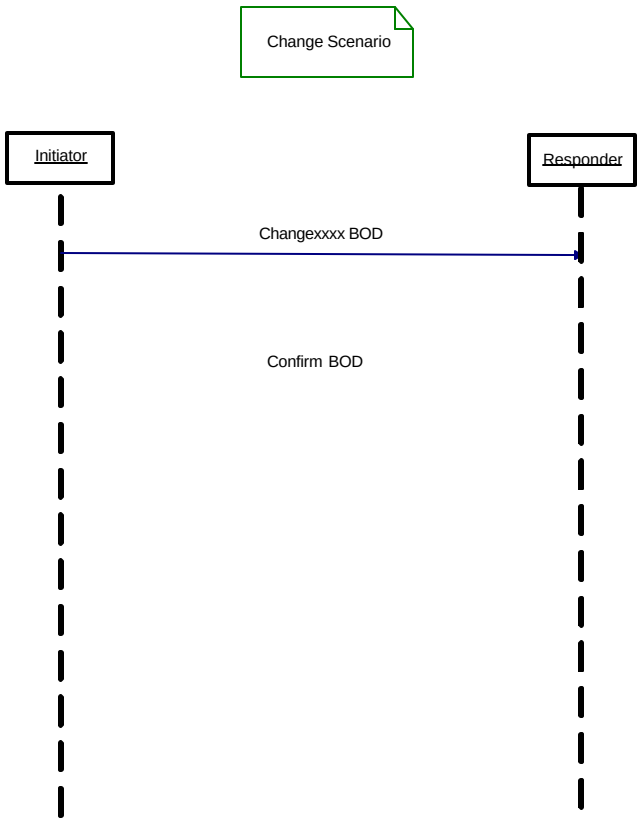
Attributes					
Attribute Name	**Attribute Type**	**Attribute Values**	**Attribute Presence**	**Attribute Default**	**Comment**
confirm	ConfirmType	“Always” "OnChange" “Never”	Required		

6.6.3.1 Change Verb Attributes

STAR defines the Change verb attributes as follows

<Change/>

6.6.3.2 Change Scenario



6.6.3.2.1 Change BPSS Parameters

BPSS Parameters	
Binary Collaboration	
Parameter Name	Value

BPSS Parameters	
name	Changexxxxx
timetoPerform	1M
Business Transaction	
Parameter Name	Value
name	Changexxxxx
fromAuthorizedRole	initiator
toAuthorizedRole	responder
isConcurrent	true
isLegallyBinding	false
timeToPerform	30S
is GuaranteedDeliveryRequired	true
Initiator	
Parameter Name	Value
Is AuthorizationRequired	true
Initiator Document Envelope	
Parameter Name	Value
isAuthenticated	none
isConfidential	none
isTamperProof	none
Responder	
Parameter Name	Value
isAuthorizationRequired	true
Responder Document Envelope	
Parameter Name	Value
isPositiveResponse	true
isAuthenticated	none
isConfidential	none
isTamperProof	none
businessDocument	ConfirmBOD

6.6.4 Confirm

The Confirm BOD is used to respond to a request to confirm receipt of information by the receiving system. The Confirm BOD indicates whether or not the original message was understood by the receiving system.

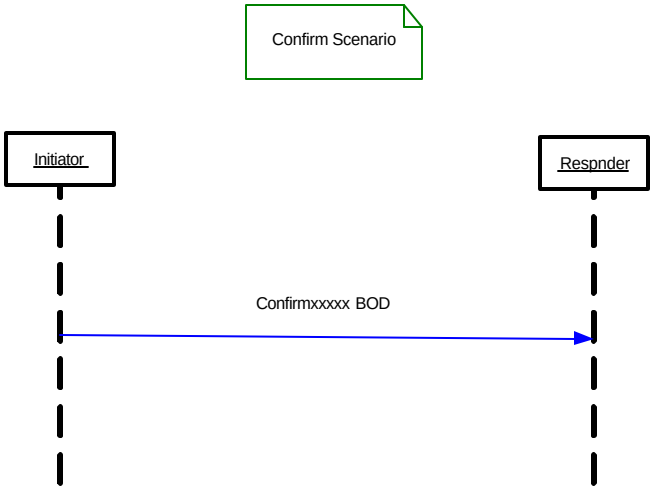
Note: Reference the Confirm BOD Implementation Guidelines for additional information on the use of the Confirm BOD

6.6.4.1 Confirm Verb Attributes

STAR defines the Confirm verb attributes as follows

<Confirm/>

6.6.4.2 Confirm Scenario



6.6.4.2.1 Confirm BPSS Parameters

BPSS Parameters	
Binary Collaboration	
Parameter Name	Value
name	ConfirmBOD
timetoPerform	1M
Business Transaction	
Parameter Name	Value
name	ConfirmBOD
fromAuthorizedRole	initiator
toAuthorizedRole	responder
isConcurrent	true
isLegallyBinding	false
timeToPerform	30S
is GuaranteedDeliveryRequired	true
Initiator	
Parameter Name	Value
Is AuthorizationRequired	true
Initiator Document Envelope	
Parameter Name	Value
isAuthenticated	none
isConfidential	none
isTamperProof	none
Responder	
Parameter Name	Value
isAuthorizationRequired	true

6.6.5 Get

The GET verb is to communicate to a business software component a request for an existing piece of information to be returned. The response to this request is the SHOW verb. The GET is designed to retrieve information by using that information's primary retrieval field, or key field. The GET verb is not used to request several documents at once.

Selection Criteria: There are two types of selection capabilities for most BOD's that use the GET verb.

1. The first selection capability is called Field-Based Selection. Within a GET-based Business Object Document, the first Data Type that occurs in a specific BOD structure is commonly used to provide the Field-Based Selection criteria. This is always defined within the specific BOD and is commonly the required fields for that specific Data type. The Field-Based Selection enables the requester to provide a value or values (in the case of multiple required Field Identifiers), in the required fields. Then the responding component uses those values to find and return the requested information to the originating business software component.
2. The second type of selection capability for GET-based BODs is called Data Type Selection. Data Type selection enables the requester to identify which Data Types within the noun are requested to be returned in the response. The use of this capability is described for each corresponding Data Type for all BODs that use the GET verb. The Data Types are identified for retrieval within the GET instance of a BOD by including the name of the Data Type in the meta data but without any Field Identifiers or Segments identified within the Data Type. This will signify to the responding application that all of the data that corresponds to that Data Type is to be included in the response. If the Data Type is not requested, the Data Type identifier is not included in the GET request and this will signify to the responding component that the Data Type is not to be returned.

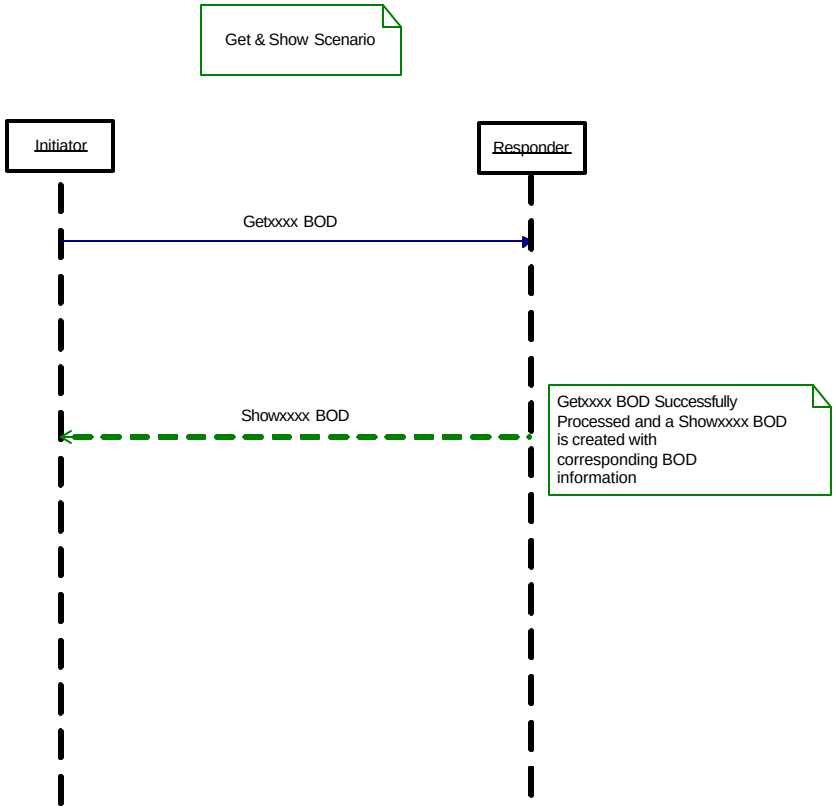
Attributes					
Attribute Name	**Attribute Type**	**Attribute Values**	**Attribute Presence**	**Attribute Default**	**Comment**
confirm	ConfirmType	“Always” "OnChange" “Never”	Required		
show	xs:string	"Always"	Required		Fixed value
ReturnCriteria	N/A	N/A	Required		STAR has defined the return criteria in the SHOW guidelines associated with each Get

6.6.5.1 Get Verb Attributes

STAR defines the Get verb attributes as follows (Reference Show verb for Show verb attributes)

```
<Get show="Always">
    <ReturnCriteria>
        <SelectExpression/>
    </ReturnCriteria>
</Get>
```

6.6.5.2 Get/Show Scenario



6.6.5.2.1 Get/Show BPSS Parameters

BPSS Parameters	
Binary Collaboration	
Parameter Name	Value
name	Getxxxxx
timetoPerform	1M
Business Transaction	
Parameter Name	Value

BPSS Parameters	
name	Getxxxxx
fromAuthorizedRole	initiator
toAuthorizedRole	responder
isConcurrent	true
isLegallyBinding	false
timeToPerform	30S
is GuaranteedDeliveryRequired	true
Initiator	
Parameter Name	Value
Is AuthorizationRequired	true
Initiator Document Envelope	
Parameter Name	Value
isAuthenticated	none
isConfidential	none
isTamperProof	none
Responder	
Parameter Name	Value
isAuthorizationRequired	true
Responder Document Envelope	
Parameter Name	Value
isPositiveResponse	true
isAuthenticated	none
isConfidential	none
isTamperProof	none
businessDocument	ConfirmBOD (If error) Showxxxxx BOD

6.6.6 Process

The Process verb is used to request processing of the associated noun by the receiving application or business to party. In a typical external exchange scenario a Process BOD is considered to be a legally binding message. For example, if a customer sends a ProcessPartsOrder BOD to a supplier and the supplier acknowledges with a positive AcknowledgePartsOrder, then the customer is obligated to fulfill the agreement, unless of course other BODs are allowed to cancel or change the original order

Attributes					
Attribute Name	**Attribute Type**	**Attribute Values**	**Attribute Presence**	**Attribute Default**	**Comment**
confirm	ConfirmType	“Always” "OnChange" “Never”	Optional		
acknowledge	AcknowledgementType	“Always” "OnChange" “Never”	Optional		

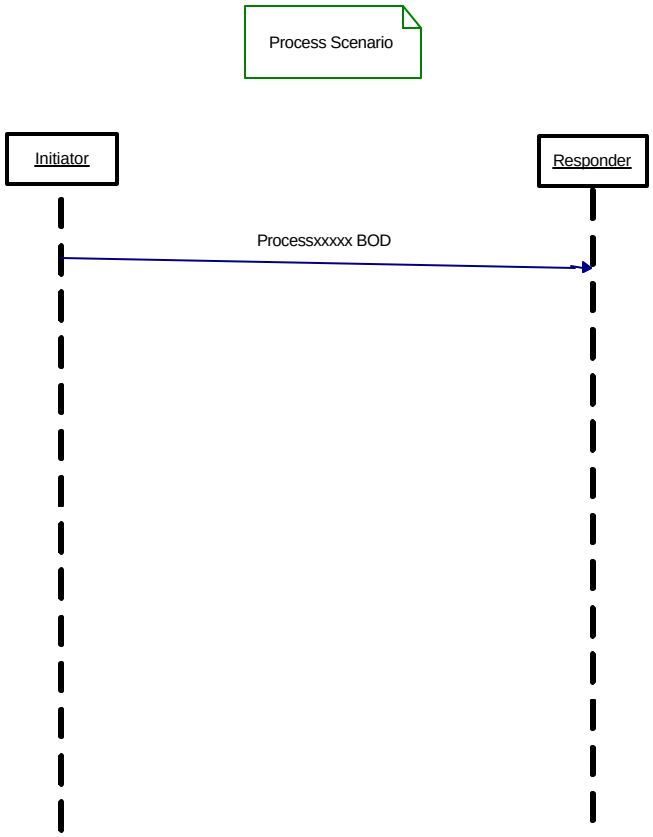
Attributes					
Attribute Name	**Attribute Type**	**Attribute Values**	**Attribute Presence**	**Attribute Default**	**Comment**
Criteria	N/A	N/A	N/A		STAR does not use - This is to define an Xpath expression to be used with the BOD.

6.6.6.1 Process Verb Attributes

STAR defines the Process verb attributes as follows

```
<Process  acknowledge="Never"/>
```

6.6.6.2 Process Scenario



6.6.6.2.1 Process BPSS Parameters

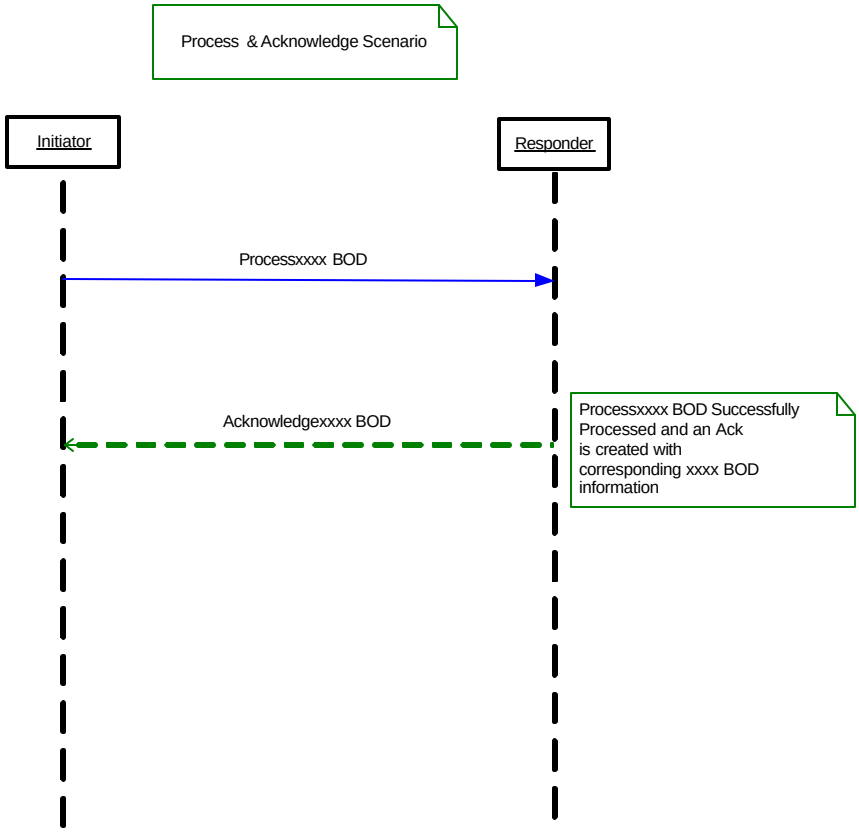
BPSS Parameters
Binary Collaboration

BPSS Parameters	
Parameter Name	Value
name	Processxxxxx
timetoPerform	1M
Business Transaction	
Parameter Name	Value
name	Processxxxxx
fromAuthorizedRole	initiator
toAuthorizedRole	responder
isConcurrent	true
isLegallyBinding	false
timeToPerform	30S
is GuaranteedDeliveryRequired	true
Initiator	
Parameter Name	Value
Is AuthorizationRequired	true
timeToAcknowledgeReceipt	10S
Initiator Document Envelope	
Parameter Name	Value
isAuthenticated	none
isConfidential	none
isTamperProof	none
Responder	
Parameter Name	Value
isAuthorizationRequired	true
Responder Document Envelope	
Parameter Name	Value
isPositiveResponse	true
isAuthenticated	none
isConfidential	none
isTamperProof	none
businessDocument	ConfirmBOD

6.6.6.3 Process/Acknowledge Scenario

STAR defines the Process verb attributes as follows. (Reference Acknowledge verb for Acknowledge verb attributes)

```
<Process acknowledge="Always"/>
```



6.6.6.3.1 Process/Acknowledge BPSS Parameters

BPSS Parameters	
Binary Collaboration	
Parameter Name	Value
name	Processxxxxx
timetoPerform	1M
Business Transaction	
Parameter Name	Value
name	Processxxxxx
fromAuthorizedRole	initiator
toAuthorizedRole	responder
isConcurrent	true
isLegallyBinding	false
timeToPerform	30S
is GuaranteedDeliveryRequired	true
Initiator	
Parameter Name	Value
Is AuthorizationRequired	true
Initiator Document Envelope	
Parameter Name	Value
isAuthenticated	none
isConfidential	none
isTamperProof	none

BPSS Parameters	
Responder	
Parameter Name	Value
isAuthorizationRequired	true
Responder Document Envelope	
Parameter Name	Value
isPositiveResponse	true
isAuthenticated	none
isConfidential	none
isTamperProof	none
businessDocument	ConfirmBOD (If error) Acknowledgexxxxx BOD

6.6.7 Show

The Show verb is used when sending the information about a specific instance of a business document or entity. The Show verb may be used to respond to a Get request or it can be used in a publish scenario, where it pushes information to other applications based on a business event. Although BODs based on this verb do not commonly cause updates to occur, there may be times when the component receiving the Show decides to use the information it receives to update. This is entirely the decision of the receiving software component and is not forbidden. The behavior of the Show verb is quite straight forward with one exception. The Show response to any Get request needs to read the request carefully to ensure the response is returning the requested Data Types.

Attributes					
Attribute Name	**Attribute Type**	**Attribute Values**	**Attribute Presence**	**Attribute Default**	**Comment**
confirm	ConfirmType	“Always” "OnChange" “Never”	Required		

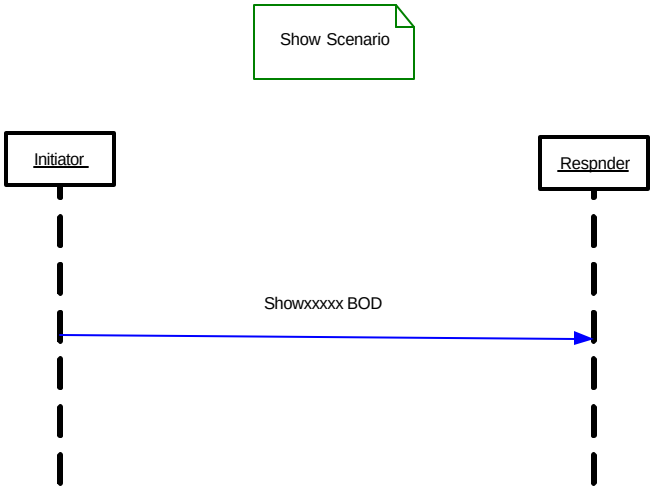
Elements					
Element Name	**Element Type**	**Element Values**	**Element Presence**	**Element Default Value**	**Comment**
OriginalBODId	xs:NMToken		Optional		Identification of BOD that made Show request (“Get”)

6.6.7.1 Show Verb Attributes

STAR defines the Show verb attributes as follows

<Show/>

6.6.7.2 Show Scenario

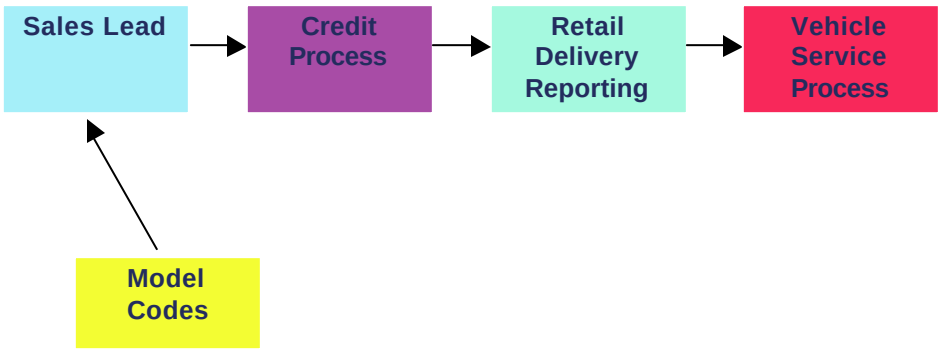


6.6.7.2.1 Show BPSS Parameters

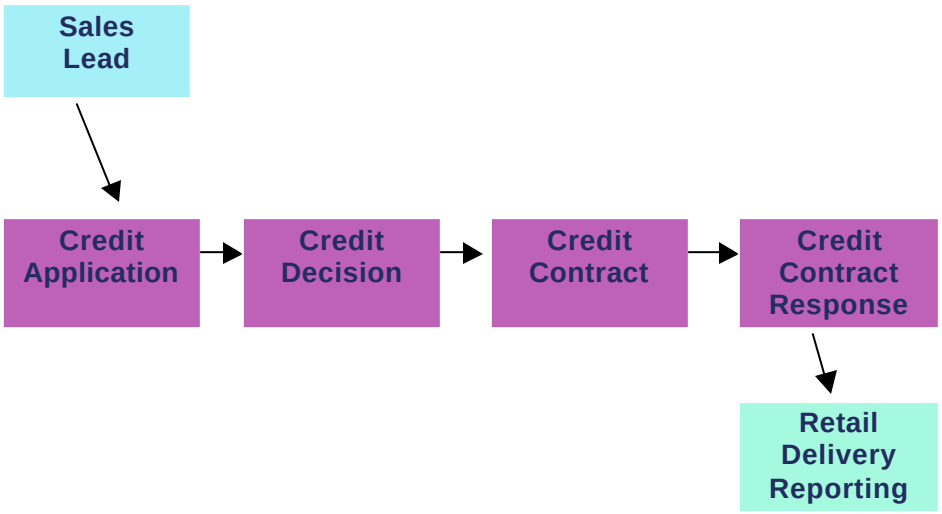
BPSS Parameters	
Binary Collaboration	
Parameter Name	Value
name	Showxxxxx
timetoPerform	1M
Business Transaction	
Parameter Name	Value
name	Showxxxxx
fromAuthorizedRole	initiator
toAuthorizedRole	responder
isConcurrent	true
isLegallyBinding	false
timeToPerform	30S
is GuaranteedDeliveryRequired	true
Initiator	
Parameter Name	Value
Is AuthorizationRequired	true
Initiator Document Envelope	
Parameter Name	Value
isAuthenticated	none
isConfidential	none
isTamperProof	none
Responder	
Parameter Name	Value
isAuthorizationRequired	true

6.7 Sample Business Flows

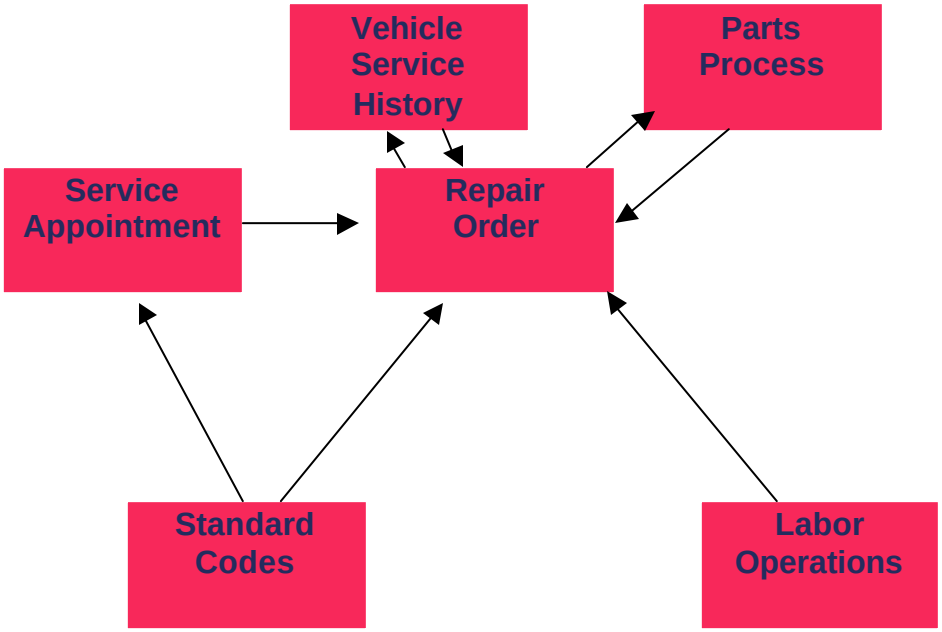
6.7.1 Customer Service



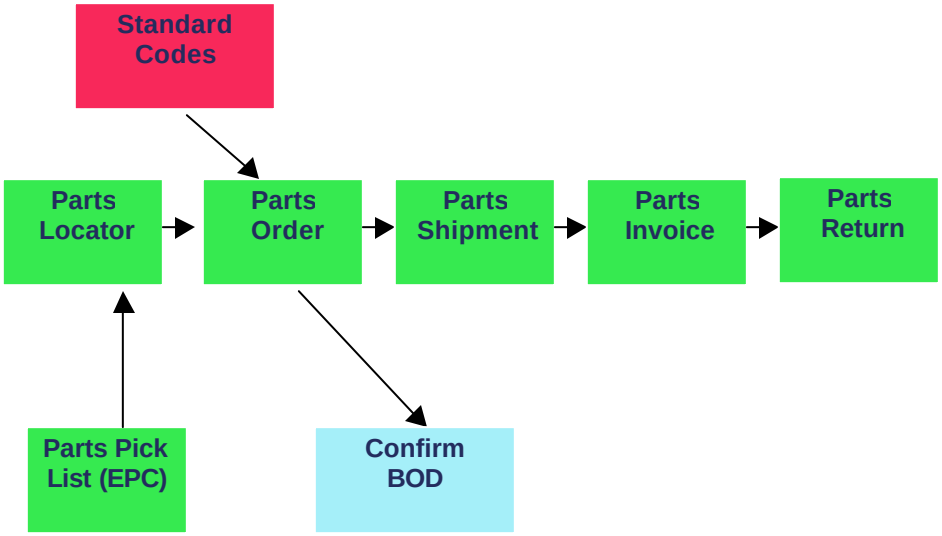
6.7.2 Credit Process



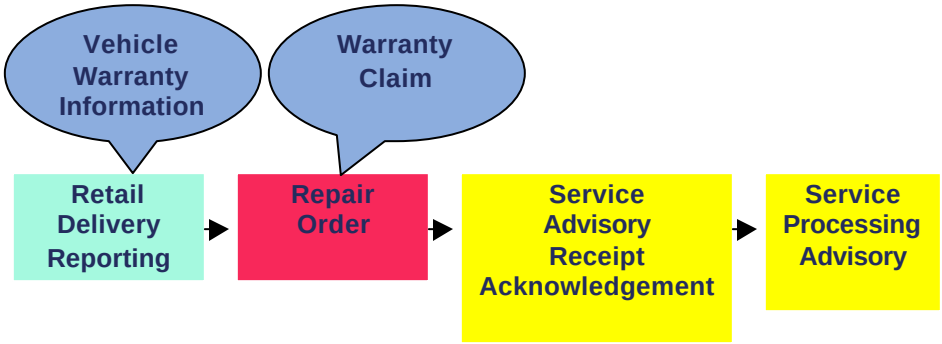
6.7.3 Vehicle Service



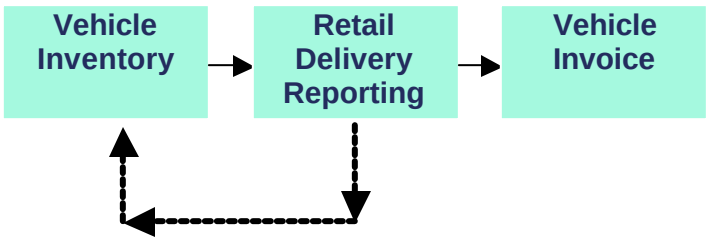
6.7.4 Parts Process



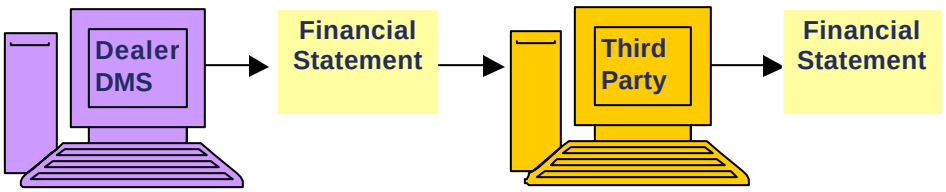
6.7.5 Warranty



6.7.6 Vehicle Inventory



6.7.7 Financial Report



6.8 Best Practices Recommendations from the Transport Guidelines

6.8.1 Reliability

Reliability assumptions are to be delivered out of the transport layer, not at the application layer.

6.8.2 Auditing Best Practices

When a BOD or other payload is ready to be sent through a transport medium, a method for relating the payload to the message that carries it and tracking that relationship is good practice. As a result, metadata exists to establish a trail of non-repudiation or at least a log that can be used to support business relationships. For more information on implementing such methods please refer to the STAR Transport Guidelines, section on Auditing Best Practices.

6.8.3 Size of BODs and Performance

When sending large BODs between business partners, it may be necessary to pre-process the BODs before sending them to address performance restrictions. Conversely, there may be a need to pre-process message packages from a business partner before they can processed in local applications. For more information regarding these types of methods for STAR payloads please refer to the STAR Transport Guidelines, section on Performance Discussions.

In general very large BODs, grouping multiple transactions or BODs into one transmission, or other extremely large payload can tie up a communications channel for an extended amount of time. When this becomes a problem, one natural approach to address it is to break large transfers into a series of smaller messages. For more information on this technique, refer to the STAR Transport Guidelines, section Collaboration, Large Message Handling.

6.8.4 Coordinating Transactions

Coordinating several transactions between business partners through the STAR transport layer to create a business “conversation” may be important. Business applications are responsible for coordinating the individual transactions into a choreographed session that achieves a business objective. The transportation layer should be provided key data items and it should be able to provide key data items to co-relate the transactions and conversations to the individual messages that are sent. For more information please refer to the STAR Transport Guidelines, section Collaboration, Best Practices.

7 Appendix

7.1 Acronyms

Acronym	Definition
BOD	Business Object Document
BPSS	Business Process Specification Schema
DMS	Dealer Management System
DSP	Dealer System Provider
DTD	Document Type Definition
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDE	Interactive Development Environment
OAG	Open Applications Group
OAGI	Open Application Group Inc.
OAGIS	Open Applications Group Integration Specification
RSP	Retail System Provider
SOAP	Simple Object Access Protocol
STAR	Standard for Technology in Automotive Retail
WWW	World Wide Web
XDR	External Data Representation Standard
XSL	Extensible Stylesheet Language
XML	Extensible Markup Language

7.2 Glossary

Term	STAR Definition
Application Program	A computer program that operates on a target platform(s) and performs a specific user function.
Business Object Document (BOD)	A Business Object Document (BOD) is the model used to communicate a request from the originating business application to the destination business application. Each BOD includes supporting details to enable the destination application to accomplish the action.
ebXML	A set of specifications that together enable a modular, yet complete electronic business framework. If the Internet is the information highway for electronic business, then ebXML can be thought of as providing the on ramps, off ramps, and the rules of the road. The ebXML architecture provides: • A way to define business processes and their associated messages and content. • A way to register and discover business process sequences with related message exchanges. • A way to define company profiles. • A way to define trading partner agreements. • A uniform message transport layer.
HTML	A markup language used to format data for presentation.
Interactive Development Environment tool (IDE)	Tool to assist in the development of XML. While STAR does not recommend any particular products, two that you might look at are XML Spy http://www.xmlspy.com or Tibco http://www.tibco.com
Infrastructure	The underlying foundation, or basic framework, as of a system.

Term	STAR Definition
Open Applications Group INC. (OAGI)	The Open Applications Group, Inc. (OAGI) is a non-profit industry consortium comprised of many of the most prominent stakeholders in the business software component industry throughout the world. This group foresaw the rapidly growing need to integrate business applications much more quickly and cost effectively. The Open Applications Group does this by continuous development of a practical and implementable best practices model for business software interoperability while providing an impartial forum for industry stakeholders to learn, cooperate, and to further improve the model.
System	A regularly interacting or interdependent group of components forming a unified whole, under the influence of a set of rules, conditions, and constraints that perform one or more vital functions.
Tags	Words bracketed by "<" and ">". XML used Tags only to delimit pieces of data.
World-Wide Web (WWW)	(WWW, W3, The Web) An Internet client-server distributed information retrieval system originated from the CERN High-Energy Physics laboratories in Geneva, Switzerland. The WWW represents objects in HTML format with hyperlinks referring to other documents by their URLs. The World Wide Web Consortium is the main standards body for the web.
XDR	A standard for the description and encoding of data. It is useful for transferring data between different computer architectures.
XML	A text-based mark-up language for structuring data content into a self-describing wrapper.
XSL	A procedural language used for document rendering, document querying and data transformations.

7.3 STAR Field Data Type Definitions

User defined data types allow for data type inheritance. STAR has defined several unique data types for inheritance. This document lists the common data types that are used by STAR for inheritance in the definition of fields.

Type	Definition	Attribute	Attribute Type (Valid Values)	Required/ Optional	Comment
AccountingDate	xsgYearMonth				Conforms to ISO 8601 Example: 2002-03
AddressLine	xs:string				
Amount	xs:decimal	currency	Currency	required	
City	xs:string				
Code	xs:string				
CompleteIncomplete	xs:string				Enumerated Values of “C” and “I”
Currency	xs:string				See schema for enumerated list of valid currency codes
Count	xs:integer				
County	xs:string				
Country	xs:string		Country		Reference enumerated Country list in schema
Date	xs:date				Conforms to ISO 8601 Example: 2002-03-11
DateTime	xs:dateTime				Must be formatted as XML Schema Datetimes in UTC/GMT format without offsets EX:2003-11-05T13:15:30Z
Department	xs:string				
Description	xs:string				
EMailAddress	xs:string	desc		Optional	Business rule that it should conform to the Internet message format as set forth by RFC 822
Encoding	xs:string				Enumerated Values of “base64”, “octal”, “binary” “plan Text”
Hours	xs:decimal				
Id	xs:string				
Indicator	xs:string				Enumerated Values of “0” and “1”
Income	xs:decimal	currency	Currency	required	
		period	IncomePeriod	required	

Type	Definition	Attribute	Attribute Type (Valid Values)	Required/ Optional	Comment
ItemId	Id				
LaborRate	Amount				
Language	xs:string		Language		Reference enumerated Language list in schema
LocationDescription	xs:string				
Make	xs:string				
Measurement	xs:decimal				
Mileage	Count	uom	MileageMeasure	optional	
Name	xs:string				
Note	xs:string				
Percent	xs:decimal				
Priority	xs:string				
Quantity	xs:decimal	uom	UOM	required	
Ratio	xs:decimal				
ReferenceNumber	xs:string				
RentLoan	xs:string				Enumerated values of “L” and “R”
SerialNumber	xs:string				
StateOrProvince	xs:string				
Terms	xs:string				
Text	xs:string				
Type	xs:string				
URI	xs:anyURI				
VIN	xs:string				
Volume	xs:decimal	uom	VolumeMeasure	required	
Weight	xs:decimal	uom	WeightMeasure	required	
Year	xs:gYear				

7.4 Frequently Asked Questions

Reference Number	Question	Response
7.4.1	**Change Control & Modification Requests**	
1	The BOD specification looks fine except it doesn't account for a field we need. What do I do to get that Field added?	Contact the STAR XML Architect to verify that no other OEM or RSP has already requested addition of this field to the BOD. To make this request, complete the *STAR BOD Modification Request Form* for the new field and submit to one of the STAR XML Architects.
2	We need a completely new BOD supported, what do I do?	Contact the STAR XML Architect to verify that no other OEM or RSP has already requested or is in the process of defining the same new BOD. Complete the *STAR Value Statement for New BOD Development* for the new BOD and submit to the STAR XML Architect.
3	We would like to make a request to delete and/or combine existing components, what is the process to do that?	Historically STAR has taken the position of not making changes that would not allow a BOD to be backward compatible. By eliminating or combining existing components that are used in approved BODs, it would no longer be backward compatible. We would only consider making such changes if a member were to have trouble implementing a BOD or if a member were to have compelling business reasons.
7.4.2	**Collaborations & Protocol**	
1	Why has STAR split each transaction into a separate collaboration?	The reason for this is because not all transactions are required and not all transactions are always used. The majority of the time, STAR sends a transaction and then it is complete. The exception to this would be a GET/SHOW. If there is an update it would be another binary collaboration. If all transactions were in one collaboration, the collaboration could be open for an extended period of time.
2	Has STAR defined CPAs?	STAR does not create the CPAs for BODs.
3	If we use BPSS and ebXML, can we just send the noun in the payload?	It is true that if you use BPSS, you can represent the verb within it and some of the routing information is also in ebXML. However, the BOD payload should include the total BOD schema (Application Area, Verb and Noun). Although some of the information may be duplicated, it is necessary to send all this information in the payload so that organizations know how to process the BOD information when the framework is not there. (i.e., sending internally from one part of an organization to another part or a different application)

Reference Number	Question	Response
4	What is the relationship between STAR BODs and SOAP messages?	The entire STAROAGI BOD (Application Area, Verb and Noun) would be contained in the body of the SOAP message. Although some of the information may be duplicated, it is necessary to send all this information in the body of the SOAP message so that organizations know how to process the BOD information when the framework is not there. (i.e., sending internally from one part of an organization to another part or a different application)
7.4.3	**Documentation**	
1	Why are all the documents in PDF file format on the starstandard.org web site?	It is STAR standard procedure to put all STAR approved documents in PDF format, other than the schema. This is done to ensure the integrity of the standard documents.
2	Why isn't there a list of valid values for the SenderNameCode and DestinationNameCode in the documentation?	The Implementation Guideline document says "Reference ShortMfg enumerator". Since this list is constantly changing and is also very large, it is impractical to include it in every BOD Implementation Guideline. The current list of valid Short Manufacturer Codes (ShortMfg) is available on the STAR web site. www.starstandard.org
3	Where can I find out information about the STAR Reference Number?	The STAR Reference Number column in the Implementation Guidelines is used to reference the DTS equivalent field, if there is one. This is helpful when our members are converting from DTS to XML. Reference **Columns in the Component Tables** in this document.
7.4.4	**Fields & Formats**	
1	Do all fields have Valid Values?	Where possible, STAR has standardized on code definitions. Valid values are shown in the Valid Value column in the STAR Implementation Guidelines and defined in the Fields.xsd File in the schema. When code values are based on setup criteria within a dealership, field valid values have been defined as "Pass Through". This indicates that whatever the dealer enters or selects in a DMS will be passed through to the OEM.
2	Would I still be conforming to the standard if I restricted the values of a pass thru field?	If you restrict your values at the application level, it is not Non-standard. However, if you ask the RSP to validate using your valid value that would be a modification prior to sending the data and the RSP could consider it Non-standard.
3	What is the format for the date?	Dates should be YYYY-MM-DD
4	DTS has defined field lengths, why has this not been carried forward to the XML?	For schema, STAR is following the OAGI standard approach and defining the type, but not fixing the length of the field. The receiving XML parser

Reference Number	Question	Response
		should be designed to read the data between the opening and closing tag of each field. For example, STAR has used data types such as decimal, date, note, code etc.
5	Names are typically in the format of Last name, First name. Is this acceptable?	Format for full names should be "FirstName LastName"
6	Why is the same field sometimes used on both a header and detail record?	STAR often use the same field at both a header level and detail level. If it is at the header level, it would apply to the total transaction. If at another level, it would just apply to that record and its descendants.
7	Do all the fields in the BOD Schema need to be sent every time or just the required fields for the specific transaction with the OEM?	You would not need to send all the fields defined in the STAR XML schema. You would only have to send the Fields that were required by the schema and/or the corresponding Implementation Guidelines (i.e., business rules) All the optional fields would only be sent if they contained data. This keeps the size of the transmission file to a much smaller size than if you had to send every possible field defined in the schema in each transmission file. Keep in mind that the parsers should be able to process all fields defined by the schema, but it does not all need to be sent every time.
8	Are the fields in the DTS and the XML the same for the same specification (i.e., Repair Order)?	STAR members made the decision to include all data that is in a DTS in the XML schema for forward compatibility when members are converting from DTS to XML. But the reverse is not true. The membership agreed that data that has been defined for a BOD that was not in the DTS specification would only be added to the DTS if a STAR member requested it. Also keep in mind that you would need to review the corresponding Implementation Guidelines for "Inactive" fields.
9	If a field is defined in a STAR BOD, does that mean that the RSP or OEM has to populate it?	The organization creating and sending a BOD can only send the information that they have available in their application. If the organization receiving the BOD requests to have a field populated that the sender does not have available in their application, this could be considered a Non-Standard request. It would be Non-Standard because the application to create the BOD information would need to be modified.
10	How do I delete one piece of data that was previously sent?	You would use the "Change" verb and send either "Null" information or some other data for the tag value that your application would recognize as an indication to delete the previous information sent.
11	Do any of the STAR BODs allow a blob attachment to be sent over the ebMS? If yes, does this/these BOD(s) have a reference field(s) to the blob. I think there is a business case where a party (dealer or OEM) may want to send a blob such as Word Docs, JPEG file, etc. along with the	At this time, only a few of the BODs allow a generic blob attachment using the OAGI "Attachment" component that allows attachment of a binary file. This requires converting the data to binary and then attaching it to the BOD. This also requires specifying what the encoding for the data would be (Usually Binary Base 64). If you have a specific need for a file to be

Reference Number	Question	Response
	BODs to the trading partner.	associated with a BOD, you would need to make a request using the STAR Modification Request form that we add the "Attachment" component to the BOD that requires this capability. Remember, even if you want to attach the file in a SOAP message, you would still need something within the BOD to notify that there is a file attachment.
12	Can the STAR data architects perform data mapping of STAR BODs to STAR members' backend systems?	Unfortunately because STAR is not familiar with our members' individual data dictionaries and back-end systems we are not qualified to do any detailed analysis of members' data mappings. Instead we try as much as possible to make very detailed the descriptions in our implementation guidelines. However, if you have specific questions about the STAR schema or the implementation guidelines, we can certainly help you there.
13	Who decides what should go there if the OEM does not need them, does not use them, and the DMS does not have them by default, and they now need to build them for the STAR compliance?	If STAR does not have a valid value for a field, the data value in the sending system for that field would be used. If a field is required and is not in the sending system, modifications may need to be made to the system in order to send a STAR complaint BOD. You would use the Implementation Guidelines to determine the content of the field.
14	Every element that requires a date field must be populated or a schema validation tool will reject the document, but many of these elements (e.g. PromisedShipDate, PromisedDeliveryDate, etc.) are optional in the implementation guidelines, as they should be. Does this mean our dealers must populate all optional date fields with bogus dates just so they will pass the schema validation test?	No, if the field is optional, it does not need to be included in the XML data file that is being sent. You only include optional fields in the file to be transmitted if you have data to populate them with.
15	There are a number of other elements that require values to be valid document, however valid values may not exist depending on the OEM. How should we handle these?	If an enumerated value has been defined for a Required field, then the sender would need to populate the field with one of the defined enumerators. If it is a Required field without an enumerator, the sender could populate it with "N/A" for text fields, and would have to put a valid date in Required date fields.
16	What changes will occur in field formatting for claim entries, and should dealers expect to look different once STAR is implemented?	STAR standardizes on the exchange of information between two systems. Any differences that this would make in the DMS applications would be determined by the Retail System Provider of the DMS and the individual OEM and is outside of the STAR charter.
17	For several of the Amount fields, there is a note in the Comments Column that states 'Requires "Currency Attribute"'. Question: Specifically what is required and where is it defined?	The currency attribute is defined in the STAR schema. If you look at "Currency" in the Fields file, you will see a list of all the valid currencies.
18	If I am the originator of an application, and am requiring	No, this only relates to the FORMAT of the field's data not the data itself.

Reference Number	Question	Response
	specific data be supplied for a pass thru field, would the following statement mean that it is not custom? - If data is sent from an originating system in a particular format, it should be sent back to the originating system in the same format.	For example, if I send you a part number with a dash in it, I want the part number to come back w/ a dash in it.
19	Would you please provide some clarification on this statement - If there is no enumerator defined for a field, it would be defined by the sender based on the originating system.	You can as the originator send your own value but if you expect the receiver to do something with data - not just pass it thru then it could be considered Non-Standard.
20	If a field is marked 'Required' and the receiver has identified the field may be 'Optionally' populated, does this mean that the field tags must be included within the XML document (Required) but data optionally supplied. eg. <BinLocation></BinLocation> or <BinLocation>C49AB</BinLocation>	If the field is required in the standard but you have no data to pass you can send "N/A", but you cannot send a blank field
21	We are getting the error "java.io.UTFDataFormatException" when the input XML carries the value of a field custLname="Bédard", which French accent. We are using encoding="UTF-8".I was told the encoding should be changed to ISO-8xxx to make it work for French letters, is that STAR recommended. Could you please direct us on how to fix this problem	1. Our schema is UTF-8 encoded which is what it should be. The schema should not be modified. 2. In order to handle such illegal characters, you must use character entity references. For example in order to represent "<" as a data value you have to use "&lt" because "<" is an illegal character. 3. So how is this handled? With a language plug in, in this case French, most all middleware tools such as SeeBeyond, WebMethods, etc. are able to handle these character entity references. So if you have an french application that allows you to enter those special character like the accent mark over an "A", the data would then be converted in the middleware to those character entity references. With this example of the accent mark over the capital "A", the reference would be "Á". Then when it goes through the parser, the parser is able to understand the entity reference "Á". I'm assuming that you would then need something on the other end of the parser to translate it back to the original capital "A" with the accent. 4. Here is a link to the character entity references: http://www.w3.org/TR/REC-html40/sgml/entities.html#h-24.1
22	We want to use a valid value that is defined in the schema, but does not show in the Implementation	Only the valid values shown in the Implementation Guidelines are valid for a BOD. If you need to use any others, you would need to submit a

Reference Number	Question	Response
	Guidelines. Would this be STAR compliant?	modification request form to have it added to the BOD.
23	What is the rationale for the requirement to send all optional fields available to the Sender? It may unnecessary increase the development costs.	The reason is that if they have to just send the fields specific to each receiver, it would be a custom implementation for each receiver, rather than the sender only having to develop it one time. The receiver should ignore the data they do not need.
24	For the DateTime, The T and Z are not required in the ISO standard, so does that mean that it is not required in the STAR standard?	STAR follows the ISO 8601 date time format. As noted by ISO, if a date and a time are displayed on the same line, then always write the date in front of the time. If a date and a time value are stored together in a single data field, then ISO 8601 ***suggests*** that they should be separated by a latin capital letter T, as in 19951231T235959. In addition, ISO notes that you ***can*** use a capital letter Z to indicate that a time is measured in Universal Time (UTC).STAR suggests the use of these letters as a best practice for readability, however it is not a requirement.
7.4.5	**Implementation**	
1	How do companies use the STAR standards?	An example would be a series of applications for vehicle loan processing that one of our members – Route One – is developing. The applications will use STAR standards for Credit Applications and Credit Contracts.
2	When an organization implements a STAR specification, how does STAR ensure that that implementation, in fact, conforms to the specification?	STAR develops voluntary standards that can support our members' information requirements. We do not certify standards implementation, but members and non-members must abide by the terms of the STAR copyright.
3	Web-based inventory listings are a dime a dozen these days: Every dealership, newspaper, and specialty automobile Web site (e.g., AutoByTel, AutoTrader, CarPoint, etc) maintains online inventory listings. How would STAR's inventory standard help me accomplish my goal of maintaining those inventory listings?	STAR has developed Vehicle Inventory and Sales Lead standards. Your request addresses standardization of vehicle description information – make, model, bodystyle, trim level and accessories. This type of information is beyond the limited amount in the VIN prefix and developing standard codes for vehicle content is not an area STAR has addressed.
4	Are you aware of plans or projects that aim to implement a dealership management system based entirely on the STAR data structures in an open-source manner?	IT vendors supporting Dealers and OEMs probably have this in their business plans. A DMS of this type would benefit the industry.
5	How are STAR standard documents and deliverables licensed?	STAR standards and documentation have no license or royalties attached to it. Members, along with non-members, are free to download and implement the specifications.
6	What is the difference between the STAR XML-based standards and data transfer standards? Would XML not be used also for data transfer?	STAR Data Transfer specifications (DTS) and STAR XML specifications were created to accomplish the same task of communicating business data between dealers and auto manufacturers but utilize different formats and technologies to do so.

Reference Number	Question	Response
7	We are having problems parsing the BODs with our EAI tool. The reason (as far as we can tell) is the number of levels and the multiple schema calls. Do you have any suggestions?	1. Be sure that the version of the tool you are using supports XML Schema. 2. If the tool you are using does have schema support, contact the vendor's tech-support and place a trouble ticket. Often times the tools do not support multiple includes and imports appropriately. If you indicate that Tibco and other tool vendors have support for this they are usually quicker to react. 3. Regarding the depth, if you are loading these into memory the JVM may need to have the memory space increased.
8	Since BizTalk does not support XSD at this time, how can I convert XSD to XDR?	XML Spy and Turbo XML have this capability built in. The instance XML is the same it is just the Schema language used to validate the instance.
9	Why do we have to wait until January for a BOD to be implemented if it is effective in July?	To allow the RSPs and the OEMs development time prior to implementing the BOD.
10	When does BOD version n-1 take effect?	BOD version n-1 takes effect six months after effective date of a new version. (i.e., if version effective date is July, n-1 date would be January of the following year)
11	Do I have to use BPSS to send the BOD?	The BOD is framework independent. Think of a BOD as the content and the framework as the envelope of the transaction.
12	Is anyone in the STAR organization that is available to provide consultation to software companies that want to focus efforts on implementing STAR-compliant software applications?	STAR does not provide any consultation for non STAR members. We do however provide a detailed document on our Web site explaining our approach to developing schema, how the repository is structured, how to download, FAQs, etc. It is recommended that users read and understand that document before they even attempt to download the repository (i.e., STAR XML Reference/Implementation)
13	How do I now go about validating my BOD locally, before I send it to the manufacturer? Is there a way of automating that process?	STAR does not have a way to automatically validate a BOD. You could use a Parser such as Xerces or MS XML to validate your data against the STAR repository. Alternatively, you could send either your specification that you have created for the BOD or your sample XML file to the STAR data architects for review.
14	Does STAR gather benchmark data for OEMs? I'm looking to compare applications that the tier 1OEMs use on their dealer portal.	At this time, STAR does not collect the benchmark information that you are looking for. You might want to check with the individual OEMs to see if they have benchmark information that they would be willing to share with you.
15	From a technical perspective, I am struggling to cross-reference our legacy data dictionary element names into the BOD file structure. I would like to extend the BOD source files to contain references to our legacy element names.	To add an additional xsd to the Resource file as a Data Dictionary Reference would be acceptable as long as the user does not attempt to implement the standard using that file. It can only be used internally to cross referencing STAR fields to legacy application fields.

Reference Number	Question	Response
	In other words, I would like to record "in our company namespace, this element is/was named xyz" so that app developers see not only the STAR name but also our legacy element name when they open an XSD. Can you suggest a way to do this, or a better strategy altogether?	There may even be transformation tools available that do this type of mapping (i.e., Softshare or Xlipstream)
16	I read – though I cannot recall exactly where – that some STAR elements are required depending on the verb . Even in the Implementation Guidelines for a specific BOD – e.g. ProcessPartsShipment – I cannot find documentation on a particular element being optional or required for a specific verb. Where is such detail documented?	The example that you've chosen, ProcessPartsShipment, does not lend itself well because that noun has only one verb applied to it - Process. If you look at Parts Order 2.0 out on the website, you'll see that the document is broken down into verbs. The first section shows you all the requirements for the ProcessPartsOrder, the second shows you all of the requirements for the AcknowledgePartsOrder, etc.
17	I have implemented a draft version (.02) of a STAR XML BOD. In the draft version, there is a required field that I am currently not able to populate and therefore an unable to pass schema validation. However, in the published version of the same BOD (2.0) that I intend to migrate to at a future date the field is not required. Am I still compliant for the draft release even though I am unable to populate the required field?	No you would not be considered compliant. To be compliant you must validate against schema and meet all business rule requirements. This rule applies to implementations of published as well as draft versions of STAR XML BOD specifications.
18	Is it a requirement of the Star Schema, that XML files be formatted with the tabs and include whitespace?	Per the direction of OAGI, we recommend that whitespace is ignored. The application receiving the XML file should not be dependent on the whitespace. Whitespace is added for human readability.
7.4.6	**Schema Repository**	
1	We operate in UNIX environment. Do you know the equivalent for the dos command subst?	There is no equivalent to subst for UNIX. UNIX does not have the concept of a drive letter...it uses logical drives and assigns a drive access underneath a "directory." The way this would work is to define an alias STAR_HOME and assign it the location where STAR lives and all the STAR files work from the alias. You would also need to do this for the OAGIS repository and all the files that work from that alias.
2	When I changed the schema for my UNIX environment, my parser doesn't seem to understand the environment variable when I put <xs:include schemaLocation= "%STAR_HOME%/Rev1.1/Resources/Nouns/CreditDecision.xsd"/>	You shouldn't be editing the schema files....You need to mount the locations star off of the / (root). You need to get your sysadmin to do this for you. The commands for this on most UNIX's is:

Reference Number	Question	Response
		mount (where ever installed) /star Of course you need authority to do this on the machine.
3	Why are all the files in the repository read-only?	They are read-only because this is the STAR published standard and should not be changed..
4	Has there been any discussion so far about adding an effective date to the schema information?	There are effective dates in the schema. The schema revision effective date is in the Meta data file in the documentation immediately after the Revision Number. <xs:attribute name="revision" use="optional" default="1.1"> <xs:annotation> <xs:documentation source="http://www.starstandards.org">STAR Revision 1.1 STAR approved 04/09/2003; effective date 07/04/2003</xs:documentation> </xs:annotation> The effective date each Noun version is in each Noun file in the documentation immediately following the Noun name. <xs:complexType name="xxxxxxxxxxx"> <xs:annotation> <xs:documentation source="http://www.starstandards.org">STAR Version 1.0, STAR approved 10/4/2002; OAGI approved 10/17/2002; effective date 1/01/2003</xs:documentation> </xs:annotation>
5	You are currently using a default namespace declaration in the root of the XML data. In doing a little research on default namespaces, I noticed that it can be far more difficult and tedious to parse the XML data with XPath by using a default namespace. The two solutions that seem to be the most common is to either remove the default namespace or assign a prefix to it. Do you have any plans to add a prefix to the default namespace?	STAR has derived this default namespace approach directly from OAGI. Until such time as OAGI changes its approach, STAR has no plans to remove the default namespace or to add a prefix.
6	Can we inject a prefix for the default namespace when we load the XML data? We could then find the nodes using Xpath by referencing the prefix.	This depends on how you are "injecting" these prefixes? Is this only going to be on your end? Are you expecting others to provide the prefix? Are you putting it in data that you are sending to others? If it is only on your end then that's okays, but requiring others to support the prefix would not be STAR standard.

Reference Number	Question	Response
7	Does the URL in the schema point to a web location that we would reference for parsing?	The URL points to the webpage where we post our XML schema specifications. The schemaLocation attribute is only a hint for the processor to use - the process may not use the provided location at all. For example, the validator may have a local copy (as in the case of STAR) of the XML Schema that it uses, instead of loading the file specified, in order to decrease processor usage. In the case of STAR (and OAGI), we have chosen not to publish a schemaLocation for validation, instead you will need to have a local copy available.
8	Does STAR support UNICODE for the XML structure?	Yes, in the Declaration statements in the STAR Repository, we state <?xml version="1.0" **encoding="utf-8**"?>
9	Is every OEM going to need their own schema to validate the STAR data to its own OEM data that they need?	No, each OEM should not need a custom schema. They should be able to parse the XML file they receive against the STAR schema. They could also use XSL to narrow down the data received to satisfy their unique requirements. But, they will need an application to process the XML file they receive into their back end systems.
10	We have fields that we do not use, but are Required in the schema. Can we change the schema we're using to make these optional without causing other problems?	All Required fields need to be populated in the XML document that is being transported or they will not validate against the STAR schema. You could change the schema, but then you would not have a STAR compliant schema.
11	Why does a schema – such as ProcessPartsShipment – contain multiple elements at the highest level ? I think we are all supposed to develop our documents to only the most encompassing of those elements – in this case ProcessPartsShipment . The presence of the other elements at the top level leads me to think that documents could be validated to the other, less encompassing top level elements – like PartsShipment . It seems illogical to produce documents that support only a portion of a BOD	We've developed the BODs so that there is one noun such as PartsOrder, with multiple verbs such as Process, Acknowledge, Change, Cancel. The noun will contain all of the fields needed to support any of the accompanying verbs. We then use the implementation guidelines to show the exact requirements per noun/verb combination. For example, the ProcessPartsOrder Implementation Guidelines might say that the BillTo party is required whereas the AcknowledgePartsOrder Implementation Guidelines say it's optional. Please note that for some of the BODs, a majority of the fields are optional for all the noun/verb combinations. We are currently working on data mapping efforts to further refine those BODs to say what fields would be required for what noun/verb combination.
12	Which BOD version we should use in production, the Developer or Standalone? I noticed that the Developer files are smaller in size	We would suggest using the Standalone for all the BODs in the runtime situation. It will speed up performance significantly. The Developer BOD file is misleading. It is smaller because you are looking at it by itself, but consider that you would have to also load all of its corresponding files (i.e., Components, Fields, DataTypes, Meta, Noun, etc.). All of those additional files have to be processed and therefore take additional time. With the

Reference Number	Question	Response
		Standalone (flattened) version, all the files have been compressed into one, therefore your incoming XML file only has one file to parse against as opposed to 8...sometimes more.
13	Is there a space between /STAR and /STAR in the schema location? xsi:schemaLocation="http://www.starstandards.org/STAR /STAR/Rev2.0/BODs/Standalone/ProcessPartsOrder.xsd"	The schemaLocation must include a whitespace character that separates a Namespace and it's suggested location. This can be either a CRLF or a regular space.
7.4.7	**What does STAR Support?**	
1	Does STAR support DTDs?	No, STAR has chosen to develop all the BODs using schema.
2	Does STAR support XDR?	No, STAR has chosen to develop all the BODs using XML schema.
3	If we make extensions to the schema to add our unique data requirements, would it still be considered STAR standard?	If any extensions or modifications are made to the STAROAGI BODs they would not be considered STAR standard. If your organization had additional data requirements they should be submitted to STAR Architect for inclusion in the STAR standard BOD
4	Does STAR provide XSL files that will transfer XML to viewable html page?	STAR does not support converting the XML files to html. STAR standards only cover the transfer of the data between two systems.
5	Does STAR provide XSL files that will transfer ADF STAR Sales Lead?	STAR does not provide XSL to do the conversion from the ADF to the STAR Sales Lead BOD. However, STAR does provide a mapping from the ADF to The STAR BOD in the STAR Implementation Guidelines for the Sales Lead.
6	What is the difference between the XML schema and the XML file that is sent across the wire?	The schema is the representation of what the XML data transfer file could be. Every field (data tag) would be in the schema definition. A XML instance file generated from the schema is what is sent on a transaction (What STAR has shown as sample XML in the STAR XML Repository). The XML file would only contain the fields that had data in them. If they wanted to have the fields with null data that would be an implementation decision, but this is not necessary.
7	How does STAR help me to automate the transfer of information between different databases and applications?	STAR BODs define the information that is to be transferred between two systems at the point of information exchange. At this time, STAR does has not determined how the BODs would be used within any application.
8	XML-based Web services are typically found via searches in Web service registries, such as UDDI or ebXML. If a Web service implementation declares compliance with a UDDI tModel (by associating that tModel's key with the service's binding), Web service clients can find all instances of a well-known, standard	The Retail Automotive industry supports a relatively small number of companies. Dealers and OEMs have business agreements and exchange business information daily. There are not new OEMs or Dealers entering our industry daily.

Reference Number	Question	Response
	Web service interface. Are there plans to register some of the STAR schema as UDDI tModels?	
9	Do I have to be a STAR member to access the STAR standards?	The standards that STAR has developed are open, voluntary standards. All STAR standards are published on a public website (www.starstandard.org) where anyone can download and make use of them. This applies to all the standards that STAR creates including all DTS, XML Messaging standards and Infrastructure standards. However those standards in process and under development are only produced in the STAR workgroups and are not available to the general public.
7.4.8	**BOD Questions**	
7.4.8.1	**ConfirmBOD**	
1	Is the Confirm BOD a required transaction?	The Confirm BOD is an Optional transaction. However, if it is used, It is a recommended best practice that the Confirm BOD only be sent "On Change", i.e., only when errors have occurred.
2	Is a Confirm BOD ever confirmed? How to stop confirm loop for ever?	The confirmation attribute should be turned off for the Confirm BOD, i.e., you should not confirm a Confirm BOD
3	How would the Confirm BOD be implemented in both the ebXML and Web Services worlds?	The use of the Confirm BOD is consistent across all collaborations. Whether the implementer is working with Repair Order, Parts Order, Credit Application, etc., the use of the Confirm BOD remains the same. This is also true for chosen transport method, the use of the Confirm BOD will always be the same. The way it is implemented however is up to the implementer and is out of scope for STAR.
4	When I send a BOD requesting an Acknowledgment in return, what happens when the attribute for the Acknowledgment is set to "Always" (i.e. <oa:Process confirm="OnChange" acknowledge="Always"/>) and an error causes the ConfirmBOD to be sent?	Once an error occurs, the ConfirmBOD provides the error information. Any further processing that the Acknowledge would communicate is either not possible or unreliable. So, the Always is ignored if a ConfirmBOD has been sent.
5	How do I know whether I should be using the Confirm BOD or the Acknowledgexxx BOD to return errors?	It depends on the type of errors that your receive. (i.e., Application/Business or Semantics/Structure errors) The Confirm BOD does not catch business level errors. All business level errors would be sent back in a responding BOD. The easiest way to explain it is with an example. Lets take Parts Order: **BUSINESS LEVEL ERRORS** 1. I send you a Process Parts Order. 2. You receive it and it parsers successfully so you send it into your

Reference Number	Question	Response
		application. 3. Once its' in your application you find that one of the Part Numbers I submitted is invalid and one of the quantities that you submitted is invalid. These are business level errors because they've made it through the parsing and into your application. 4. You would then send me back an Acknowledge Parts Order indicating that there were two errors for one of my line items, but the rest of the order can be fulfilled. **CONFIRM BOD ERRORS (SEMANTICS & STRUCTURE)** 1. I send you a Process Parts Order. One of the fields in the Process PO, lets say OrderType requires a list of enumerated values. I put in a value that is not in that list. 2. You receive it and try to parse it. Because I have put in a value that does not match the list of valid values the parser fails 3. You send me back a Confirm BOD with those Semantic errors.
6	Why doesn't Confirm BOD have a request and a response?	When a BOD is sent, there is an option to say whether you want a confirmation. If you say "Yes", then the Confirm BOD is sent. The Confirm BOD tells you the success, partial success or failure of either the entire BOD or for each noun within the BOD.
7.4.8.2	**CreditApplication**	
1	One of our RSPs is stating that all communication between us and themselves should be done using the ProcessCreditApplication message type and never the ProcessCreditDecision message type. We do not have a real strong opinion either way but we want to be sure any implementation/standard being set is consistent with your defined use of these message types.	So to recap, you will be generating a Credit Application as the intermediary, sending it off to the finance co, then sending a copy to the DMS for reporting purposes. The DSPs however, have built their systems to only accept a ProcessCreditDecision. Given that Star's collaborations are merely "suggestions" and the way the BODs are implemented is ultimately up to the user, my recommendation would be to use the ProcessCreditDecision. There are some things to keep in mind however if this is the approach that you take, some of which may be obvious but worth mentioning: 1. The Application fields must be used as they have been defined, as must the Decision fields. 2. All required Decision-related fields must be populated even though you are sending a "null Credit Decision". 3. We will need to ensure that any modifications made to the Application portion of the Credit Decision BOD must also be reflected in the Credit

Reference Number	Question	Response
		Application BOD and vice versa. These two BODs must remain consistent.
7.4.8.3	**CreditContract**	
1	Credit Contract 'Financing' component element "FederalTILDisclosures" has an element called 'FirstPaymentDate'. Can you tell me which definition to apply, A or B? A) the date the first scheduled payment would be due (which would likely be a future date from the date of the contract; e.g.: the first date of a billing cycle); B) the date money is needed to release the car from a dealer to the customer (something like a payment a customer might have to make before being given the vehicle, which might include down payments or vehicle pick-up payments)	The answer is A) the date the first scheduled payment would be due (which would likely be a future date from the date of the contract; e.g.: the first date of a billing cycle)
2	Credit Contract 'Vehicle' component element "Option" has an element called 'PortInstalledInd'. The description reads "Indicates Port-installed accessories". Is this a typo?	The answer is it is Port-Installed. There is no typo. The description was right
3	Credit Contract 'Financing' component element "Fee" has an element called 'FeeType'. The description reads "Identifies a type of fee." It is an element with a restricted enumeration list in the schema...Is "LocalFees" one of the enumerated values?	There's no Local Fees. You'd have to request to have that added.
7.4.8.4	**PartsOrder**	
1	It we need to make updates to a Parts Order, would this necessitate resending of the PO? And if so, would we have to somehow flag it as a re-send?	Yes, it would. If you resend an updated Parts Order, you would use the ChangePartsOrder BOD. The Change verb assumes replacement of fields sent, with the exception of the identifying fields of the Parts Order.
7.4.8.5	**Retail Delivery Reporting**	
1	Does the STAR Retail Delivery Reporting DTS handle all of the following sales transactions? Sales - Retail Delivery Reporting, Vehicle Transfer, RDR Cancel and RDR Change	Yes it does. You would use the ProcessRetailDelivery to report a sale. There is a Dealer Transfer Date and a Transferring Dealer component that would be used on the ProcessRetailDelivery BOD to transfer a vehicle. You would use the ChangeRetailDeliveryReporting BOD to change the prior reporting and the CancelRetailDeliveryReporting BOD to cancel Prior reporting. Please refer to the Retail Delivery Reporting Implementation

Reference Number	Question	Response
		Guidelines for the information available on each BOD.
7.4.8.6	**SalesLead**	
1	In the Sales Activity section, I need two time stamps. The standard provides for one (the ActivityDateTime, which I'm planning to use for the date and time that the lead was acted on. I would also like to have a date and time stamp for the time that the lead was received by the dealership. In this way, we can determine how long it took the dealership to act on the lead. Can we just add an xml tag for this date/time?	What I would propose is to use the Activity component to address both of these needs. The Activity component can occur multiple times. It could also be sent in two different BODs. For Example -- The first BOD "ProcessSaelsLead" would have the Activity component with the ActivityId identifying the first event. The second BOD "ChangeSalesLead" would have the Activity component with the ActivityId identifying the second event. Each of these activities would have their own ActivityDateTime.
7.4.8.7	**ServiceProcessingAdvisory / ServiceProcessingReceiptAcknowledgement**	
1	Are both of these BODs specifically limited to Warranty claim payments versus Service Contract, Transportation, Parts (e.g. Mopar), Recall?	Yes, these BODs were defined for the Warranty Claim payments. If you require BODs for other types of payments, please submit a request for new BODs to accommodate your needs. Keep in mind that the ClaimType field on the Warranty component can be used to define the type of Claim.
2	What is the relationship between a "claim submission" from a dealer to OEM and ServiceProcessingReceiptAcknowledgement? This is unclear based on current dealership and OEM processes	The Warranty Claim submission from a dealer usually comes in on the RepairOrder BOD. The ServiceProcessingReceiptAcknowledgement BOD is used to notify the dealer that the Warranty Claim request has been received.
3	Is the use of the ServiceProcessingAdvisory to return 'batched' claim payment information a misuse of the BOD? We're looking for the BOD basically detailing the claims being paid in an EFT batch which the DMS uses to update claim payments in the accounting portion of their software. This use is technically a status in that it indicates payment or a chargeback, but not a true status which would typically contain ALL open claims submitted thru paid. Further it is more than a status, it's actually payment detail.	ServiceProcessingAdvisory BOD was designed to send a batch of claim payment information to a dealer. For example, you provide the method of payment (EFT, Check, Parts Credit or Other), the number of claims being sent and either the Payment or Disposition information for each claim being sent in this BOD.
7.4.9	**Data Mapping**	
1	There is some confusion here on BOD Data Mapping. My understanding from the past is that I have had only to tell you whether or not our application supports the different fields on the BOD spreadsheet. How I've heard this	File name and field lengths??? No way. On the spreadsheet it tells you exactly what needs to be completed. In the population column you indicate if it's Required, Optional, Populated or Not Populated. Those have all been defined on the top of the spreadsheet. In the Qualifier column you

Reference Number	Question	Response
	morning that you are now looking for file names and field lengths also. Is this true?	put in any valid values that you use. In the Definition column you indicate how you are defining the field, especially if you are defining it differently.
2	Can you explain to be what you want to see in each of the fields besides the legend indicating the letters. Other words what do you mean by field population, what do you mean by Valid Values or Qualifier and what do you mean by Field description? Are you looking just for whether we support it? Or do you want the information like you have a part order and it's sent to the host. We have an order number, date, and dealer codes. How would we indicate in these columns?	Maybe an example would help. Let's say we are talking about Process Parts Order. Typically the DMS would be sending the information to the OEM. So the RSP would be the Sender and the OEM is the receiver. So keeping that in mind, lets look at the **Field Population** column. If there is a field that you send you would put a **P** in the column. If there is a field that you do not send you would put **N**. Now let's say we are talking about the Acknowledge PO. Typically the OEM would be sending the information to the DMS. So the RSP would be the Receiver and the OEM would be the Sender. So keeping that in mind, lets look at the **Field Population** column. If there is a field that you require the OEM to send you would put an **R** in the column. If there is a field that you do not require but would use if the OEM sent it you would put **O** for optional. Now if there is a field that you just completely ignore you would put an **I**. Now let's look at the **Valid Values or Qualifier** column. There may be a field that you as either the sender or the receiver may have certain valid values for. This column is where you would define those values. For the **Definition** column, you would use this column to indicate if you have defined a field differently from the way STAR has defined a field. This will help us figure out if OEMs are using fields for purposes other than what they've been defined for
3	Can you please give us an example of Valid Values?	Before we identified an enumerated list for AlternateShipmentPriorites, an OEM had their own list of values that they would accept: 1=Air Freight Collect 2=Fastest Way 3=Best Surface
4	Can you please give us an example of Definition?	We define DocumentDateTime as "Date and time document was created". But one OEM was defining at as "Software release date".