

# ACES DOCUMENTATION

**Aftermarket Catalog Exchange Standard**

Version 4.0

8/28/2018



**Revision History**

Revision Number	Revision Date	Summary of Changes	Changes Marked?
1.0	7/22/02	First issue	N
1.01	9/19/02	Added a required vcdbReleaseNumber tag to the Header section. Also added a discussion on delivering invalid applications (Section 4.9)	N
1.02		Misc. Changes	N
1.03	9/25/03	Update tag definitions and schema. Remove VQ narrative and modify Legacy section.	N
1.04	3/05/04	Add new name “ACES”. Change AAIA tag to ACES. Split Mfr tag into TransmissionMfr and TransferCaseMfr tags. Change all RegionAbbr references to Region. Change all SteeringGearType references to SteeringType. Add discussion of abbreviations used for missing vehicle attributes. Add draft for Vehicle Qualifiers. Update Appendix B. Update the PCDB schema in Appendix H.	N
1.05	3/25/04	Remove Mfr tag from Appendix A. Remove support for Paired Values format and correct Appendix references. Correct VQ Tag description. Add examples of complex expressions. Add clarification to Net Change output.	N
1.06	3/26/04	Miscellaneous clarifications.	N
1.07	8/18/04	Updated schema diagrams. Removed all references to VQ until final approval by Committee.	N
1.08	10/19/05	Corrected Example in Appendix C. Correct character encoding used in 4.2. Add RecordCount to tag definitions. Change EngVIN to EngineVIN in Appendix A & C. Correct Reserved Symbols in 5.3. Change the usage to optional for DocumentFormNumber and required for DocumentTitle in 4.10. Add 5.7 on XML element order.	N
2.0	5/19/06	ACES 2006. Removed ER diagrams from Appendix (now delivered in separate documentation).	N
3.0	7/21/09	Dropped RestraintType, TransferCase, TransferCaseBase, and TransferCaseMfr elements. Added multi-brand support. Added support for Application Digital Assets. Added delivery tags for PowerOutput and VehicleSubType. General clean up and corrections.	N
3.0	8/24/09	Removed VehicleSubType reference since it was dropped before final release of the xsd.	N
3.0.1	11/26/2012	General Document clean-up: typos, grammar, etc. Information added on the VCDB Efficiency Project, PCDB Part To Use table, and Legacy Sunset.	N
3.0.1	8/30/2013	PcdbVersionDate, QdbVersionDate, VCdbVersionDate – Removed “Zero fill if necessary	N

3.1	10/28/2013	Updated Field Definition Document to reflect changes to DataType in EngineConfig table. Updated all documentation to reflect version "3.1"	N
3.2	10/27/2015	Change Aftermarket Catalog Enhanced Standard to Aftermarket Catalog Exchange Standard; Updated XSD; added definitions N/R, N/A, and N/S; Added VCdb Change Log Table Field Definitions	N
3.2 rev 2	12/16/2015	Updated documentation for the digital asset metadata moving from the asset tag to a new DigitalAsset tag. This minimizes duplication of digital asset metadata repeating within an ACES xml file.	N
3.2 rev 3	2/2/2016	Fixed spelling/typos in section 5.11. Removed confusing Examples in Appendix D related to application and vehicle asset delivery.	N
3.2 rev 4	2/29/2016	Updated 5.8.1 and 5.8.4 how an image is defined and updated verbiage in Appendix D.	N
3.2 rev 5	3/31/2016	Modified 5.8 to improve verbiage for how the image section can be used.	N
4.0 rev 1	8/28/2018	Updated VCdb Rules, added new content for vehicle class, engine bore & stroke, body configuration, and equipment in the documentation and in the ACES XSD. XSD has also been updated to restrict submission type. On page 43, change "Either the BaseVehicle tag or the Make and Years tags are required for each application" to "One of the following must be sent in the Vehicle Ident Group: a Base Vehicle ID; a Make / Year or Make / Year-Range combination must be included with each application; a Base Equipment ID; or a Mfr / Equipment Model / Vehicle Type". No changes to XSD or other documentation. Similar changes should be made for "Make", "Model", and "Year" on pages 46 & 48	N
4.0 rev 2	1/18/2019	Added the source field to VehicleToBodyConfig table and updated order of fields VehicleToBodyConfig table in section 2.7 Changed field name from Class to ClassName in Class Table in section 2.7	N

## Table of Contents

1 Introduction .....	6
2 Vehicle Configuration database (VCdb).....	6
2.1 VCdb Mission Statement.....	6
2.2 VCdb Structure .....	6
2.3 VCdb Efficiency - Stages .....	6
2.4 VCdb Change Management.....	7
2.5 VCdb Change Log .....	7
2.6 VCdb ID Retention.....	8
2.7 VCdb Table Field Definitions.....	9
3 Parts Classification database (PCdb) .....	40
3.1 PCdb Mission Statement.....	40
3.2 PCdb Structure .....	40
3.3 Positions .....	40
3.4 Valid Positions .....	41
3.5 Multifunction Parts .....	41
3.6 PCdb Change Management.....	41
4 Qualifier database (Qdb).....	42
4.1 Reasons for having “Coded Qualifiers” .....	42
4.2 Qdb Structure.....	42
4.3 Qdb Version Control .....	43
4.4 Qdb Parameters.....	43
4.5 Qdb Parameter Types.....	44

4.6 Qualifier Language Translation .....	44
4.7 Qdb Data Rules .....	45
4.8 Qdb Review Process .....	47
5 Application Data Transfer File .....	48
5.1 Transfer File Purpose.....	48
5.2 Physical Characteristics .....	48
5.3 Transfer File Naming Convention .....	48
5.4 Delivering Changes Only .....	49
5.5 ACES Root Element .....	49
5.6 Header Information .....	49
5.7 Applications.....	50
5.8 Digital Assets .....	56
5.9 Footer Information.....	57
5.10 General Rules .....	57
5.11 Delivering Invalid Applications .....	58
5.12 XML Elements.....	59
6 XML Specifics .....	77
6.1 Declaration Statement.....	77
6.2 Character Encoding .....	77
6.3 Reserved Symbols.....	77
6.4 Comments.....	77
6.5 Empty Element Tags.....	78
6.6 Element Tag Order .....	78
7 Legacy Make Model Table .....	78

Appendix A. ACES Element Data Tags .....	79
Appendix B. Example Transfer File.....	81
Appendix C. Delivering “Invalid” Applications .....	83
Appendix D. Delivering Application Digital Assets .....	84

**Notice**

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

©Copyright 2002-2018 Auto Care Association

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

# 1 Introduction

This document explains the structure of the Aftermarket Vehicle Database and how to deliver your catalog information in terms of that database. This specification was designed with extensive input from a broad coalition of participants in the Automotive Aftermarket, including all leading electronic catalog companies. The objective was to arrive at a standard sufficient to meet the needs of the widest variety of trading partners. With a widely accepted industry standard for the exchange of applications catalog data, the number of proprietary methods of coding and exchange can be reduced, lessening the cost and time delays associated with supporting multiple formats. Some of the rules in the standard allow flexibility in how data is formatted and exchanged. Specific data partners may impose requirements in how the standard is practiced with them. But it is essential for the industry to follow the standard as closely as possible if the full business value is to be realized.

## 2 Vehicle Configuration database (VCdb)

### 2.1 VCdb Mission Statement

Design, populate, and implement a data repository that contains the most common attributes required to describe a vehicle in the Automotive Aftermarket. Vehicle configurations will be defined and validated at the lowest level that is economically feasible but will not exclude the delivery of applications at higher levels. Publish the database and a specification to deliver catalog applications for the purpose of establishing an industry standard.

### 2.2 VCdb Structure

The Vehicle Configuration database is a normalized relational database made up of over sixty-five tables, representing over fifty vehicle attributes. Configuration tables are used to define valid combinations between the vehicle and several major vehicle systems.

### 2.3 VCdb Efficiency - Stages

The VCdb Efficiency Project was introduced in 2012 as a way to get information into the VCdb much faster. To accomplish this, vehicle additions to the database are made in four stages, beginning with the Base Vehicle, Submodel, and Base Engine information (Stage 1). As more complete information about the vehicle becomes available from OEM sources, the vehicles will be published and move through the remaining stages. The value, PublicationStageID, is based on a new table, PublicationStage. The PublicationStageID of a vehicle tracks the current stage and is stored in the Vehicle table.

Stages are defined as:

- **Stage 1:** Base Vehicle, Submodel, and Base Engine Information is published
- **Stage 2:** Full Engine Configuration is published
- **Stage 3:** Remaining VCDB Attributes are published
- **Stage 4:** Complete Vehicle Configuration\* is published

\*The Vehicle Configuration table was retired in October 2013.

## 2.4 VCdb Change Management

Representatives of the Auto Care Association are responsible for updating and publishing the VCdb monthly. In addition to a fully refreshed dataset, all changes are documented and reported separately in the release notes <http://www.autocare.org/What-We-Do/Technology/release-notes/>. Petitions for changes or additions to the VCdb will be accepted from any licensed subscriber and reviewed in the regular update cycle. These change requests are maintained and monitored through <https://autocarevip.com>

## 2.5 VCdb Change Log

When data changes, changes are recorded in the Vehicle Configuration database (VCdb) change log tables\*.

VCdb change log new structure includes:

- A reference ID to what table is changing
- The primary key and the column (field) within the table that is changing
- The before and after values of the change
- An ID to identify the change (Add, Delete, Modify)
- A change table stating when the change occurred

\*The VCdb change log tables were introduced in April 2015

## 2.6 VCdb ID Retention

The VCdb structure is composed of over 70 tables. Data within these tables are linked together by record identification (IDs) fields. Within the VCdb, each table row or record has a unique numeric Primary Key ID (PKID). The unique PKID value can be used in one or more tables as Foreign Key IDs (FKID) to identify the relationship between the records in various data tables. The IDs numeric value is the same in both tables which creates relationship between the tables and specific records within each table. ID fields within the VCdb have a suffix of “ID”.

When the VCdb was created, the ability to update records within the database could only change by adding and deleting records. This means when a change to a record occurred, the IDs are deleted for all records linking to PKID, all the FKID records from multiple tables would be removed as well. New records with the changes are then added to the database with new ID record connections.

Between VCdb releases, the IDs could change drastically. For a cataloging mapping software company using VCdb IDs within their system, the IDs changing from month to month created a need for tracking differences between the releases and updating their systems accordingly. This tracking process can take hours, days or weeks depending on the automation levels available in each system. In addition to the challenge of tracking changes, lack of ID retention makes systems more prone to errors such as dropping records, improper deletes, orphan records, and others data errors which are difficult to detect and cumbersome to correct with constantly changing IDs.

With the implementation of VCdb ID Retention, data value changes can occur with no change to the IDs. This process enables mapping software to choose an update process that minimizes ID errors and potentially reduces the update processing time within their systems.

The VCdb ID Retention introduced in April 2015

## 2.7 VCdb Table Field Definitions

Table: Abbreviation

Field	Data Type	Description/Example
Abbreviation	Text (3)	N/A, N/R, U/K, - (dash). Each base table will have one row for each of these values. U/K will initially be used internally by MOTOR to differentiate values that have not been researched yet from values that are N/A.  The dash (-) indicates a value intentionally left blank, except for Wheelbase which was not populated for cars.
Description	Text (20)	Not Available (N/A), Not Required (N/R), Unknown (U/K), Intentionally blank (-).
LongDescription	Text(200)	A sentence describing the abbreviation.

Table: Aspiration

Field	Data Type	Description/Example
AspirationID	Number (Long Integer)	Unique, system generated identifier.
AspirationName	Text (30)	Aspiration or Boost Type name - Normal, Turbocharger, Supercharged, etc.

Table: Attachment

*Note: This table is currently delivered without data.*

Field	Data Type	Description
AttachmentID	Number (Long Integer)	Unique, system generated identifier.
AttachmentFileName	Text (50)	TBD
AttachmentURL	Text(100)	TBD
AttachmentDescription	Text(50)	TBD

AttachmentTypeID	Number (Long Integer)	Unique, system generated identifier from the AttachmentType table
------------------	-----------------------	---

Table: AttachmentType

*Note: This table is currently delivered without data.*

Field	Data Type	Description
AttachmentTypeID	Number (Long Integer)	Unique, system generated identifier.
AttachmentTypeName	Text(20)	TBD

Table: BaseVehicle

Field	Data Type	Description
BaseVehicleID	Number (Long Integer)	Unique, system generated identifier.
YearID	Number (Long Integer)	Unique, system generated identifier from the Year table – ID and year value are same.
MakeID	Number (Long Integer)	Unique, system generated identifier from the Make table.
ModelID	Number (Long Integer)	Unique, system generated identifier from the Model table.

Table: BedConfig

Field	Data Type	Description
BedConfigID	Number (Long Integer)	Unique, system generated identifier.
BedLengthID	Number (Long Integer)	Unique, system generated identifier from the BedLength table.
BedTypeID	Number (Long Integer)	Unique, system generated identifier from the BedType table.

Table: BedLength

Field	Data Type	Description
BedLengthID	Number (Long Integer)	Unique, system generated identifier.
BedLength	Text (10)	Pickup bed length represented in inches to one decimal point of precision
BedLengthMetric	Text(10)	Pickup bed length represented in centimeters to one decimal point of precision

Table: BedType

Field	Data Type	Description
BedTypeID	Number (Long Integer)	Unique, system generated identifier.
BedTypeName	Text (50)	Marketing or industry accepted terminology for bed type (Styleside, Fleetside).

Table: BodyNumDoors

Field	Data Type	Description
BodyNumDoorsID	Number (Long Integer)	Unique, system generated identifier.
BodyNumDoors	Text(3)	Numerical representation for number of doors ranging from 0 to 6.

Table: BodyStyleConfig

Field	Data Type	Description
BodyStyleConfigID	Number (Long Integer)	Unique, system generated identifier.
BodyNumDoorsID	Number (Long Integer)	Unique, system generated identifier from the BodyNumDoors table.

BodyTypeID	Number (Long Integer)	Unique, system generated identifier from the BodyType table.
------------	-----------------------	--

Table: BodyType

Field	Data Type	Description
BodyTypeID	Number (Long Integer)	Unique, system generated identifier.
BodyTypeName	Text (50)	Industry accepted name for body type - Coupe, Sedan, etc.

Table: BrakeABS

Field	Data Type	Description/Example
BrakeABSID	Number (Long Integer)	Unique, system generated identifier.
BrakeABSName	Text (30)	Brake ABS: Non-ABS, 2-Wheel ABS or 4-Wheel ABS.

Table: BrakeConfig

Field	Data Type	Description
BrakeConfigID	Number (Long Integer)	Unique, system generated identifier.
FrontBrakeTypeID	Number (Long Integer)	Unique, system generated identifier from the BrakeType table – for the front of the vehicle.
RearBrakeTypeID	Number (Long Integer)	Unique, system generated identifier from the BrakeType table – for the rear of the vehicle.
BrakeSystemID	Number (Long Integer)	Unique, system generated identifier from the BrakeSystem table.
BrakeABSID	Number (Long Integer)	Unique, system generated identifier from the BrakeABS table.

Table: BrakeSystem

Field	Data Type	Description/Example
BrakeSystemID	Number (Long Integer)	Unique, system generated identifier.
BrakeSystemName	Text (30)	Brake system: Manual or Power.

Table: BrakeType

Field	Data Type	Description/Example
BrakeTypeID	Number (Long Integer)	Unique, system generated identifier.
BrakeTypeName	Text (30)	Brake type: Disc or Drum.

Table: ChangeAttributeStates

Field	Data Type	Description
ChangeAttributeStateID	Number (Long Integer)	Unique, system generated identifier
ChangeAttributeState	Text (255)	An indicator of the intent of the change request in terms of physical action on the target record within the target table to insert, update, or delete data.

Table: ChangeDetails

Field	Data Type	Description
ChangeDetailID	Number (Long Integer)	Unique, system generated identifier
ChangeID	Number (Long Integer)	Foreign key ID which links the associated record from the Changes table.
ChangeAttributeStateID	Number (Long Integer)	Foreign key ID which links the associated ChangeAttributeState to reference the record's state before and/or after the change is applied.

TableNameID	Number (Long Integer)	Foreign key ID which references the affected table within VCdb.
PrimaryKeyColumnName	Text (55)	The field name containing the primary key id for the table where the data is changing.
PrimaryKeyBefore	Number (Long Integer)	ID value which references the associated record prior to change within the table which is referenced by the TableNameID, and field referenced by the PrimaryKeyColumnName (applicable for delete and modify requests).
PrimaryKeyAfter	Number (Long Integer)	ID value which references the associated record after change within the table which is referenced by the TableNameID, and field referenced by the PrimaryKeyColumnName (applicable for delete and modify requests).
ColumnName	Text (55)	The field impacted by the change.
ColumnValueBefore	Text (255)	The actual value of the field referenced by the change request.
ColumnValueAfter	Text (255)	The actual value of the field impacted by the change.

Table: ChangeReasons

Field	Data Type	Description
ChangeReasonID	Number (Long Integer)	Unique, system generated identifier
ChangeReason	Text (255)	The reason that is selected upon approval of the change request to indicate the basis of the request.

Table:Changes

Field	Data Type	Description
ChangeID	Number (Long Integer)	Unique, system generated identifier.
RequestID	Number (Long Integer)	Foreign key ID which references the request in the Request table.
ChangeReasonID	Number (Long Integer)	Foreign key ID which provides the reason that is selected upon approval of the change request to indicate the basis of the request.
RevDate	Date	Date of last revision of the record.

Table: ChangeTableNames

Field	Data Type	Description
TableNameID	Number (Long Integer)	Unique, system generated identifier
TableName	Text (255)	The VCdb table name to which the change is applied.
TableDescription	Text (255)	A description of the table’s purpose as it relates to the VCdb.

Table: Class

Field	Data Type	Description/Example
ClassID	Number (Long Integer)	Unique, system generated identifier.
ClassName	Text (1)	Vehicle weight class (1-8)

Table: CylinderHeadType

Field	Data Type	Description/Example
CylinderHeadTypeID	Number (Long Integer)	Unique, system generated identifier.
CylinderHeadTypeName	Text (30)	Value of cylinder head name, SOHC, DOHC and OHV.

Table: DriveType

Field	Data Type	Description/Example
DriveTypeID	Number (Long Integer)	Unique, system generated identifier.
DriveTypeName	Text (30)	Drive Type name or description – FWD, AWD, RWD, 4WD.

Table: ElecControlled

Field	Data Type	Description/Example
ElecControlledID	Number (Long Integer)	Unique, system generated identifier
ElecControlled	Text (3)	A boolean (Y/N) indicator of whether the Transmission is electronically controlled

Table: EngineBase

Field	Data Type	Description/Example
EngineBaseID	Number (Long Integer)	Unique, system generated identifier.
Liter	Text (6)	Number of engine liters to one decimal point of precision - 2.8, 3.0, etc.
CC	Text (8)	Cubic centimeters represented as a whole number – 2800, 3200, etc.
CID	Text (7)	Cubic inches of displacement represented as a whole number – 350, 400, etc.
Cylinders	Text (2)	Number of engine cylinders represented as a whole number – 4, 6, 8, etc.
BlockType	Text (2)	Engine block type – V, In-line, Rotary, etc.

EngBoreIn	Text (10)	Engine cylinder bore in inches to a minimum of two decimal points of precision – 1.14, 3.50, etc., or to a maximum of four decimal points of <u>significant</u> precision – eg. 1.233 not 1.2330.
EngBoreMetric	Text (10)	Engine cylinder bore in millimeters to a minimum of one decimal point of precision – 28.0, 32.5, etc., or to a maximum of four decimal points of <u>significant</u> precision – eg. 2.093 not 2.0930
EngStrokeIn	Text (10)	Engine cylinder stroke in inches to a minimum of two decimal points of precision – 1.14, 3.50, etc., or to a maximum of four decimal points of <u>significant</u> precision – eg. 1.233 not 1.2330.
EngStrokeMetric	Text (10)	Engine cylinder stroke in millimeters to a minimum of one decimal point of precision – 54.0, 67.5, etc., to a maximum of four decimal points of <u>significant</u> precision – eg. 2.093 not 2.0930.

Table: EngineBase2

Field	Data Type	Description/Example
EngineBaseID	Number (Long Integer)	Unique, system generated identifier.
EngineBlockID	Number (Long Integer)	Unique, system generated identifier from the EngineBlock table.
EngineBoreStrokeID	Number (Long Integer)	Unique, system generated identifier from the EngineBoreStroke table.

Table: EngineBlock

Field	Data Type	Description/Example
EngineBlockID	Number (Long Integer)	Unique, system generated identifier.

Liter	Text (6)	Number of engine liters to one decimal point of precision - 2.8, 3.0, etc.
CC	Text (8)	Cubic centimeters represented as a whole number – 2800, 3200, etc.
CID	Text (7)	Cubic inches of displacement represented as a whole number – 350, 400, etc.
Cylinders	Text (2)	Number of engine cylinders represented as a whole number – 4, 6, 8, etc.
BlockType	Text (2)	Engine block type – V, In-line, Rotary, etc.

Table: EngineBoreStroke

Field	Data Type	Description/Example
EngineBoreStrokeID	Number (Long Integer)	Unique, system generated identifier.
EngBoreIn	Text (10)	Engine cylinder bore in inches to a minimum of two decimal points of precision – 1.14, 3.50, etc., or to a maximum of four decimal points of <u>significant</u> precision – eg. 1.233 not 1.2330.
EngBoreMetric	Text (10)	Engine cylinder bore in millimeters to a minimum of one decimal point of precision – 28.0, 32.5, etc., or to a maximum of four decimal points of <u>significant</u> precision – eg. 2.093 not 2.0930
EngStrokeIn	Text (10)	Engine cylinder stroke in inches to a minimum of two decimal points of precision – 1.14, 3.50, etc., or to a maximum of four decimal points of <u>significant</u> precision – eg. 1.233 not 1.2330.
EngStrokeMetric	Text (10)	Engine cylinder stroke in millimeters to a minimum of one decimal point of precision – 54.0, 67.5, etc., to a maximum of four decimal points of <u>significant</u> precision – eg. 2.093 not 2.0930.

Table: EngineConfig

Field	Data Type	Description
EngineConfigID	Number (Long Integer)	Unique, system generated identifier.
EngineDesignationID	Number (Long Integer)	Unique, system generated identifier from the EngineDesignation table.
EngineVINID	Number (Long Integer)	Unique, system generated identifier from the EngineVIN table.
ValvesID	Number (Long Integer)	Unique, system generated identifier from the Valves table.
EngineBaseID	Number (Long Integer)	Unique, system generated identifier from the EngineBase table.
FuelDeliveryConfigID	Number (Long Integer)	Unique, system generated identifier from the FuelDeliveryConfig table.
AspirationID	Number (Long Integer)	Unique, system generated identifier from the Aspiration table.
CylinderHeadTypeID	Number (Long Integer)	Unique, system generated identifier from the CylinderHeadType table.
FuelTypeID	Number (Long Integer)	Unique, system generated identifier from the FuelType table.
IgnitionSystemTypeID	Number (Long Integer)	Unique, system generated identifier from the IgnitionSystemType table.
EngineMfrID	Number (Long Integer)	Unique, system generated identifier from the Mfr table for the engine manufacturer, i.e. Chevrolet, Honda, etc.
EngineVersionID	Number (Long Integer)	Unique, system generated identifier from the EngineVersion table.
PowerOutputID	Number (Long Integer)	Unique, system generated identifier from the PowerOutput table

Table: EngineConfig2

Field	Data Type	Description
EngineConfigID	Number (Long Integer)	Unique, system generated identifier.
EngineDesignationID	Number (Long Integer)	Unique, system generated identifier from the EngineDesignation table.
EngineVINID	Number (Long Integer)	Unique, system generated identifier from the EngineVIN table.
ValvesID	Number (Long Integer)	Unique, system generated identifier from the Valves table.
EngineBaseID	Number (Long Integer)	Unique, system generated identifier from the EngineBase table.
EngineBlockID	Number (Long Integer)	Unique, system generated identifier from the EngineBlock table.
EngineBoreStrokeID	Number (Long Integer)	Unique, system generated identifier from the EngineBoreStroke table.
FuelDeliveryConfigID	Number (Long Integer)	Unique, system generated identifier from the FuelDeliveryConfig table.
AspirationID	Number (Long Integer)	Unique, system generated identifier from the Aspiration table.
CylinderHeadTypeID	Number (Long Integer)	Unique, system generated identifier from the CylinderHeadType table.
FuelTypeID	Number (Long Integer)	Unique, system generated identifier from the FuelType table.
IgnitionSystemTypeID	Number (Long Integer)	Unique, system generated identifier from the IgnitionSystemType table.
EngineMfrID	Number (Long Integer)	Unique, system generated identifier from the Mfr table for the engine manufacturer, i.e. Chevrolet, Honda, etc.

EngineVersionID	Number (Long Integer)	Unique, system generated identifier from the EngineVersion table.
PowerOutputID	Number (Long Integer)	Unique, system generated identifier from the PowerOutput table

Table: EngineDesignation

Field	Data Type	Description/Example
EngineDesignationID	Number (Long Integer)	Unique, system generated identifier.
EngineDesignationName	Text (30)	Manufacturer unique code, 2-TC, etc.

Table: EngineVersion

Field	Data Type	Description/Example
EngineVersionID	Number (Long Integer)	Unique, system generated identifier.
EngineVersion	Text (20)	Engine version, i.e. "Windsor", "Cleveland", "Modified", etc.

Table: EngineVIN

Field	Data Type	Description/Example
EngineVINID	Number (Long Integer)	Unique, system generated identifier.
EngineVINName	Text (5)	Manufacturers code for engine configuration found in VIN (Vehicle Identification Number), i.e. 2, H, etc.

Table: EnglishPhrase

*Note: This table is currently delivered without data.*

Field	Data Type	Description
EnglishPhraseID	Number (Long Integer)	Unique, system generated identifier.
EnglishPhrase	Text (100)	English phrase.

Table: EquipmentBase

Field	Data Type	Description/Example
EquipmentBaseID	Number (Long Integer)	Unique, system generated identifier.
MfrID	Number (Long Integer)	Unique, system generated identifier from the Mfr table.
EquipmentModelID	Number (Long Integer)	Unique, system generated identifier from the EquipmentModel table.
VehicleTypeID	Number (Long Integer)	Unique, system generated identifier from the VehicleType table.

Table: Equipment

Field	Data Type	Description/Example
EquipmentID	Number (Long Integer)	Unique, system generated identifier.
EquipmentBaseID	Number (Long Integer)	Unique, system generated identifier from the EquipmentBase table.
RegionID	Number (Long Integer)	Unique, system generated identifier from the Region table.

Table: Model

Field	Data Type	Description
EquipmentModelID	Number (Long Integer)	Unique, system generated identifier.
EquipmentModelName	Text (100)	Name of model based on manufactures marketing nomenclature.

Table: EquipmentToEngineConfig

Field	Data Type	Description/Example
EquipmentToEngineConfigID	Number (Long Integer)	Unique, system generated identifier.
EquipmentID	Number (Long Integer)	Unique, system generated identifier from the Equipment table.
EngineConfigID	Number (Long Integer)	Unique, system generated identifier from the EngineConfig table.

Table: FuelDeliveryConfig

Field	Data Type	Description
FuelDeliveryConfigID	Number (Long Integer)	Unique, system generated identifier.
FuelDeliveryTypeID	Number (Long Integer)	Unique, system generated identifier from the FuelDeliveryType table.
FuelDeliverySubTypeID	Number (Long Integer)	Unique, system generated identifier from the FuelDeliverySubType table.
FuelSystemControlTypeID	Number (Long Integer)	Unique, system generated identifier from the FuelSystemControlType table.
FuelSystemDesignID	Number (Long Integer)	Unique, system generated identifier from the FuelSystemDesign table.

Table: FuelDeliverySubType

Field	Data Type	Description/Example
FuelDeliverySubTypeID	Number (Long Integer)	Unique, system generated identifier.
FuelDeliverySubTypeName	Text (50)	Description of fuel system configuration, 2BBL (2 barrel carb), TBI (throttle body injection), etc.

Table: FuelDeliveryType

Field	Data Type	Description/Example
FuelDeliveryTypeID	Number (Long Integer)	Unique, system generated identifier.
FuelDeliveryTypeName	Text (50)	Description of fuel delivery method - Carburetion, Fuel Injection, etc.

Table: FuelSystemControlType

Field	Data Type	Description/Example
FuelSystemControlTypeID	Number (Long Integer)	Unique, system generated identifier.
FuelSystemControlTypeName	Text (50)	Fuel system method of control - Electronic, Mechanical, etc.

Table: FuelSystemDesign

Field	Data Type	Description/Example
FuelSystemDesignID	Number (Long Integer)	Unique, system generated identifier.
FuelSystemDesignName	Text (50)	The design or manufacturer of the fuel system - Holley, Bosch, etc.

Table: FuelType

Field	Data Type	Description/Example
FuelTypeID	Number (Long Integer)	Unique, system generated identifier.
FuelTypeName	Text (30)	Fuel type - Diesel, Gas, etc.

Table: IgnitionSystemType

Field	Data Type	Description/Example
IgnitionSystemTypeID	Number (Long Integer)	Unique, system generated identifier.
IgnitionSystemTypeName	Text (30)	Basic ignition system type - Distributor-Breaker, Distributorless, etc.

Table: Language

*Note: This table is currently delivered without data.*

Field	Data Type	Description
LanguageID	Number (Long Integer)	Unique, system generated identifier.
LanguageName	Text (20)	TBD
DialectName	Text(20)	TBD

Table: LanguageTranslation

*Note: This table is currently delivered without data.*

Field	Data Type	Description
LanguageID	Number (Long Integer)	Unique, system generated identifier.
LanguageName	Text (20)	TBD
DialectName	Text(20)	TBD

Table: LanguageTranslationAttachment

*Note: This table is currently delivered without data.*

Field	Data Type	Description
LanguageTranslationAttachment ID	Number (Long Integer)	Unique, system generated identifier.
EnglishPhraseID	Number (Long Integer)	Unique, system generated identifier from the EnglishPhrase table.
LanguageID	Number (Long Integer)	Unique, system generated identifier from the Language table.
Translation	Text(150)	TBD

Table: Make

Field	Data Type	Description/Example
MakeID	Number (Long Integer)	Unique, system generated identifier.
MakeName	Text (50)	Make name - Subaru, Volvo, etc.

Table: Mfr

Field	Data Type	Description/Example
MfrID	Number (Long Integer)	Unique, system generated identifier. Referenced in the EngineConfig and Transmission
MfrName	Text (30)	Name of component manufacturer – Allison, Borg Warner, Ford, Toyota, etc.

Table: MfrBodyCode

Field	Data Type	Description/Example
MfrBodyCodeID	Number (Long Integer)	Unique, system generated identifier.
MfrBodyCodeName	Text (10)	Manufacturers unique code for body platform - "A" body, "C" Platform - Not all manufacturers designate body codes.

Table: Model

Field	Data Type	Description
ModelID	Number (Long Integer)	Unique, system generated identifier.
ModelName	Text (100)	Name of model based on manufactures marketing nomenclature – Malibu, Corolla, etc.
VehicleTypeID	Number (Long Integer)	Unique, system generated identifier from the VehicleType table.

Table: PowerOutput

Field	Date Type	Description
PowerOutputID	Number (Long Integer)	Unique, system generated identifier
HorsePower	Text (10)	Engine Power output expressed in Horsepower, including a decimal point only when required to represent fractional values.
KiloWattPower	Text (10)	Engine Power output expressed in Kilowatts, including a decimal point only when required to represent fractional values.

Table: Region

Field	Data Type	Description/Example
RegionID	Number (Long Integer)	Unique, system generated identifier.
ParentID	Number (Long Integer)	Unique, system generated identifier for the parent region, e.g. this ID would refer to North America for the USA region. Currently, this field is not populated in the VCdb.
RegionAbbr	Text (3)	ISO standard country abbreviations.
RegionName	Text (50)	Description of Region – United States, Canada, Mexico

Table: SpringType

Field	Data Type	Description/Example
SpringTypeID	Number (Long Integer)	Unique, system generated identifier.
SpringTypeName	Text (50)	Suspension spring type - Coil, Leaf, Torsion Bar, Air.

Table: SpringTypeConfig

Field	Data Type	Description
SpringTypeConfigID	Number (Long Integer)	Unique, system generated identifier.
FrontSpringTypeID	Number (Long Integer)	Unique, system generated identifier from the SpringType table – for the front of the vehicle.
RearSpringTypeID	Number (Long Integer)	Unique, system generated identifier from the SpringType table – for the rear of the vehicle.

Table: SteeringConfig

Field	Data Type	Description
SteeringConfigID	Number (Long Integer)	Unique, system generated identifier.
SteeringTypeID	Number (Long Integer)	Unique, system generated identifier from the SteeringType table.
SteeringSystemID	Number (Long Integer)	Unique, system generated identifier from the SteeringSystem table.

Table: SteeringSystem

Field	Data Type	Description/Example
SteeringSystemID	Number (Long Integer)	Unique, system generated identifier.
SteeringSystemName	Text (30)	Steering system: Manual or Power.

Table: SteeringType

Field	Data Type	Description/Example
SteeringTypeID	Number (Long Integer)	Unique, system generated identifier.
SteeringTypeName	Text (30)	Steering type: Gear or Rack.

Table: SubModel

Field	Data Type	Description
SubModelID	Number (Long Integer)	Unique, system generated identifier.
SubModelName	Text (50)	Name of submodel based on manufacturers marketing nomenclature.

Table: Transmission

Field	Data Type	Description/Example
TransmissionID	Number (Long Integer)	Unique, system generated identifier.
TransmissionBaseID	Number (Long Integer)	Unique, system generated identifier from the TransmissionBase table.
TransmissionMfrCodeID	Number (Long Integer)	Unique, system generated identifier from the TransmissionMfrCode table.
TransmissionElecControlledID	Number (Long Integer)	Unique, system generated identifier from the ElecControlledID table
TransmissionMfrID	Number (Long Integer)	Unique, system generated identifier from the Mfr table

Table: TransmissionBase

Field	Data Type	Description
TransmissionBaseID	Number (Long Integer)	Unique, system generated identifier.
TransmissionTypeID	Number (Long Integer)	Unique, system generated identifier from the TransmissionType table.
TransmissionControlTypeID	Number (Long Integer)	Unique, system generated identifier from the TransmissionControlType table.
TransmissionNumSpeedsID	Number (Long Integer)	Unique, system generated identifier from the TransmissionNumSpeeds table.

Table: TransmissionControlType

Field	Data Type	Description/Example
TransmissionControlTypeID	Number (Long Integer)	Unique, system generated identifier.
TransmissionControlTypeName	Text (30)	Type of transmission or transaxle - Automatic, Standard, etc.

Table: TransmissionMfrCode

Field	Data Type	Description/Example
TransmissionMfrCodeID	Number (Long Integer)	Unique, system generated identifier.
TransmissionMfrCode	Text (30)	A manufacturer assigned model or unit designation code - AW03-72L, 4HP22H, etc.

Table: TransmissionNumSpeeds

Field	Data Type	Description/Example
TransmissionNumSpeedsID	Number (Long Integer)	Unique, system generated identifier.
TransmissionNumSpeeds	Text (3)	Number of forward speeds for transmission or transaxle - 3, 4, 5, etc.

Table: TransmissionType

Field	Data Type	Description
TransmissionTypeID	Number (Long Integer)	Unique, system generated identifier.
TransmissionTypeName	Text (30)	Transmission or Transaxle.

Table: Valves

Field	Data Type	Description/Example
ValvesID	Number (Long Integer)	Unique, system generated identifier
ValvesPerEngine	Text (3)	Total number of intake and exhaust vales in the engine – 8, 16, 24, etc.

Table: vcdbChanges

Field	Data Type	Description/Example
Versiondate	Date	Refers to the publication date of the VCdb
TableName	Text (30)	Refers to the name of a Table in the VCdb
ID	Number (Long Integer)	Unique, system generated identifier from a table in the VCdb
Action	Text (1)	The Action made upon the record corresponding with the ID. Add (A), Delete (D), Change (C), U/K ID addition (U)

Table: Vehicle

Field	Data Type	Description/Example
VehicleID	Number (Long Integer)	Unique, system generated identifier.
BaseVehicleID	Number (Long Integer)	Unique, system generated identifier from the BaseVehicle table.
SubModelID	Number (Long Integer)	Unique, system generated identifier from the SubModel table.
RegionID	Number (Long Integer)	Unique, system generated identifier from the Region table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToBedConfig

Field	Data Type	Description/Example
VehicleToBedConfigID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
BedConfigID	Number (Long Integer)	Unique, system generated identifier from the BedConfig table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToBodyConfig

Field	Data Type	Description/Example
VehicleToBodyConfigID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
WheelbaseID	Number (Long Integer)	Unique, system generated identifier from the Wheelbase table.
BedConfigID	Number (Long Integer)	Unique, system generated identifier from the BedConfig table.
BodyStyleConfigID	Number (Long Integer)	Unique, system generated identifier from the BodyStyleConfig table.
MfrBodyCodeID	Number (Long Integer)	Unique, system generated identifier from the MfrBodyCode table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToBodyStyleConfig

Field	Data Type	Description/Example
VehicleToBodyStyleConfigID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
BodyStyleConfigID	Number (Long Integer)	Unique, system generated identifier from the BodyStyleConfig table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToBrakeConfig

Field	Data Type	Description/Example
VehicleToBrakeConfigID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
BrakeConfigID	Number (Long Integer)	Unique, system generated identifier from the BrakeConfig table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToClass

Field	Data Type	Description/Example
VehicleToClassID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
ClassID	Number (Long Integer)	Unique, system generated identifier from the Class table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToDriveType

Field	Data Type	Description/Example
VehicleToDriveTypeID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
DriveTypeID	Number (Long Integer)	Unique, system generated identifier from the DriveType table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToEngineConfig

Field	Data Type	Description/Example
VehicleToEngineConfigID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
EngineConfigID	Number (Long Integer)	Unique, system generated identifier from the EngineConfig table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToMfrBodyCode

Field	Data Type	Description/Example
VehicleToMfrBodyCodeID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
MfrBodyCodeID	Number (Long Integer)	Unique, system generated identifier from the MfrBodyCode table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToSpringTypeConfig

Field	Data Type	Description/Example
VehicleToSpringTypeConfigID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
SpringTypeConfigID	Number (Long Integer)	Unique, system generated identifier from the SpringTypeConfig table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToSteeringConfig

Field	Data Type	Description/Example
VehicleToSteeringConfigID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
SteeringConfigID	Number (Long Integer)	Unique, system generated identifier from the SteeringConfig table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToTransmission

Field	Data Type	Description/Example
VehicleToTransmissionID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
TransmissionID	Number (Long Integer)	Unique, system generated identifier from the Transmission table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleToWheelbase

Field	Data Type	Description/Example
VehicleToWheelbaseID	Number (Long Integer)	Unique, system generated identifier.
VehicleID	Number (Long Integer)	Unique, system generated identifier from the Vehicle table.
WheelbaseID	Number (Long Integer)	Unique, system generated identifier from the Wheelbase table.
Source	Text (10)	Applies only to changes submitted by subscribers, not original data. Contains the source of the information, e.g. OE, internet, catalog.

Table: VehicleType

Field	Data Type	Description/Example
VehicleTypeID	Number (Long Integer)	Unique, system generated identifier.
VehicleTypeName	Text (50)	Car, Truck or Van.
VehicleTypeID	Number (Long)	Unique, system generated identifier from the VehicleTypeGroup table.

Table: VehicleTypeGroup

Field	Data Type	Description/Example
VehicleTypeGroupID	Number (Long Integer)	Unique system generated identifier
VehicleTypeGroupName	Text (50)	Aggregation of vehicle types. For example, Motorcycle and Snowmobile are VehicleTypes of the VehicleTypeGroup Powersports

Table: Version

Field	Data Type	Description
VersionDate	Smalldatetime	Date this version was released to the public.

Table: WheelBase

Field	Data Type	Description
WheelBaseID	Number (Long Integer)	Unique, system generated identifier.
WheelBase	Text (10)	Wheelbase length represented in inches to one decimal point of precision

WheelBaseMetric	Text (10)	Wheelbase length represented in to one decimal point of precision.
-----------------	-----------	--

Table: Year

Field	Data Type	Description
YearID	Number (Long Integer)	Unique identifier for year - Note: value for year is the unique number.

## 3 Parts Classification database (PCdb)

### 3.1 PCdb Mission Statement

Design, populate, and implement a data repository for coded hierarchical terminology describing replacement parts, service items and supplies commonly sold in the Automotive Aftermarket. The design of the database will support terminology for many industry segments, including Collision Repair, and will support classification of application-specific as well as generic commodities found in the Aftermarket. PCdb codes are central to the exchange of electronic catalog application data and serve to classify and categorize application records. The design of the database allows the hierarchy to be integrated into other automotive industry efforts such as the Product Information Exchange Standard (PIES).

### 3.2 PCdb Structure

The Parts Classification Database is a normalized relational database. While each item in the PCdb hierarchy follows a single path from Category and SubCategory, there are no limitations issued by the designers on how PartTerminology (xml attribute:PartType) codes are classified. Trading Partners may agree on other, or multiple, Category and SubCategory classifications for a particular PartTerminology (xml attribute:PartType). In any case, only the lowest level will be delivered with an application.

### 3.3 Positions

A “position” describes where a part is installed on the vehicle beyond what can be determined by the PartTerminology (PartType) alone when there is more than one of the part type on the vehicle. If there is only one quantity available on the vehicle, no position is necessary. Directional type positions are from a

forward facing reference. A “Position” table in the PCdb is used to hold all positions and also to provide a unique ID value for delivering position information on an application.

### 3.4 Valid Positions

The PCdb also includes valid vehicle “positions” where a part may be installed. Each PartTerminology (PartType) contains its own list of valid positions. A special “N/A” position is used to indicate that a position is not required for that PartTerminology (PartType). A part type may have both valid positions and the “N/A” position defined. In this case a Position tag is not required, but if a Position tag is included, it must match one of the valid positions. The “Differential” part type, for example has “Front”, “Rear” and “N/A” positions. If the vehicle has just one differential, no position is necessary. Otherwise, you must indicate which differential you are replacing.

### 3.5 Multifunction Parts

A Multifunction part is a single physical item that is used in more than one way on one or more vehicles. It is important to classify Multifunction Parts exactly as they are used on a vehicle. The use of the part on the vehicle is how the search will be done. Therefore, the PCdb must include all of the uses of a part in all vehicle applications.

#### Part to Use Table

To manage Multifunction Parts in the PCdb a PartsToUse table has been added to indicate whether each Part Terminology is appropriate for an ACES application file, a PIES Product Information file or both. The PartsToUse table joins Part Terminology ID’s with Use codes from the Use table in valid combinations

- Use ID1 indicates the Terminology is appropriate for an ACES application file
- Use ID 2 indicates the Terminology is appropriate for a PIES product information file

### 3.6 PCdb Change Management

The PCdb will be updated as often as bi-monthly in response to petitions from the industry for changes and additions. In addition to a fully refreshed dataset, the Auto Care Association will publish documentation of all changes made to the PCdb. **All petitions will be accepted unless the petition duplicates an existing position.** This includes somewhat non-standard “positions” such as “AC to Alternator” for belt routing. This policy allows standardized positions and keeps positions out of free-form notes.

The delivery XML schema includes a header tag (PcdbVersionDate) to designate which version of the PCdb was used to code PartTerminology (PartType) and Position information.

## 4 Qualifier database (Qdb)

A Qualifier database is supplied to help standardize terminology within the industry and reduce the number of potentially confusing free-text expressions found in applications. Qualifiers not covered by a vehicle attribute will be contained in the Qdb. In general, only individual qualifiers are stored in the table. You can create compound expressions by joining multiple entries from this table. A placeholder (i.e. parameter) will replace variable data found in a qualifier. This mechanism greatly reduces the number of qualifiers stored in the table.

### 4.1 Reasons for having “Coded Qualifiers”

**Faster selection of the correct part at the counter.** If the terminology is standardized and the presentation of complex qualifiers is also standardized, there should be less training and experience required to understand and use the catalog.

**More sophisticated data management opportunities.** If all data is coded, it should be easier to match information from different sources. This might include OE (original equipment) data and VIO (vehicles in operation) data, for example. If you can match this information accurately, you should be able to make better product management decisions.

**Centralized language translation.** If all qualifiers are coded, it helps facilitate a centralized translation effort. Without a centralized Qdb, each sender would need to translate qualifier data independently.

### 4.2 Qdb Structure

The Qualifier Table contains six fields: QualifierID, QualifierText, ExampleText, QualifierTypeID, NewQualifierID, WhenModified. The QualifierID is a unique primary key for this table. The QualifierText field contains the actual qualifier expression. The QualifierTypeID field is used to segment the table entries for special use (some examples are “Fitment”, “OE Only”, “Product Information” and “Installation Instruction”). The ExampleText field is used to provide assistance in selecting the correct qualifier and is especially helpful for qualifiers that have parameters. The NewQualifierID is used to supersede the current qualifier to another. Qualifiers with a NewQualifierID should not be used on new applications. Qualifiers with a NewQualifierID of “-1” should not be used and represent administrative or technical errors. The WhenModified field contains the date when the record was added or last changed.

### 4.3 Qdb Version Control

The Qdb includes a Version table and a QdbChanges table (similar to the VCdb). Each release of the Qdb posts a new release date in the Version table and includes a list of changes to the qualifiers in the QdbChanges table.

Any changes to an existing qualifier definition (except simple spelling, punctuation or word choice that does not affect the current meaning) will result in a new QualifierID being assigned. The NewQualifierID field will contain a reference to the new Qualifier record.

The delivery XML schema includes a header tag (QdbVersionDate) to designate which version of the QDB was used to code the qualifier data.

### 4.4 Qdb Parameters

Qualifiers found in applications often contain common strings but may differ in small ways. For example, dates or serial numbers are often used to identify mid-year production changes (“From 2/20/98”, “To Chassis #92817820”).

Rather than create a new Qdb entry for every date or chassis number that might be needed, a “parameter” is used. Instead of “From 12/1/2002” for example, the entry would include just the static text (“From”) and leave a placeholder for the actual date.

All parameters found in the English (template) expression must be uniquely identified to support language translation. (It may be incorrect to simply rearrange the translation and then substitute parameters by the order they are found. The first parameter in English might become the second parameter in another language.) By convention, the first parameter tag in the English qualifier expression will be **p1**, the second **p2**, and so on.

Standard XML coding is used to define qualifier parameters. This provides flexibility to add validation or other features in the future (as with the “min” and “max” attributes, for example). The following table shows several qualifier examples that might be found in the Qdb:

QualifierID	QualifierText
1	With Air Conditioning
2	From <p1 type="date"/> To <p2 type="date"/>
3	Except <p1 type="size"/> Wheels

A parameter may also represent a list of related values. (For example 60, 70, 80 Amp Alternator, where “60, 70, 80” is the one parameter to the Qdb template: “<p1 type="idlist"/> Amp Alternator”)

## 4.5 Qdb Parameter Types

The following table lists the available parameter types along with a description. These types are used in the parameter tags found in the Qualifier table (e.g. <p1 type="num" min="0" max="100"/>).

Type	Arguments	Description
<b>date</b>		A valid date in “m/d/y” or “m/y” format
<b>idlist</b>		One or more identifying strings separated by a comma
<b>name</b>		A company name, company brand, or geographic place.
<b>num</b>	<i>min, max</i>	A single whole, decimal or fractional number. “Shilling fractions” can be used (“2-1/2”), but symbol (case) fractions (“2½”) should not. It is left to the receiver to change presentation if desired.
<b>part</b>		A related part number that must be mentioned with the application
<b>phone</b>		Phone number
<b>size</b>		A single whole, decimal or fractional number. A linear unit of measure (uom) is required when delivering size values. Valid uom strings include “in”, “ft”, “mm”, and “cm”. “Shilling fractions” can be used (“2-1/2”), but symbol (case) fractions (“2½”) should not. It is left to the receiver to change presentation if desired.
<b>clock</b>		Used to indicate relative positions on a clock. For example “2 o'clock”. Valid values are whole numbers between 1 and 12.
<b>type</b>		Similar to “name”, but specifically referring to a well-known term for a component that has become generic.
<b>weight</b>		A single whole or decimal number. A weight unit of measure (uom) is required when delivering weight values. Valid uom strings include “mg”, “g”, “kg”, “oz”, “lb”, and “ton”.

## 4.6 Qualifier Language Translation

Since many suppliers must deliver Canada (French) and Mexico (Spanish) catalog data, we need to provide support for non-English translations for each qualifier expression. Complicating matters is the fact

that some languages require phrases to be rearranged so they can be expressed correctly. In addition, the formatting of certain parameters such as **dates** and **decimal numbers** may need to be changed according to the rules of the target language.

If a receiver of a file wants to translate a qualifier to French for the Canadian market, they might look up the French version of QualifierID 2 above in a translation table and find:

À Partir De <p1 type="date"/> Jusqu'à <p2 type="date"/>

They could then make the required substitutions to generate the following French translation:

À Partir De 1980-05-01 Jusqu'à 1980-10-01

Notice the date formats were localized properly for the target language. This translation method will work as long as a standard date format is used for English and the parameter values themselves do not need to be translated. When substituting parameter data, the parameter positions must match the parameter numbers in the qualifier expression (p2 may actually appear before p1 in the translated text). See [Section 5 – Application Data Transfer File below](#) for more information.

## 4.7 Qdb Data Rules

The following rules are used to control the inclusion of listings in the Qdb. These rules should be followed when submitting changes to the Qdb through [autocarevip.com](http://autocarevip.com).

1. **Use Title Case.** The first letter of each word should be capitalized. All upper case qualifiers are not acceptable.
2. **No Duplicates.** Do not allow duplicate qualifiers. Duplicates would include qualifiers where the wording is similar, but the meaning is the same.
3. **Use only approved abbreviations.** A list of “industry approved” abbreviations will be published. As a rule, abbreviations are limited to generally accepted terminology where common use and need for brevity are obvious. Examples include “cm” for centimeters and “GVW” for Gross Vehicle Weight. All other terminology should be spelled out completely. OE specific abbreviations (ASR, E.E.C., RPO Codes) should be defined in the qualifier as well as abbreviated in parenthesis.
4. **No embedded VCdb Attributes.** A vehicle attribute should not be included in the qualifier expression (even if used with Except logic) except as noted below. For example, “(Exc. Manual Transmission w/Air Conditioning)” should be delivered as two separate applications: (1) <TransmissionControlType id="5">Automatic</TransmissionControlType>, and (2) <Qual

id="18067">Without Air Conditioning</Qual>. The one exception is if an attribute is excepted in an installation instruction or product note. For example, “Valve Cover Set Not Included For CRX HF.” Otherwise, we would be forced to write the application at a very low level (the Submodel in this case), and that goes against the basic idea of delivering data at the highest level possible.

5. **Avoid OE terminology used in place of vehicle attributes.** OE specific terminology is allowed, but not if it means the same thing as an existing vehicle attribute. For example, use the AWD attribute instead of “4-Matic”.
6. **Avoid ANDED "compound" expressions.** Break unrelated compound expressions into separate qualifiers wherever possible. Closely related qualifiers may be kept together. (For example, “To <p1 type="date"/> & Serial # <p2 type="id"/>, or “<p1 type="num"/> Blade, without Leads” are both acceptable). Expressions with “Except Logic” may contain compound qualifiers (there may not be enough information available to translate it to positive logic). For example, “Except 14” Wheels with Traction Control” is allowed. Parenthetical information that further explains the qualifier may also be included in an expression.
7. **Compound expressions with OR logic is allowed.** Since two or more qualifiers on an application are always interpreted to be joined by an AND, the only way to indicate an OR condition is to break it up into multiple applications. This separation may not be the best way to display the information and so compound expression with OR conditions are allowed. (For example, “Police or Taxi”, “Dual Rear Wheels or Heavy Duty Suspension”).
8. **No Positions.** Qualifiers should not include position information. All positions are coded in the Position tag, even if the valid position for a PartTerminology (xml attribute: PartType) is unknown or un-researched (i.e. N/A, N/R, U/K).
9. **Avoid Part Numbers.** Where possible, qualifiers should not include part numbers. If you must include a part number because it is important to mention a related part for an application, make sure the related part also has its own application, and create a “part” parameter for the part. For example: “A<p1 type="part"/> adapter is required when replacing the widget.”
10. **Don't over use parameters.** Only use parameters for information that is truly variable and generally application specific. (For example, don't substitute a parameter for the “1” in “1st Design”). Do not use a parameter if the value will never change (in context with the rest of the qualifier).

11. **Avoid over use of optional plural.** Don't use optional plural on end of terms that must be plural as used. (<p1 type="num"> Degrees, not <p1 type="num"> Degree(s) unless you expect "1" to be a possible value.)
12. **Consistent use of quoted terms.** Enclose stamping/markings or other identification strings in quotes to distinguish them from abbreviations.
13. **Rules for Quantities.** Use spelled out numbers for quantities in ten or less. "with Leaf Spring Type Clutch with **three** Mounting Holes" instead of "with Leaf Spring Type Clutch with (3) Mounting Holes".
14. **No special characters.** Do not include special characters above 128 ASCII including a degree mark, plus/minus sign or registration mark.

## 4.8 Qdb Review Process

The Qdb will be updated as often as daily in response to petitions from the industry for changes and additions. All petitions following the rules will be accepted. Existing Qdb entries will never be deleted, but could be "superseded" to another (duplicate) entry. Word changes to an existing entry will never be allowed to change meaning.

### Qdb Submissions

Qdb Submissions may be made one at a time or in batch by uploading a file. Both submissions must be made through the autocarevip.com website interface.

### Qdb Bulk Submission Format

Bulk Qdb submission files must follow a pre-defined format to be accepted. Required fields (separated by "~");

action~source~notes~qualifier id~qualifiertext~exampletext~type~customID

### Sample Text file Contents

MODIFY~source~notes~19701~qualifiertext~exampletext~Fitment~Custom ID

ADD~source 1~notes 1~0~qualifiertext 1~exampletext 1~Fitment~Custom ID

## 5 Application Data Transfer File

The Application Data Transfer File ("Transfer File") is the mechanism used to transmit new (or changed) catalog records between data partners. Its physical characteristics, format and content are described in this section.

### 5.1 Transfer File Purpose

The purpose of the Transfer File is to provide a predictable record format and syntax that can be widely used to exchange application-driven data coded to the relational databases, collectively referred to as the Aftermarket Catalog Exchange Standards (ACES). By publishing a recommended data format, data partners can devote more resources to quality assurance and presentation of the data, and fewer resources to communication of the data.

This record format must not impose a technology requirement for either party beyond what is commonly used in the industry today. The format should impose rules and minimum requirements that reflect the needs of all major data partners. However, the format should be easily customized to meet specific data partner requirements.

### 5.2 Physical Characteristics

The Transfer File must include only plain text delimited by "valid" XML tags (as defined below). A CR/LF (ASCII 13,10 decimal) should separate lines and the text should use UTF-8 character encoding. An example of this format is included in Appendix B.

### 5.3 Transfer File Naming Convention

The file name should include the company, a short catalog title, the delivery date (yyyy-mm-dd) and the word "FULL", "UPDATE", or "TEST" all separated by underlines and an "xml" extension. For example:

ACMESupply\_BrakeHardware\_2001-01-18\_FULL.xml

When delivering one or more Transfer Files, use an archiving program (such as WinZip, or Windows XP "Compressed (zipped) Folders") to compress the files into a single ZIP archive. Name the archive as above but without the catalog title and with a ".zip" extension. (**Note**, to remain compatible with older versions, use only classic pkzip "deflate" format, not PPMd or bzip2 formats now available in WinZip 10 and later).

## 5.4 Delivering Changes Only

As indicated in Section [5.3](#), there are two ways to deliver (non-test) application data (1) all applications found in the catalog ("FULL"), and (2) only changes made to the catalog since the last delivery ("UPDATE"). The "action" attribute is used to indicate changes. All records in the initial FULL load of a catalog should be coded with an "A". If desired, after the initial FULL load with a trading partner, updates can be sent instead of the entire file. When delivering updates, only transmit applications that are new or have changed in some way. Changes are accomplished with a pair of "D" (delete) and "A" (add) records. The "D" record should match the existing record as originally delivered, and the corresponding "A" record should contain the information as you now want it stored.

If you originally delivered a year-range application and you now want to change a single year of that range, it is acceptable to delete one year using the old information and add the updated information for that year.

## 5.5 ACES Root Element

XML documents must contain a single begin and end tag to define the “root” element. All other elements must be nested within this root element. The **ACES** tag is the root element for ACES data files and contains a single “version” attribute. There are currently four valid values for the version attribute: “2.0”, “3.0”, “3.0.1”, 3.1”, “3.2” or “4.0”. ACES 2006 formatted files should use the 2.0 value and are known as ACES 2.0 formatted files.

```
<ACES version="2.0"> <!-- ACES 2006 formatted files -->
```

```
...
```

```
</ACES>
```

All versions are included in the current specification and XML schema files. Receivers of ACES data should check the version number to determine how to interpret the file; however, each schema version is designed to be backward compatible.

## 5.6 Header Information

Information is included at the beginning of the Transfer File to describe what is being delivered as well as the company and contact information. See the valid header tags in section [5.11](#) below. The order of the tags within the header is defined by the XML schema. (See section [6.7](#) for more information on XML element ordering). All “required” header tags must be included.

## 5.7 Applications

Catalog data should only include the information necessary to unambiguously select the correct part. The following information is required for each application: (1) Vehicle Identification, (2) Part Type, (3) Part Number, and (4) Part Quantity. Optionally, you may include (5) Vehicle Attributes, (6) Comments, and (7) Position information as necessary.

### Vehicle Identification

Either a Base Vehicle (which by definition includes a year) or a Make / Year-Range combination must be included with each application. (In practice, Make / Year-Range by itself is not very useful without adding optional Vehicle Attributes such as Model, Engine or Transmission.) Some data receivers may not accept both methods or may require one over the other.

In ACES 4.0 In addition to vehicle applications, equipment content can also be sent. Either an Equipment Base or a Manufacturer, Equipment Model and Vehicle Type must be sent with each equipment application.

Check with your trading partner before sending application data.

### Part Type (Part Terminology)

A valid Part Terminology ID, the lowest level of the Parts Classification Database (PCdb), must be included with each application. Position information, if supplied, will be validated against this ID.

### Part Number

A single part number, formatted for end-user viewing, should be included with each application. Do not include parenthetical information (e.g. quantity, position or footnote information) with the part number (this information is handled separately).

### Part Quantity

All applications must contain a “Per Car Quantity” that reflects the appropriate number of parts needed for the given part number, part type and vehicle.

If the application (or part description) contains Position information (e.g. Brake Hose-Left Rear) the “Per Car Quantity” should reflect the number for that specific Position (in this case “1”). If the application does not contain *complete* Position information for the vehicle, but instead refers to multiple Positions, the “Per Car Quantity” should reflect the total quantity associated with that part number. For instance, if the vehicle has two “Brake Hose-Rear” applications that use the same part number, then the quantity would be “2”.

(In this case, “Brake Hose-Rear” might be used to mean both Left Rear and Right Rear). See example below.

Description	Part Number	Per Car Quantity
Brake Hose-Left Rear	1234	1
Brake Hose-Right Rear	1234	1
Brake Hose-Rear	1234	2

For parts that are typically purchased one at a time and cannot be validated using the Valid Vehicle Table (e.g., Wheel Bolts or Lug Nuts) the “Per Car Quantity” should be “1”. It is very unlikely that someone will want to purchase a full set of wheel bolts or lug nuts.

Consideration should be given to the “Per Car Quantity” for sets. For instance, a vehicle may have two Valve Cover Gaskets but if they are packaged as a “set” then the “Per Car Quantity” should be “1”.

### Vehicle Attributes

All ID fields that make up the vehicle or engine configuration (as defined in the VCdb) may be used to further qualify (i.e., limit the scope of) an application. These vehicle attributes are references to an identifier (primary key) in a related table. For example, ID 10 in the BodyType table represents a “Convertible”. Since only the ID is used, there is no concept of “Except” when using these vehicle attribute fields. (See Section [5.7.6](#))

One or more of these “vehicle attribute” tags may be used on a single application. If more than one of these tags is used in an application, however, they will be interpreted to be joined with a logical “AND.” As a result of this rule, it is not valid to use two of the same vehicle attribute tags in an application (e.g., a vehicle cannot be both a Coupe and a Sedan).

Special codes (also called “Abbreviations” in the VCdb) are used as attribute values to indicate missing data. Vehicle attribute IDs containing one of these abbreviations should **not be used** when delivering applications.

Abbreviation	Description	Explanation
-	Intentionally blank.	Indicates the attribute was intentionally left blank because it is both VCdb and Legacy, and the Legacy value was blank. Also used for
N/A	Not Available	Indicates the attribute was researched, but no information is available at this time.
N/R	Not Required	Indicates the attribute is not required for the vehicle.
U/K	Unknown	Indicates the attribute has not been researched yet.

## Except Logic

As mentioned above, there is no direct support for Except Logic with Vehicle Attributes. Instead, you must translate the Excepted Expression into positive logic. This is possible in most cases because the VCdb supplies the complete list of valid attributes for a vehicle.

## Excepted Attributes when Values are Not Known

Vehicles before 1975 and vehicles with incomplete information (N/A or U/K values), however, will not have a complete list of valid attributes. In these cases, you should include the **Excepted Expression** as a Note (e.g., <Note>Exc. Sedan</Note>).

## Qualifiers [ACES 2.0+]

Any qualifiers needed for an application should be delivered using one or more **Qual** tags. The Qual tag (1) references a primary key in the Qualifier table, (2) provides optional **param** elements as necessary, and (3) shows the expanded qualifier expression in a required **text** element. The following is an example of a simple qualifier referencing a Qdb entry (123, "With Air Conditioning").

```
Qual id="123">
<text>With Air Conditioning</text>
</Qual>
```

The text element is required to help with early adoption of coded qualifiers. All qualifiers on an application must be coded. The Note tag is only to be used for Excepted attribute strings on vehicles where the attributes are not researched or are unknown.

Use a **param** tag to deliver parameter values for qualifiers that need them. The order of the param tag is important and should match the order implied by the numbered parameters. For example:

**Qdb:** 1797, Camber Adjustment From <p1 type="num"/> to <p2 type="num"/> Degrees

```
xml: <Qual id="1797">
      <param value="1"/>    <-- 1st parameter -->
      <param value="2"/>    <-- 2nd parameter -->
      <text>Camber Adjustment From 1 to 2 Degrees</text>
</Qual>
```

## Decimal and Fractional Parameters

Certain parameter types (e.g., “num”, “size” and “weight”) may include decimal or fractional values. The following table shows several common values in both fractional and decimal form:

Value	Fractional	Decimal
Three Quarters	3/4	0.75
One and One-Half	1-1/2	1.5
Nine Sixteenths	9/16	0.5625

The parameter concept allows you to deliver the number in either format as long as mixed fractions are delivered in a standard way (i.e. #-#/#). For example,

```
<param value="1-1/2"/>
```

With these conventions, the receiver can check for a valid value and perform format conversions as necessary. In most cases, the data will be displayed exactly as received (as a fraction or a decimal), so you should use the format you would most like to see published.

### Unit of Measure for Parameters

Certain parameter types (e.g. “size” and “weight”) must include a “unit of measure” (uom) value along with their actual value. For example,

**Qdb:** 100, With <p1 type="size"/> Diameter Alternator Case

```
xml: <Qual id="100">
      <param value="4-3/4" uom="in"/>
      <text>With 4-3/4" Diameter Alternator Case</text>
    </Qual>
```

The **uom** attribute of the param tag tells the receiver what measurement the value represents. There is a small list of valid values for the uom attribute. For example: “in”, “mm”, “lb”, “kg”, “g”. (See *the table in Section 4.5*).

It is important to remember that the display of the qualifier is up to the “presenter.” They could choose to display “With 4.75 inch Diameter Alternator Case”, for example.

### Alternate Values for Parameters

If you want to deliver two representations for the same value (in different units), use the **altvalue** and **altuom** attributes of the param tag.

```
xml: <param value="4-3/4" uom="in" altvalue="120" altuom="mm"/>
```

It would then be up to the presenter to put the alternate value in parenthesis (or in some other form to indicate the equivalent value).

### Representing Complex Expressions

When translating paper catalogs to electronic form, it's common to find applications with complex qualifier expressions (possibly created to save space). Here are a few example expressions that include OR logic:

1. 4WD Japan Built or RWD
2. Convertible or Sedan w/Disc Brakes
3. 8" Diameter Booster w/ABS or 10" Diameter Booster

Since Vehicle Attribute/Qualifier tags on an application are combined logically with “AND”, we need a way to represent “OR”. The solution is to create separate applications. Here is the coding for the first expression shown above. Notice two applications were created. (Some required tags are not shown).

<pre>&lt;App action="A" id="1" ref="1"&gt;   &lt;DriveType id="8"&gt;4WD&lt;/DriveType&gt;   &lt;Note&gt;Japan Built&lt;/Note&gt; &lt;/App&gt;</pre>	<pre>&lt;App action="A" id="2" ref="1"&gt;   &lt;DriveType id="7"&gt;RWD&lt;/DriveType&gt; &lt;/App&gt;</pre>
--	---

The second example is like the first, except you need to interpret what it means. It could mean the part fits all Convertibles as well as Sedans with Disc Brakes. Or it could mean it fits all Convertibles with Disc Brakes as well as all Sedans with Disc Brakes. A further complication is the need to designate the correct front or rear brake tag (<FrontBrakeType id="5"/>). A similar problem is found with 2 and 4 wheel ABS. The third example shown above would produce the following three applications (assuming the vehicle came with both 2/4 ABS):

<pre>&lt;App action="A" id="1" ref="3"&gt;   &lt;BrakeABS id="7"&gt;2-Wheel ABS&lt;/BrakeABS&gt;   &lt;Note&gt;8" Diameter Booster&lt;/Note&gt; &lt;/App&gt;</pre>	<pre>&lt;App action="A" id="2" ref="3"&gt;   &lt;BrakeABS id="8"&gt;4-Wheel ABS&lt;/BrakeABS&gt;   &lt;Note&gt;8" Diameter Booster&lt;/Note&gt; &lt;/App&gt;</pre>
<pre>&lt;App action="A" id="3" ref="3"&gt;   &lt;Note&gt;10" Diameter Booster&lt;/Note&gt; &lt;/App&gt;</pre>	

Applications written with “Except” logic can be particularly challenging since Except is not supported on vehicle attributes. (Except can be included in Vehicle Qualifier text, however). The following rules (known as DeMorgan’s Laws) can be used to translate Except logic in your applications.

Exc. (A and B) ↔ Exc. A or Exc. B

Exc. (A or B) ↔ Exc. A and Exc. B

An understanding of these rules is necessary because parentheses are often implied in many real-world expressions (and since this kind of grouping is not supported in the delivery of applications). These rules can be used to put applications in the form: (A and B) or (C and D) or (E and F). It is then a simple matter to split the ORed applications as required (in this case, into three separate applications).

### Comments

Any additional information not covered by a vehicle attribute or qualifier tag should be included in a *Note* tag. The Note tag contains free-form text and should be used primarily for informational comments (i.e. comments not used to determine which part fits which vehicle). Normally, this text would contain information the manufacturer wants the catalog reader to be aware of when ordering a part. An example of this type of comment might be an installation instruction such as “Coil wire modification required. Splice in original connector.” (See the <Note> tag in Section [5.12](#) for more information).

It is preferred that logically distinct notes be placed in separate <Note> tags for an application (multiple note tags are allowed in a single application). It should be understood, however, that these notes would most likely be appended together and displayed without further modification. An optional ID may be included as an attribute of the note tag to reference an external comment file. Since this external comment file must be agreed upon between trading partners, this method should be considered less universal.

### Position

Position is used to indicate where the part should be applied to the vehicle. The position ID is validated against the Part Terminology used on the application (by the CodeMaster table in the PCdb) and may only be used if valid on that Part Terminology. (See Sections [3.3](#) & [3.4](#) for more information).

### MfrLabel

MfrLabel is an optional text field that permits a manufacturer to specify the part label used to describe their specific parts. The label is a short description of the specific part or a manufacturer specified part differentiation. It is also useful when the part description is **more specific** than the Part Terminology being used.

Example: Platinum and Copper core spark plugs are provided in the same Part Terminology. The manufacturer again wishes to differentiate these parts by using a specific part label.

### **DisplayOrder**

DisplayOrder is an optional field that permits a manufacturer to specify the order in which parts are presented for display in an electronic catalog. The order value is used when multiple parts are available for a specific application within a part type.

Example 1: A Sparkplug wire manufacturer offers 3 different wire sets for each application. They wish to dictate that the premium wire should always display first, followed by the better grade, and finally the standard grade.

Example 2: A tailpipe for a specific vehicle is composed of multiple pieces. The manufacturer wishes to display the parts in order from Front-to-Rear to avoid confusion.

## **5.8 Digital Assets**

ACES supports looking up *any type* of content, not just part numbers. This content could include *application-specific* digital assets such as diagrams (exhaust, belts, suspension, etc.), technical bulletins, installation instructions; vehicle images, etc. (Note that part-specific digital asset links are defined using PIES). See Appendix D for further information and examples. As of this release, 3 types of asset references are defined: Application, Vehicle and Digital File Information.

### **Logical Asset Identifier (AssetName Element)**

Key to delivering application or vehicle level digital assets is the concept of a logical asset identifier. A manufacturer will often have several “physical” files (different formats, resolutions, URL’s, etc.) representing a single digital asset for the App or Asset. A key reference to these files, as defined in the DigitalFileInformation element, is represented by a “logical” name for each file. The AssetName element references this logical name and is valid in the App and Asset elements. Each physical file intended to be referenced by the App or Asset element containing an AssetName element should be represented in a DigitalFileInformation element with the same AssetName attribute.

### **Application Level Assets (within the App Element)**

The delivery of an application-specific digital asset is optional to deliver a single application with the use of the AssetName element.

### Vehicle Level Assets (within the Asset Element)

A digital asset may be vehicle specific as opposed to a part number specific. This element is identical to the App element but without the Part, Part Type, Position, Qty, DisplayOrder, AssetItemOrder and AssetItemRef elements.

### Stand Alone Digital Assets Metadata (DigitalAsset Element)

In addition to supplying relational information between an Asset and a vehicle or application, the metadata of the actual asset is needed when providing digital assets within an ACES file to help with easy consumption of the data by data receivers. Within the DigitalAsset container includes the DigitalFileInformation element. The DigitalFileInformation element is comprised of sub-elements that further define the actual asset. The AssetName is an attribute of the DigitalFileInformation (and is required) that will provide the key relationship to the asset's use via the AssetName element in the Application (App element) or Vehicle assets (Asset element). Each physical file that needs to be referenced by an App element or Asset element should be delivered in a DigitalFileInformation element with the same AssetName attribute.

### Identifying Asset Elements

While the part number is the focus of an application record. When identifying an item in a diagram, additional App elements available to help indicate *portions* of a digital asset: AssetItemOrder, AssetItemRef.

- **AssetItemOrder** is optional to show relative position (usually front-to-rear, and when applicable left-to-right.).
- **AssetItemRef** is optional to use when Part and AssetItemOrder don't apply. AssetItemRef is used to ensure proper identification within the image, but how this identifier may be used is left to the receiver.

## 5.9 Footer Information

Footer Information is included in the Transfer File to indicate the end of the data and provide a count of the applications transferred.

## 5.10 General Rules

1. Catalog applications should be able to "stand alone." Context cannot be guaranteed when applications are displayed in an electronic catalog.

2. Only include information that is necessary to determine the proper part to use.

## 5.11 Delivering Invalid Applications

While it is possible to deliver applications with vehicle and attribute combinations not found in the VehicleTo configuration tables, this is usually not a good practice and most likely would result in those applications being removed by the receiving trading partner. An exception is made for applications with “un-researched” attributes (N/A, U/K). **These attributes should be delivered using normal coded tags.** Applications that are unable to be represented by identifiers in the VCdb should not (and indeed cannot) be delivered. (See Appendix C for more information on this topic.)

A method for subscribers to petition for changes and additions to the VCdb is available at: [autocarevip.com](http://autocarevip.com).

To reduce confusion with the following abbreviations N/R, N/A, and N/S, definitions have been provided.

### **N/R – Not Required**

A data supplier may feel the need to send an N/R when a vehicle does not contain a part applicable to their product line or the vehicles covered within their data set. If the vehicle did not have a certain part, there is no need to send any information for this vehicle.

If a part is not required on a vehicle, (it did not come with it), a customer will not look to replace the nonexistent part on their vehicle, and therefore it does not need to be in the catalogue. The industry has discussed Negative Fitments, and does not wish to pursue this approach. This would require significantly more records to represent all the non-applicable combinations of car parts to cars. This exercise would not provide value for members.

### **N/A - Not Applicable**

See N/R

### **N/A – Not Available**

If a data supplier does not currently have a part number for an application, but does have application information for a new part number yet to be determined, the N/A should **NOT** be used as a place holder for future applications.

A data supplier should not send data until the valid information can be provided.

## **N/S – Not Serviceable or Non-Serviceable**

This can be used in instances in which a part does exist on a vehicle but due to vehicle configuration the part cannot be replaced. It is not intended to be used if a data supplier does not make a replacement part for a specific application. In most instances these applications are identified by the OE as Non-Serviceable.

Data can be provided for applications in which an OE Manufacture may list an application as N/S however a data supplier has come up with a Solution to provide a replacement part for this application.

The key will be to ensure the fitment note specifies replacing this part may require 'Modification to the Original Vehicle Configuration'.

## **5.12 XML Elements**

This section defines the valid XML Elements that may be used in the Transfer File (in alphabetical order).

**XML Element** lists the XML name for this field per the ACES XML Schema. When used within an ACES file they will include an opening and closing tags, Example would be <App></App>. Note that the second tag includes “/” which comes before the name, this denotes that this is the closing element.

**XML Attribute** lists the XML attribute name(s) for the field per the ACES XML Schema.

**Segment** lists the specific area within the XML the element is used.

**Requirement** is part of the Segment column there are two values: req = required, opt = optional.

**Description** will contain general descriptive information about each element.

The **Example XML Code** cell will show data value samples from most fields.

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
ACES	version	Structure <i>req</i>	Root element. Top level document tag. The version attribute should contain the ACES version number (currently “2.0”, “3.0”, “3.0.1”, “3.1”, “3.2”, 4.0).	<pre>&lt;ACES version="4.0"&gt; &lt;Header&gt;...&lt;/Header&gt; &lt;App&gt;...&lt;/App&gt; &lt;Asset&gt;...&lt;/Asset&gt; &lt;DigitalAsset&gt;...&lt;/DigitalAsset&gt; &lt;Footer&gt;...&lt;/Footer&gt; &lt;/ACES&gt;</pre>
App	action, id, ref, validate	Structure <i>opt</i>	Groups information that defines an application. One or more can be used in the file. The <b>action</b> attribute can be “A” or “D” (for add or delete). The <b>id</b> attribute should be a sequential number that uniquely identifies this application in the transfer file. Both the action and id attributes are required. The <b>ref</b> attribute can optionally be used to reference a source record. The “validate” attribute defaults to “yes”. If “no”, the receiver will not validate the application against the VehicleTo tables.	<pre>&lt;App action="A" id="234"&gt; ... &lt;/App&gt;</pre>
ApprovedFor	<i>None</i>	Header <i>opt</i>	Groups information that defines which countries applications are approved for.	<pre>&lt;ApprovedFor&gt; &lt;Country&gt;US&lt;/Country&gt;</pre>
Aspiration	id	VehAttr <i>opt</i>	References the Aspiration table. AspirationID should be used for valid values of the id attribute.	<pre>&lt;Aspiration id="74"/&gt;</pre>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
Asset	action, id, ref, validate ACES 3.0.1	Structure <i>opt</i>	Groups information that defines an asset lookup. The <b>action</b> attribute can be “A” or “D” (for add or delete). The <b>id</b> attribute should be a sequential number that uniquely identifies this application in the transfer file. Both the action and id attributes are required. The <b>ref</b> attribute can optionally be used to reference a source record. The “validate” attribute defaults to “yes”. If “no”, the receiver will not validate the application against the VehicleTo tables.	<Asset action="A" id="234"> ... </Asset>
AssetDimensions	UOM	DigitalAsset <i>opt</i>	Groups Asset Dimensions together for height and width of the application-specific digital asset.	<AssetDimensions UOM="PX"> <AssetHeight>250</AssetHeight> <AssetWidth>250</AssetWidth> </AssetDimensions>
AssetDescription	None	DigitalAsset <i>opt</i>	Free text field to describe the application-specific digital asset.	<AssetDescription>High resolution image from brochure</AssetDescription>
AssetHeight	None	DigitalAsset <i>opt</i>	Vertical measurement of application specific digital asset file.	<AssetHeight>250</AssetHeight>
AssetItemOrder	None ACES 3.0	App <i>opt</i>	To support application-specific Digital Assets.	<AssetItemOrder>1</AssetItemOrder>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
AssetItemRef	None ACES 3.0	App <i>opt</i>	To support application-specific Digital Assets.	<AssetItemRef>A</AssetItemRef>
AssetName	None ACES 3.0	App Asset <i>opt</i>	To support application-specific Digital Assets.	<AssetName>A12345</AssetName>
AssetType	None	DigitalAsset <i>opt</i>	Code identifying the Asset Type of the application-specific Digital Asset.	<AssetType>BRO</AssetType>
AssetWidth	None	DigitalAsset <i>opt</i>	Horizontal measurement of application-specific digital asset file.	<AssetWidth>250</AssetWidth>
Background	None	DigitalAsset <i>opt</i>	Code identifying the Background of the application-specific Digital Asset.	<Background>WHI</Background>
BaseVehicle	id	VehAttr <i>opt</i>	References the BaseVehicle table. BaseVehicleID should be used for valid values of the id attribute. One of the following must be sent in the Vehicle Ident Group: a Base Vehicle ID; a Make / Year or Make / Year-Range combination must be included with each application; a Base Equipment ID; or a Mfr / Equipment Model / Vehicle Type	<BaseVehicle id="9946"/>  <!-- 2002 Ford Taurus -->
BedLength	id	VehAttr <i>opt</i>	References the BedLength table. BedLengthID should be used for valid values of the id attribute.	<BedLength id="3"/>  <!-- 72.0 inches -->
BedType	id	VehAttr <i>opt</i>	References the BedType table. BedTypeID should be used for valid values of the id attribute.	<BedType id="5"/>  <!-- Step Side -->

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
BodyNumDoors	id	VehAttr <i>opt</i>	References the BodyNumDoors table. BodyNumDoorsID should be used for valid values of the id attribute.	<BodyNumDoors id="1"/>  <!-- 2 Doors -->
BodyType	id	VehAttr <i>opt</i>	References the BodyType table. BodyTypeID should be used for valid values of the id attribute.	<BodyType id="4"/>  <!-- Sedan -->
BrakeABS	id	VehAttr <i>opt</i>	References BrakeABS table. BrakeABSID should be used for valid values of the id attribute.	<BrakeABS id="7"/>  <!-- 2-Wheel ABS -->
BrakeSystem	id	VehAttr <i>opt</i>	References the BrakeSystem table. BrakeSystemID should be used for valid values of the id attribute.	<BrakeSystem id="5"/>  <!-- Power -->
BrandAAIAID	None ACES 3.0	Header <i>opt</i>	The default brand for the file. This element is optional but strongly recommended. It is a 4 character reference to the "Brand Code" level of the AAIA Parent/Brand table.	<BrandAAIAID>BBBB</BrandAA IAID>
ColorMode	None	Asset <i>opt</i>	Code identifying the ColorMode of the application-specific Digital Asset.	<ColorMode>RGB</ColorMode>
Company	None	Header <i>req</i>	The company delivering the data.	<Company>ACME Mfg.</Company>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
Country	None	Header DigitalAsset <i>opt</i>	Child Element of ApprovedFor and DigitalFileInformation. ISO published a document 3166-1 that catalogs all the ISO “Country Codes.” Country Codes are 2-letters (alpha).  Example United States = US. The PIES standard uses this table and it is recommended that these codes be used here as well.	<Country>US</Country>
CylinderHeadType	id	VehAttr <i>opt</i>	References the CylinderHeadType table. CylinderHeadTypeID should be used for valid values of the id attribute.	<CylinderHeadType id="6"/>  <!-- DOHC -->
DigitalFileInformation	AssetName  LanguageCode	DigitalAsset <i>opt</i>	Groups information that defines the application-specific digital asset meta data. The <b>action</b> attribute can be “A” – Add, “D” – Delete. The <b>AssetName</b> attribute should be a group identifier for digital assets in the transfer file. Both the action and AssetName attributes are required. The languagecode attribute is used to reference the language of textual values. Values come from the ISO 639-1 table.	<DigitalFileInformation AssetName="234" action="A" LanguageCode="EN" > ... </DigitalFileInformation>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
DisplayOrder	None	App <i>opt</i>	Allows a manufacturer to specify the order in which parts are presented for display in an electronic catalog. The order value is used when multiple parts are available for a specific application within a part type.	<DisplayOrder>1</DisplayOrder>
DocFormNumber	None	Header <i>opt</i>	Catalog Form number	<DocFormNumber>F02991</DocFormNumber>
DocumentTitle	None	Header <i>req</i>	Catalog name or other identifying information	<DocumentTitle>ACME Brake Hardware</DocumentTitle>
DriveType	id	VehAttr <i>opt</i>	References the DriveType table. DriveTypeID should be used for valid values of the id attribute.	<DriveType id="5"/>  <!-- FWD -->
EffectiveDate	None	Header <i>req</i> DigitalAsset <i>opt</i>	The date shown on the catalog. Format is CCYY-MM-DD where "CC" represents the century, "YY" the year, "MM" the month and "DD" the day. Zero fill if necessary.	<EffectiveDate>2000-06-01</EffectiveDate>
EngineBase	id	VehAttr <i>opt</i>	References the EngineBase table. EngineBaseID should be used for valid values of the id attribute.	<EngineBase id="315"/>  <!-- V6 3.0L -->
EngineBlock	id	VehAttr <i>opt</i>	References the EngineBlock table. EngineBlockID should be used for valid values of the id attribute.	<EngineBlock id="117"/>

XML Element	XML Attribute	Segment Requireme	Description	Example XML Code
EngineBoreStroke	id	VehAttr  <i>opt</i>	References the EngineBoreStroke table. EngineBoreStrokeID should be used for valid values of the id attribute.	<EngineBoreStroke id="10"/>
EngineDesignation	id	VehAttr  <i>opt</i>	References the EngineDesignation table. EngineDesignationID should be used for valid values of the id attribute.	<EngineDesignation id="22"/>  <!-- 2TC -->
EngineMfr	id	VehAttr  <i>opt</i>	The manufacturer that actually built the engine. References the Mfr table. MfrID should be used for valid values of the id attribute.	<EngineMfr id="544"/>  <!-- Ford -->
EngineVersion	id	VehAttr  <i>opt</i>	References the EngineVersion table. EngineVersionID should be used for valid values of the id attribute.	<EngineVersion id="45"/>  <!-- Cleveland -->
EngineVIN	id	VehAttr  <i>opt</i>	References the EngineVIN table. EngineVINID should be used for valid values of the id attribute.	<EngineVIN id="15"/>  <!-- X -->
ExpirationDate	None	DigitalAsset  <i>opt</i>	The date shown on the catalog. Format is CCYY-MM-DD where "CC" represents the century, "YY" the year, "MM" the month and "DD" the day. Zero fill if necessary.	<ExpirationDate>2000-06-01</ExpirationDate>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
EquipmentBase	id	VehAttr <i>opt</i>	References the EquipmentBase table. EquipmentBaseID should be used for valid values of the id attribute. Either the EquipmentBase tag or the Mfr, EquipmentModel and VehicleType tags are required for each application.	<EquipmentBase id="101"/>
EquipmentModel	id	VehAttr <i>opt</i>	If a EquipmentBase tag is not used in an application, an EquipmentModel tag must be supplied.	<EquipmentModel id="42"/>
FileDateModified	None	DigitalAsset <i>opt</i>	The date shown on the catalog. Format is CCYY-MM-DD where "CC" represents the century, "YY" the year, "MM" the month and "DD" the day. Zero fill if necessary.	<FileDateModified>2000-06-01</FileDateModified>
FileName	None	DigitalAsset <i>opt</i>	File name (including file extension) of application-specific digital asset.	<FileName>xyz.jpg</FileName>
FilePath	None	DigitalAsset <i>opt</i>	Location of file in application-specific digital asset collection provided by Supplier. Path should be identified from the Root level (\). Generally, the collection refers to a CD/DVD/Archive File which contains multiple Digital Assets.	<FilePath>\Mfg\xyz.jpg</FilePath>
FileSize	None	DigitalAsset <i>opt</i>	File size of application-specific digital asset as measured in Kilobytes (kb).	<FileSize>123456</FileSize>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
FileType	None	DigitalAsset <i>opt</i>	Code identifying the File Type and purpose of the application-specific digital asset.	<FileType>JPG</FileType>
Footer	None	Structure <i>req</i>	Occurs after the last application tag in the file. Used as a container for footer tags.	<Footer> ... </Footer>
FrontBrakeType	id	VehAttr <i>opt</i>	The brake type used on the front wheels. References the BrakeType table.	<FrontBrakeType id="1"/> <!-- Disc -->
FrontSpringType	id	VehAttr <i>opt</i>	The basic suspension type used in the front of the vehicle. References the SpringType table. SpringTypeID should be used for valid values of the id attribute.	<FrontSpringType id="45"/> <!-- Coil -->
FuelDeliverySubType	id	VehAttr <i>opt</i>	References the FuelDeliverySubType table. FuelDeliverySubTypeID should be used for valid values of the id attribute.	<FuelDeliverySubType id="6"/> <!-- 2BBL -->
FuelDeliveryType	id	VehAttr <i>opt</i>	References the FuelDeliveryType table. FuelDeliveryTypeID should be used for valid values of the id attribute.	<FuelDeliveryType id="4"/> <!-- Carb -->
FuelSystemControlType	id	VehAttr <i>opt</i>	References the FuelSystemControlType table. FuelSystemControlTypeID should be used for valid values of the id attribute.	<FuelSystemControlType id="6"/> <!-- Mechanical -->

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
FuelSystemDesign	id	VehAttr <i>opt</i>	References the FuelSystemDesign table. FuelSystemDesignID should be used for valid values of the id attribute.	<FuelSystemDesign id="47"/>  <!-- Carter -->
FuelType	id	VehAttr <i>opt</i>	References the FuelType table. FuelTypeID should be used for valid values of the id attribute.	<FuelType id="84"/>  <!-- Gas -->
Header	<i>None</i>	Structure <i>req</i>	Occurs at the beginning of the file before any application tags. Used as a container for header tags.	<Header> ...  </Header>
IgnitionSystemType	id	VehAttr <i>opt</i>	References the IgnitionSystemType table. IgnitionSystemTypeID should be used for valid values of the id attribute.	<IgnitionSystemType id="78"/>  <!-- Distributor Breakerless -->
Make	id	VehAttr <i>opt</i>	References the Make table. MakeID should be used for valid values of the id attribute. One of the following must be sent in the Vehicle Ident Group: a Base Vehicle ID; a Make / Year or Make / Year-Range combination must be included with each application; a Base Equipment ID; or a Mfr / Equipment Model / Vehicle Type	<Make id="26"/>  <!-- Ford -->
MapperCompany	<i>None</i>	Header <i>opt</i>	Company responsible for mapping the data to ACES.	<MapperCompany>ACME Mapping, Inc.</MapperCompany>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
MapperContact	None	Header <i>opt</i>	Person to contact at the mapping company	<MapperContact>Joe Mapper</MapperContact>
MapperEmail	None	Header <i>opt</i>	Mapping contact email address	<MapperEmail>joe@mapper.c om</MapperEmail>
MapperPhone	None	Header <i>opt</i>	Phone number of mapping contact	<MapperPhone>111-111-1111 </MapperPhone>
MapperPhoneExt	None	Header <i>opt</i>	Phone number extension of mapping contact	<MapperPhoneExt>111 </MapperPhoneExt>
Mfr	id	VehAttr <i>opt</i>	The manufacturer that actually built the equipment. References the Mfr table. MfrID should be used for valid values of the id attribute. This tag is required if Equipment tags are being supplied.	<Mfr id="544"/>  <!-- Ford -->
MfrBodyCode	id	VehAttr <i>opt</i>	References the MfrBodyCode table. MfrBodyCodeID should be used for valid values of the id attribute.	<MfrBodyCode id="18"/>  <!-- K -->
MfrLabel	None	App <i>opt</i>	Allows a manufacturer to specify the part label used to describe their specific parts.	<MfrLabel>ACME Superduper Strut</MfrLabel>
Model	id	VehAttr <i>opt</i>	If a BaseVehicle tag is not used in an application, a Model tag must be supplied.	<Model id="697"/>  <!-- Taurus -->

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
Note	id, lang	App <i>opt</i> ACES 1.0	The Note tag is used to deliver comment information not covered by the reference tags. An optional “id” may be supplied to reference an external coded note file. An optional “lang” attribute may be used for multi-language notes (ISO table 639 values). If multiple Note tags are used in an application, they are logically joined by “and.” Only use this tag in special cases for ACES 2.0 files. <b>The Note tag is deprecated in favor of the Qual tag and may be removed in future releases.</b>	<Note>w/AC</Note>  <Note id="258"/>  <Note lang="fr">je ne parle pas français</Note>  <Note>Exc. A.C. or H.D. tow package</Note>
OrientationView	None	DigitalAsset <i>opt</i>	Code identifying the Orientation View of application-specific Digital Asset.	<OrientationView>ANG</OrientationView>
Part	BrandAAIAID	App <i>req</i>	The part number as it should be displayed to the user. The optional BrandAAIAID attribute is used to override the default brand defined in the Header. It is a 4 character reference to the “Brand Code” level of the AAIA Parent/Brand table.	<Part BrandAAIAID="BBBB">P192-12</Part>
PartType	id	App <i>req</i>	A reference to the PCdb Part Terminology ID.	<PartType id="15"/>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
PcdbVersionDate	None	Header <i>req</i>	Contains value of PcdbVersionDate from the PCdb Version table used to create the Transfer File. Format is YYYY-MM- DD where "YYYY" is the year, "MM" the month and "DD" the day.	<PcdbVersionDate>2001-02-15 </PcdbVersionDate>
Position	id	App <i>opt</i>	References the Position table. (Part of the relational PCdb). PositionID should be used for valid values of the id attribute.	<Position id="23"/>  <!-- Upper -->
PowerOutput	id  ACES 3.0	VehAttr <i>opt</i>	References the PowerOutput table. PowerOutputID should be used for valid values of the id attribute.	<PowerOutput id="5"/>
QdbVersionDate	None	Header <i>req</i>	Contains value of QdbVersionDate from the Qdb Version table used to create the Transfer File. Format is YYYY-MM- DD where "YYYY" is the year, "MM" the month and "DD" the day.	<QdbVersionDate>2001-02-15 </QdbVersionDate>
Qual	id	App <i>opt</i> ACES 2.0	Coded replacement for the Note tag. The required "id" attribute references the Qdb table. Sub-elements include one or more optional "param" tag and a required "text" tag.	<Qual id="123">  <param value="14" uom="in"/>  <text>With 14" Wheels</text>  </Qual>
Qty	None	App <i>req</i>	A required value to denote the number of parts required at that position for the application.	<Qty>1</Qty>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
RearBrakeType	id	VehAttr <i>opt</i>	The brake type used on the rear wheels. References the BrakeType table.  BrakeTypeID should be used for valid values of the id attribute.	<RearBrakeType id="1"/>  <!-- Disc -->
RearSpringType	id	VehAttr <i>opt</i>	The basic suspension type used in the rear of the vehicle. References the SpringType table. SpringTypeID should be used for valid values of the id attribute.	<RearSpringType id="45"/>  <!-- Coil -->
RecordCount	None	Footer <i>req</i>	The actual number of application tags contained in the file.	<RecordCount>5</RecordCount>
Region	id	VehAttr <i>opt</i>	Region where sold. References the Region table. RegionID should be used for valid values of the id attribute.	<Region id="2"/>  <!-- USA -->
Representation	None	DigitalAsset <i>opt</i>	Code identifying the Representation of the application-specific Digital Asset.	<Representation>A</Representation>
Resolution	None	DigitalAsset <i>opt</i>	Code identifying the Resolution of the application-specific Digital Asset.	<Resolution>72</Resolution>
SenderName	None	Header <i>opt</i>	Contact name at the Company	<SenderName>Joe Sender</SenderName>
SenderPhone	None	Header <i>opt</i>	Contact Phone number	<SenderPhone>111-111-1111</SenderPhone>
SenderPhoneExt	None	Header <i>opt</i>	Contact Phone Extension	<SenderPhoneExt>111</SenderPhoneExt>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
SteeringType	id	VehAttr <i>opt</i>	References the SteeringType table. SteeringTypeID should be used for valid values of the id attribute.	<SteeringType id="1"/>  <!-- Rack -->
SteeringSystem	id	VehAttr <i>opt</i>	References the SteeringSystem table. SteeringSystemID should be used for valid values of the id attribute.	<SteeringSystem id="5"/>  <!-- Power -->
SubmissionType	<i>None</i>	Header <i>req</i>	FULL, UPDATE, or TEST. See the discussion in section <a href="#">5.4</a> .	<SubmissionType>FULL  </SubmissionType>
SubModel	id	VehAttr <i>opt</i>	References the SubModel table. SubModelID should be used for valid values of the id attribute. If a <SubModel> tag is used, you must also include a <Model> tag.	<SubModel id="176"/>  <!-- Deluxe -->
TransElecControlled	id	VehAttr <i>opt</i>	References the ElecControlled table. ElecControlledID should be used for valid values of the id attribute.	<TransElecControlled id="2"/>  <!-- Yes -->
TransferDate	<i>None</i>	Header <i>req</i>	When the transfer file was created. Format is YYYY-MM-DD where "YYYY" the year, "MM" the month and "DD" the day. Zero fill if necessary.	<TransferDate>2001-02-15  </TransferDate>
TransmissionMfr	id	VehAttr <i>opt</i>	The manufacturer that actually built the transmission. References the Mfr table. MfrID should be used for valid values of the id attribute.	<TransmissionMfr id="544"/>  <!-- Ford -->

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
TransmissionMfrCode	id	VehAttr <i>opt</i>	References the TransmissionMfrCode table. TransmissionMfrCodeID should be used for valid values of the id attribute.	<TransmissionMfrCode id="37"/>  <!--AW70 -->
TransmissionBase	id	VehAttr <i>opt</i>	References the TransmissionBase table. TransmissionBaseID should be used for valid values of the id attribute.	<TransmissionBase id="13"/>  <!-- 4 Speed, AT-->
TransmissionControlType	id	VehAttr <i>opt</i>	References the TransmissionControlType table. TransmissionControlTypeID should be used for valid values of the id attribute.	<TransmissionControlType id="107"/>  <!-- AT-->
TransmissionNumSpeeds	id	VehAttr <i>opt</i>	References the TransmissionNum-Speeds table. TransmissionNum-SpeedsID should be used for valid values of the id attribute.	<TransmissionNumSpeeds id="5"/>  <!-- 4 Speed-->
TransmissionType	id	VehAttr <i>opt</i>	References the TransmissionType table. TransmissionTypeID should be used for valid values of the id attribute.	<TransmissionType id="6"/>  <!-- Transaxle -->
URI	<i>none</i>	DigitalAsset <i>opt</i>	URI (Uniform Resource Indicator) or URL location of the application-specific digital asset. This can refer to a specific Digital asset item, or a page of content.	<URI>http://www.mfg.com/images/xyz.jpg</URI>

XML Element	XML Attribute	Segment Requirement	Description	Example XML Code
ValvesPerEngine	id	VehAttr <i>opt</i>	References the Valves table. ValvesID should be used for valid values of the id attribute.	<ValvesPerEngine id="5"/>  <!-- 24 Valves -->
VcdbVersionDate	None	Header <i>req</i>	Contains value of VcdbVersionDate from the VCDB Version table used to create the Transfer File. Format is YYYY-MM-DD where "YYYY" is the year, "MM" the month and "DD" the day.	<VcdbVersionDate>2001-02-15 </VcdbVersionDate>
VehicleType	id	VehAttr <i>opt</i>	References the VehicleType table. VehicleTypeID should be used for valid values of the id attribute. This tag is required if Equipment tags are being supplied.	<VehicleType id="2"/>  <!-- Truck-->
WheelBase	id	VehAttr <i>opt</i>	References the WheelBase table. WheelBaseID should be used for valid values of the id attribute.	<WheelBase id="21"/>  <!-- 101 Inches-->
Years	from, to	VehAttr <i>opt</i>	Reference the Year table. Both attributes are required, even if they are the same. One of the following must be sent in the Vehicle Ident Group: a Base Vehicle ID; a Make / Year or Make / Year-Range combination must be included with each application; a Base Equipment ID; or a Mfr / Equipment Model / Vehicle Type	<Years from="1982" to="1990"/>

## 6 XML Specifics

This section defines some of the special requirements of the XML file.

### 6.1 Declaration Statement

The following line should be placed at the beginning of the Transfer File. It tells XML parsers that they are reading a valid file, what version of XML it is coded with, and the character set used.

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

### 6.2 Character Encoding

The currently supported character encoding for ACES XML files is “UTF-8”. You should not, however, assume all 191 characters of this encoding are accepted by all trading partners. For example, you should ask before sending these common symbols: (a) degree sign (°); (b) plus or minus (±); or (c) registered trademark (®).

### 6.3 Reserved Symbols

The following symbols may not be used in the **element text** and should be coded with “entity tags” as shown.

Symbol	Entity Tag
&	&amp;
<	&lt;
>	&gt;

The following symbols may not be used in **XML attribute values** and should be coded with “entity tags” as shown. (There should be no reason for these codes when the attribute represents an “id” value, however).

Symbol	Entity Tag
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&apos;

### 6.4 Comments

Standard XML comments may be used for documentation or other purposes. Receivers of the data will not process these comments and so should be used for internal use only. The syntax is:

<!-- comment goes here -->

## 6.5 Empty Element Tags

Empty element tags are supported in two ways: <tag/> and <tag></tag>. The first style is a simplified form often used with tags that contain only an attribute value (such as an id).

## 6.6 Element Tag Order

A specific order of the sub-elements within the Header and App elements must be followed to create a “valid” delivery file. For example, the Company tag must be the first tag in the Header section followed by the SenderName and SenderPhone. See the *XML schema* (xsd) files for details. (**Note:** some receivers of the data may relax this restriction, but it is always a good practice to create an XML file that will validate against the xsd.)

# 7 Legacy Make Model Table

As of December 31, 2012 the Legacy Make Model Table is no longer supported.

## Appendix A. ACES Element Data Tags

### Main Structure Tags

ACES  
App  
Asset [ACES 3.0]  
DigitalAsset  
Footer Header

FrontBrakeType  
FrontSpringType  
FuelDeliverySubType  
FuelDeliveryType  
FuelSystemControlType  
FuelSystemDesign  
FuelType

### Vehicle Identification Tags

BaseVehicle Make  
Years  
EquipmentBase  
Mfr  
EquipmentModel  
VehicleType

IgnitionSystemType  
MfrBodyCode  
Model  
PowerOutput [ACES 3.0]  
RearBrakeType  
RearSpringType  
Region

### Vehicle Attribute Tags\*

Aspiration  
AssetName [ACES 3.0]  
BedLength  
BedType  
BodyNumDoors  
BodyType  
BrakeABS  
BrakeSystem  
CylinderHeadType  
DriveType  
EngineBase  
EngineBlock  
EngineBoreStroke  
EngineDesignation  
EngineMfr  
EngineVersion  
EngineVIN

SteeringType  
SteeringSystem  
SubModel  
TransElecControlled  
TransmissionBase  
TransmissionControlType  
TransmissionMfr  
TransmissionMfrCode  
TransmissionNumSpeeds  
TransmissionType  
ValvesPerEngine  
VehicleType  
WheelBase

### App-Specific Tags

AssetItemOrder [ACES 3.0]  
AssetItemRef [ACES 3.0]  
DisplayOrder  
MfrLabel

Note\* [deprecated]

Part PartType Position Qty

Qual\* [ACES 2.0]

**Header/Footer Tags**

ApprovedFor

BrandAAIAID [ACES 3.0]

Company

Country

DocFormNumber

DocumentTitle

EffectiveDate

MapperCompany

MapperContact

MapperEmail

MapperPhone

MapperPhoneExt

MfrCode

PcdbVersionDate

QdbVersionDate

RecordCount

SenderName

SenderPhone

SenderPhoneExt

SubmissionType

TransferDate

VcdbVersionDate

**Digital Asset Tags**

AssetDescription

AssetDimensions

AssetHeight

AssetType

AssetWidth

Background

ColorMode

Country

DigitalAsset

DigitalFileInformation

EffectiveDate

ExpirationDate

FileDateModified

FileName

FilePath

FileSize

FileType

OrientationView

Representation

Resolution

URI

\* Also valid in Asset element groups.

## Appendix B. Example Transfer File

```
<?xml version="1.0" encoding="UTF-8"?>
<ACES version="4.0">
  <Header>
    <Company>AcmeSupply</Company>
    <SenderName>Joe Smith</SenderName>
    <SenderPhone>999-999-9999</SenderPhone>
    <TransferDate>2018-06-04</TransferDate>
    <BrandAAIAID>ZZZZ</BrandAAIAID>
    <DocumentTitle>Electrical Switches</DocumentTitle>
    <EffectiveDate>2018-06-04</EffectiveDate>
    <SubmissionType>FULL</SubmissionType>
    <VcdbVersionDate>2018-05-25</VcdbVersionDate>
    <QdbVersionDate>2018-05-25</QdbVersionDate>
    <PcdbVersionDate>2018-05-25</PcdbVersionDate>
  </Header>

  <App action="A" id="1">
    <BaseVehicle id="2771"/>      <!-- 1997 Cadillac Catera -->
    <Qty>1</Qty>
    <PartType id="4472"/> <!-- Electrical/Headlight/Switch -->
    <Part>SW1406</Part>
  </App>

  <App action="A" id="2">
    <BaseVehicle id="2772"/>      <!-- 1998 Cadillac Catera -->
    <Qty>1</Qty>
    <PartType id="4472"/> <!-- Electrical/Headlight/Switch -->
    <Part>SW1406</Part>
  </App>

  <App action="A" id="3">
    <Years from="1997" to="1998"/>
    <Make id="46"/>      <!-- 1997-1998 Cadillac Catera -->
    <Model id="404"/>
    <Qty>1</Qty>
  </App>
</ACES>
```

```

        <PartType id="4472"/> <!-- Electrical/Headlight/Switch -->
        <Part>SW1406</Part>
</App>
<App action="A" id="4">
    <BaseVehicle id="4472"/>      <!-- 1997 Cadillac Catera -->
    <EngineBase id="389"/>      <!-- V6 181ci 3.0L -->
    <EngineVIN id="18"/>  <!-- [R] -->
    <Qual id="23">
    <text>With Air Conditioning</text>
    </Qual>
    <Qual id="929">
    <param value="90487546"/>
    <text>w/Starter 90487546 (1st Design)</text>
    </Qual>
    <Qty>1</Qty>
    <PartType id="4188"/> <!-- Electrical/Starter/Solenoid Switch -->
    <Part>SS769</Part>
</App>
<App action="A" id="5">
    <BaseVehicle id="9281"/>      <!-- 1972 Ford LTD -->
    <Note>Exc. Wagon</Note>      <!-- Allowed because BodyType is U/K -->
    <Qty>1</Qty>
    <PartType id="4472"/> <!-- Electrical/Headlight/Switch -->
    <Part>SW1406</Part>
</App>
<App action="A" id="6" validate="no">
    <BaseVehicle id="5404"/>      <!--1980 Ford Fairmont -->
    <Aspiration id="6"/>      <!-- Turbo (not valid in VehicleTo table) -->
    <Qty>1</Qty>
    <PartType id="10068"/>      <!-- Radiator Coolant Hose -->
    <Position id="46"/>      <!-- Upper -->
    <Part>H9281</Part>
</App>
<Footer>
    <RecordCount>6</RecordCount>

```

</Footer>

</ACES>

## Appendix C. Delivering “Invalid” Applications

It is understood that the VCdb cannot be “perfect.” For example, there will always be un-researched information on older vehicles. The following are common conditions you may encounter and how they should be handled.

1. Your application has an attribute not found in the VehicleTo table on a “completely researched” vehicle (i.e. there are no N/A or U/K values for that attribute).

*You should petition for it to be added. (If you can't wait, include it with an existing attribute id and flag the application with **validate="no"** in the App or Asset tag so receivers can easily identify them.)*

2. Your application has an attribute on an “incompletely researched” vehicle that is not in the VehicleTo table.

*You are allowed to code it as if it was valid. It should not be rejected by the receiver.*

3. Your application has an “excepted” attribute on an “incompletely researched” vehicle.

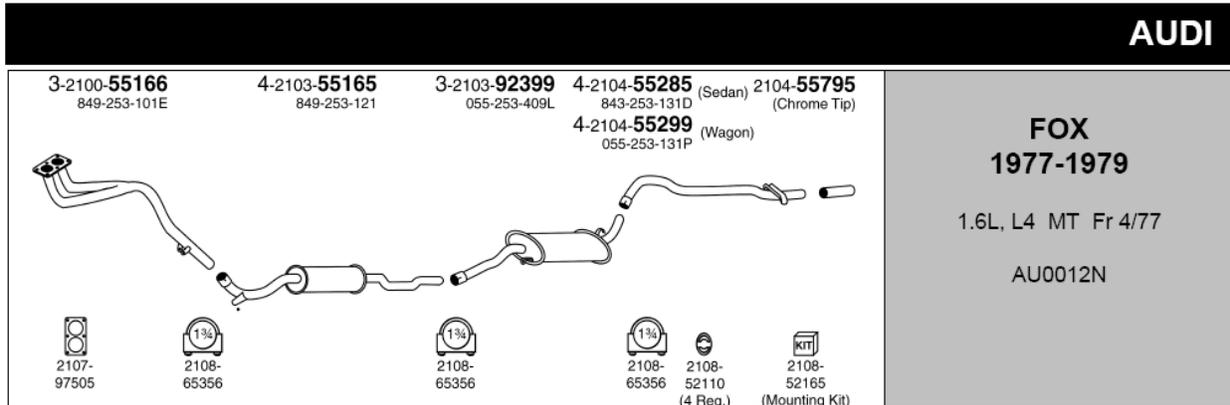
*This has always been handled as an “excepted attribute” in the Notes (e.g. “Exc. Sedan”). A future release may assign a special tag for this condition instead of using Notes.*

4. Your application has a new attribute that doesn't exist in the vehicle table at all.

*This should be an infrequent occurrence, but if it happens, you should petition for the new attribute value to be added to the appropriate table. (Exceptions are transmission codes or engine designations which tend to change more often).*

## Appendix D. Delivering Application Digital Assets

### Catalog Diagram 1



#### Key Points:

1. The manufacturer may have several “physical” files (format, resolutions, URL’s, etc.) to represent this “logical” image (diagram AU0012N).
2. This image includes fitment detail in the heading (Make, Model, Years, Engine, TransControlType) that must be propagated to App records
3. Unvalidated, this image represents approx. 12 App records using the YearRange-Make method, and approx. 36 App records using the BaseVehicle method

```
<?xml version="1.0" encoding="UTF-8"?>
<ACES version="4.0">
<Header>
...
</Header>
<App action="A" id="1">
    <Years from='1977' to='1979'/>
    <Make id="73"/>        <!-- Audi -->
    <Model id="954"/>     <!-- Fox -->
    <EngineBase id="953"/>    <!-- L4-1588cc/97cid 1.6L -->
    <TransmissionControlType id="6"/>    <!-- Standard -->
    <Note>Fr 4/77</Note>
```

```

        <Qty>1</Qty>
        <PartType id="5836"/> <!-- Exhaust Pipe Flange Gasket -->
        <Part>2107-97505</Part>
        <AssetName>AU0012N</AssetName>
        <AssetItemOrder>1</AssetItemOrder>
    </App>
    <App action="A" id="9">
        <Years from='1977' to='1979'/>
        <Make id="73"/> <!-- Audi -->
        <Model id="954"/> <!-- Fox -->
        <BodyType id="5"/> <!-- Sedan -->
        <EngineBase id="953"/> <!-- L4-1588cc/97cid 1.6L -->
        <TransmissionControlType id="6"/> <!-- Standard -->
        <Note>Fr 4/77</Note>
        <Qty>1</Qty>
        <PartType id="10023"/> <!-- Exhaust Tail Pipe -->
        <Part>4-2104-55285</Part>
        <DisplayOrder>1</DisplayOrder>
        <AssetName>AU0012N</AssetName>
        <AssetItemOrder>9</AssetItemOrder>
    </App>
    <App action="A" id="10">
        <Years from='1977' to='1979'/>
        <Make id="73"/> <!-- Audi -->
        <Model id="954"/> <!-- Fox -->
        <BodyType id="6"/> <!-- Wagon -->
        <EngineBase id="953"/> <!-- L4-1588cc/97cid 1.6L -->
        <TransmissionControlType id="6"/> <!-- Standard -->
        <Note>Fr 4/77</Note>
        <Qty>1</Qty>
        <PartType id="10023"/> <!-- Exhaust Tail Pipe -->
        <Part>4-2104-55299</Part>
        <DisplayOrder>2</DisplayOrder>
        <AssetName>AU0012N</AssetName>
        <AssetItemOrder>9</AssetItemOrder>
    
```

```

</App>
<App action="A" id="12">
  <Years from='1977' to='1979'/>
  <Make id="73"/>      <!-- Audi -->
  <Model id="954"/>    <!-- Fox -->
  <EngineBase id="953"/>      <!-- L4-1588cc/97cid 1.6L -->
  <TransmissionControlType id="6"/>    <!-- Standard -->
  <Note>Fr 4/77</Note>
  <Qty>1</Qty>
  <PartType id="5812"/> <!-- Exhaust Tail Pipe Chrome Tip -->
  <Part>2104-55795</Part>
  <AssetName>AU0012N</AssetName>
  <AssetItemOrder>11</AssetItemOrder>
</App>
...
<DigitalAsset>
  <DigitalFileInformation AssetName="AU0012N" action="A" LanguageCode="EN">
    <FileName>abc.jpg</FileName>
    <AssetDetailType>LIN</AssetDetailType>
    <FileType>JPG</FileType>
    <Representation>A</Representation>
    <FileSize>123456</FileSize>
    <Resolution>72</Resolution>
    <ColorMode>RGB</ColorMode>
    <Background>WHI</Background>
    <OrientationView>NUL</OrientationView>
    <AssetDimensions UOM="PX">
      <AssetHeight>250</AssetHeight>
      <AssetWidth>250</AssetWidth>
    </AssetDimensions>
    <AssetDescription>Line Art Diagram</AssetDescription>
    <FilePath>\Mfg\abc.jpg</FilePath>
    <URI>http://www.mfg.com/Images/abc.jpg</URI>
    <FileDateModified>2013-01-31</FileDateModified>
    <EffectiveDate>2013-01-31</EffectiveDate>
  </DigitalFileInformation>
</DigitalAsset>

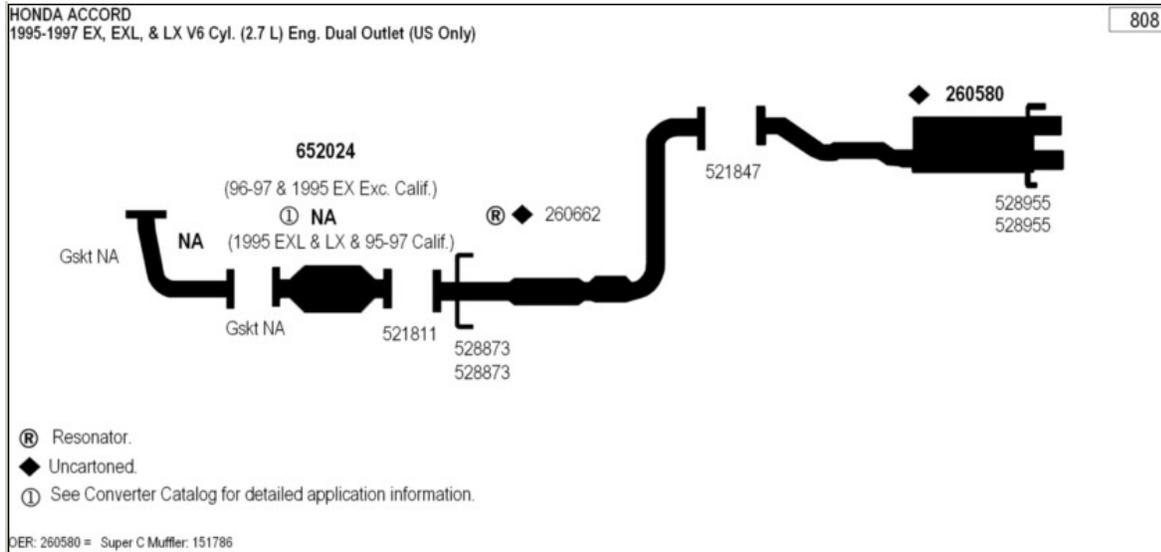
```

```
<ExpirationDate>2013-12-31</ExpirationDate>
<Country>US</Country>
</DigitalFileInformation>
</DigitalAsset>
<Footer>
<RecordCount>12</RecordCount>
</Footer>
</ACES>
```

### Key Points:

1. AssetName is the “logical” name of the image and is a key reference to the DigitalFileInformation with the same value for the AssetName attribute.
2. AssetItemOrder is the logical order of the items within the Asset, specified by the data provider.
3. Apps and Assets are not directly related. The receiver can choose how/whether to integrate them, using the AssetName.
4. Asset heading fitment details have been propagated to the Apps (Years, Make, Model, Engine, TransControlType, Note)
5. Certain Apps have fitment details (i.e. BodyType) that are in addition to those propagated from the Asset heading.

**Catalog Diagram 2**



**Key Points:**

1. The manufacturer has several “physical” files in different formats to represent this “logical” image (diagram 808).
2. This image includes parts that are not available (part number N/A)
3. This image includes graphical fitment detail (bullets, footnotes)
4. This image includes fitment detail in the heading (Make, Model, Years, Submodel, Engine, Note, Region) that must be propagated to App records
5. This image includes fitment detail in the heading that requires multiple values for an Aces tag (Submodel), which requires separate App records for each value, and that must be resolved with values already applied to the App records
6. Unvalidated, this image represents approx. 48 App records using the YearRange-Make method, and approx. 126 App records using the BaseVehicle method

```
<?xml version="1.0" encoding="UTF-8"?>
<ACES version="4.0">
<Header>...</Header>
<App action="A" id="1">
    <Years from='1995' to='1997'/>
    <Make id="59"/>    <!-- Honda -->
```

```

    <Model id="751"/>      <!-- Accord -->
    <SubModel id="191"/> <!-- EX-->
    <EngineBase id=" 1738"/>      <!-- V6-2675cc 2.7L -->
    <Region id="1"/>      <!-- USA -->
    <Note>Dual Outlet</Note>
    <Qty>1</Qty>
    <PartType id="5836"/> <!-- Exhaust Pipe Flange Gasket -->
    <Part>NA</Part>
    <AssetName>808</AssetName>
    <AssetItemOrder>1</AssetItemOrder>
</App>
<App action="A" id="2">
    <Years from='1995' to='1997'/>
    <Make id="59"/>      <!-- Honda -->
    <Model id="751"/>      <!-- Accord -->
    <SubModel id="191"/> <!-- EX-->
    <EngineBase id=" 1738"/>      <!-- V6-2675cc 2.7L -->
    <Region id="1"/>      <!-- USA -->
    <Note>Dual Outlet</Note>
    <Qty>1</Qty>
    <PartType id="10038"/> <!-- Exhaust Pipe -->
    <Part>NA</Part>
    <AssetName>808</AssetName>
    <AssetItemOrder>2</AssetItemOrder>
</App>
<App action="A" id="3">
    <Years from='1995' to='1997'/>
    <Make id="59"/>      <!-- Honda -->
    <Model id="751"/>      <!-- Accord -->
    <SubModel id="191"/> <!-- EX-->
    <EngineBase id=" 1738"/>      <!-- V6-2675cc 2.7L -->
    <Region id="1"/>      <!-- USA -->
    <Note>Dual Outlet</Note>
    <Qty>1</Qty>
    <PartType id="5836"/> <!-- Exhaust Pipe Flange Gasket -->

```

```

    <Part>NA</Part>
    <AssetName>808</AssetName>
    <AssetItemOrder>3</AssetItemOrder>
</App>
<App action="A" id="4">
    <Years from='1995' to='1997'/>
    <Make id="59"/>      <!-- Honda -->
    <Model id="751"/>    <!-- Accord -->
    <SubModel id="191"/> <!-- EX-->
    <EngineBase id=" 1738"/> <!-- V6-2675cc 2.7L -->
    <Region id="1"/>    <!-- USA -->
    <Note>Dual Outlet</Note>
    <Note>Exc. Calif.</Note>
    <Qty>1</Qty>
    <PartType id="5808"/> <!-- Catalytic Converter -->
    <Part>652024</Part>
    <AssetName>808</AssetName>
    <AssetItemOrder>4</AssetItemOrder>
</App>
...
<DigitalAsset>
    <DigitalFileInformation AssetName="808" action="A" LanguageCode="EN">
        <FileName>808_LIN_NUL.jpg</FileName>
        <AssetDetailType>LIN</AssetDetailType>
        <FileType>JPG</FileType>
        <Representation>A</Representation>
        <FileSize>123456</FileSize>
        <Resolution>72</Resolution>
        <ColorMode>RGB</ColorMode>
        <Background>WHI</Background>
        <OrientationView>NUL</OrientationView>
        <AssetDimensions UOM="PX">
            <AssetHeight>700</AssetHeight>
            <AssetWidth>1500</AssetWidth>
        </AssetDimensions>
    </DigitalFileInformation>
</DigitalAsset>

```

```

        <AssetDescription>Line Art Diagram</AssetDescription>
        <FilePath>\Mfg\808_LIN_NUL.jpg</FilePath>
        <URI>http://www.mfg.com/Images/808\_LIN\_NUL.jpg</URI>
        <FileDateModified>2013-01-31</FileDateModified>
        <EffectiveDate>2013-01-31</EffectiveDate>
        <ExpirationDate>2013-12-31</ExpirationDate>
        <Country>US</Country>
    </DigitalFileInformation>
<DigitalFileInformation AssetName="808" action="A" LanguageCode="EN">
    <FileName>808_LIN_NUL.png</FileName>
    <AssetDetailType>LIN</AssetDetailType>
    <FileType>PNG</FileType>
    <Representation>A</Representation>
    <FileSize>123456</FileSize>
    <Resolution>72</Resolution>
    <ColorMode>RGB</ColorMode>
    <Background>WHI</Background>
    <OrientationView>NUL</OrientationView>
    <AssetDimensions UOM="PX">
        <AssetHeight>700</AssetHeight>
        <AssetWidth>1500</AssetWidth>
    </AssetDimensions>
    <AssetDescription>Line Art Diagram</AssetDescription>
    <FilePath>\Mfg\808_LIN_NUL.png</FilePath>
    <URI>http://www.mfg.com/Images/808\_LIN\_NUL.png</URI>
    <FileDateModified>2013-01-31</FileDateModified>
    <EffectiveDate>2013-01-31</EffectiveDate>
    <ExpirationDate>2013-12-31</ExpirationDate>
    <Country>US</Country>
</DigitalFileInformation>
</DigitalAsset>
<Footer>
<RecordCount>41</RecordCount>
</Footer>
</ACES>
    
```

**Key Points:**

1. AssetName is the “logical” name of the image and is a key reference to all DigitalFileInformation elements with the same value for the AssetName attribute.
2. AssetItemOrder is the logical order of the items within the Asset, specified by the data provider.
3. Apps and Assets are not directly related. The receiver can choose how/whether to integrate them, using the AssetName.
4. Asset heading fitment details have been propagated to the Apps (Years, Make, Model, Submodels, Engine, Region, Note)
5. Unavailable parts are included, in order to provide context for available parts

**Catalog Diagram 3**

**Key Points:**

1. The manufacturer may have several “physical” files (format, resolutions, URL’s, etc.) to represent this “logical” image (Figure 1451190).
2. There are no part numbers.
3. There is no vehicle or other description image information – unlike exhaust, the image is the target of a lookup and doesn’t define the lookup.
4. This image illustrates some visual information (numbered locations).

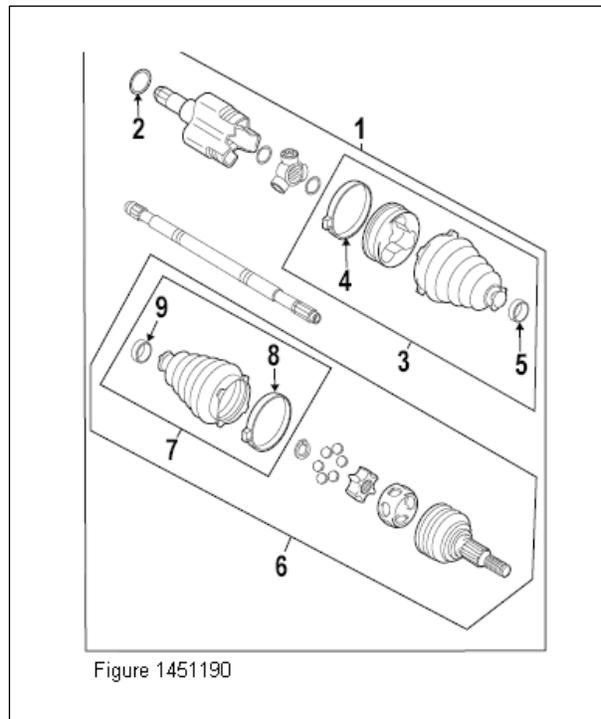


Figure 1451190

```
<?xml version="1.0" encoding="UTF-8"?>
<ACES version="4.0">
<Header>
...
</Header>
<Asset action="A" id="1">
```

```

        <BaseVehicle id="23938"/><!-- 2007 Suzuki XL-7 -->
        <AssetName>1451190</AssetName>
    </Asset >
    ...
    <DigitalAsset>
        <DigitalFileInformation AssetName="1451190" action="A" LanguageCode="EN">
            <FileName>1451190_LIN_NUL.jpg</FileName>
            <AssetDetailType>LIN</AssetDetailType>
            <FileType>JPG</FileType>
            <Representation>A</Representation>
            <FileSize>123456</FileSize>
            <Resolution>72</Resolution>
            <ColorMode>RGB</ColorMode>
            <Background>WHI</Background>
            <OrientationView>NUL</OrientationView>
            <AssetDimensions UOM="PX">
                <AssetHeight>1600</AssetHeight>
                <AssetWidth>1200</AssetWidth>
            </AssetDimensions>
            <AssetDescription>Line Art Diagram</AssetDescription>
            <FilePath>Mfg\1451190_LIN_NUL.jpg</FilePath>
            <URI>http://www.mfg.com/Images/1451190\_LIN\_NUL.jpg</URI>
            <FileDateModified>2013-01-31</FileDateModified>
            <EffectiveDate>2013-01-31</EffectiveDate>
            <ExpirationDate>2013-12-31</ExpirationDate>
            <Country>US</Country>
        </DigitalFileInformation>
    </DigitalAsset>
    <Footer>
    <RecordCount>24</RecordCount>
    </Footer>
</ACES>

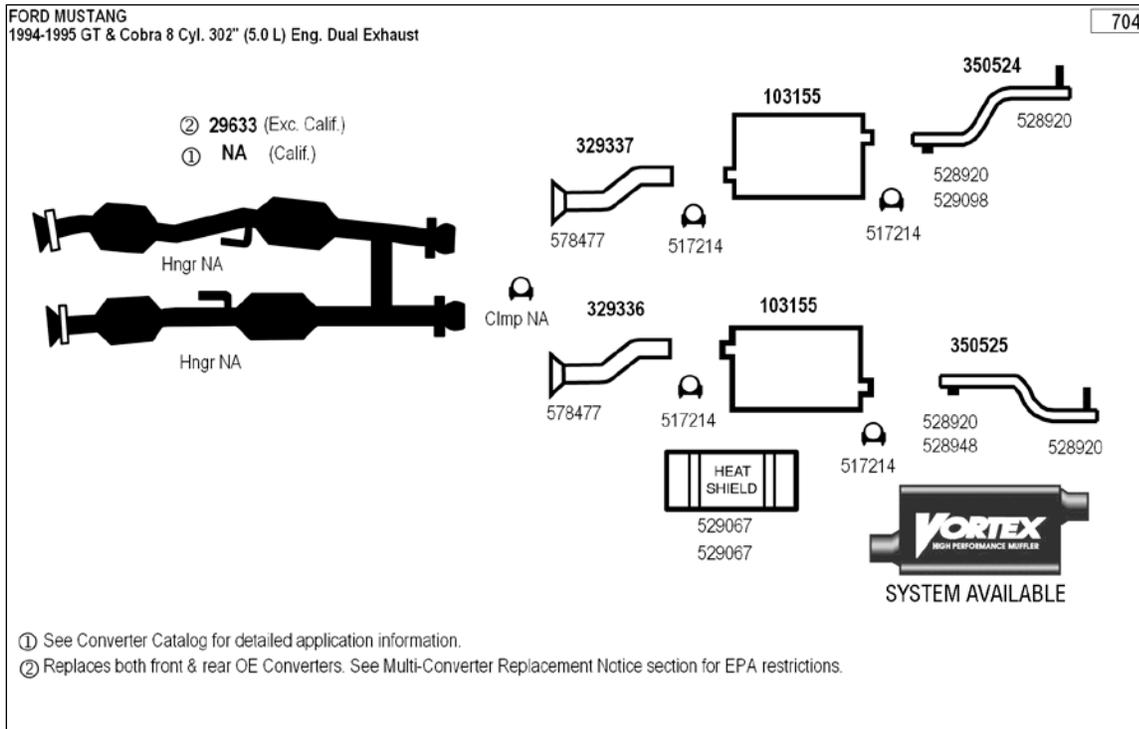
```

**Key Points:**

1. The Asset has no fitment detail and is related to the vehicles represented by the BaseVehicleID.

- AssetName is the “logical” name of the image and is a key reference to the DigitalFileInformation with the same value for the AssetName attribute.

**Catalog Diagram 4**



**Key Points:**

- The manufacturer may have several “physical” files (format, resolutions, URL’s, etc.) to represent this “logical” image (diagram 704).
- This image includes parts that are not available (part number NA)
- This image includes graphical fitment detail (P outlined components, footnotes)
- This image includes fitment detail in the heading (Make, Model, Years, Submodel, Engine, Note) that must be propagated to App records
- This image includes fitment detail in the heading that requires multiple values for an Aces tag (Submodel), which requires separate App records for each value
- Unvalidated, this image represents approx. 50 App records using the YearRange-Make method, and approx. 100 App records using the BaseVehicle method

ACES Document – Version 4.0

```
<?xml version="1.0" encoding="UTF-8"?>
<ACES version="4.0">
<Header>
...
</Header>
<App action="A" id="1">
  <Years from='1994' to='1995'/>
  <Make id="54"/>      <!-- Ford -->
  <Model id="688"/>   <!-- Mustang -->
  <SubModel id="39"/> <!-- GT -->
  <EngineBase id=" 143"/>      <!-- V8-302cid 5.0L -->
  <Note>Dual Exhaust</Note>
  <Note>Exc. Calif. </Note>
  <Note>Replaces both front And rear OE Converters</Note>
  <Qty>1</Qty>
  <PartType id="10038"/> <!-- Exhaust Pipe -->
  <Part>29633</Part>
  <AssetName>704</AssetName>
  <AssetItemOrder>1</AssetItemOrder>
</App>
...
<App action="A" id="43">
  <Years from='1994' to='1995'/>
  <Make id="54"/>      <!-- Ford -->
  <Model id="688"/>   <!-- Mustang -->
  <SubModel id="39"/> <!-- GT -->
  <EngineBase id=" 143"/>      <!-- V8-302cid 5.0L -->
  <Note>Dual Exhaust</Note>
  <Note>If Welded Replace All Required Parts</Note>
  <Qty>1</Qty>
  <PartType id="10023"/> <!-- Tail Pipe -->
  <Position id="2"/> <!-- Left -->
  <Part>350524</Part>
  <AssetName>704</AssetName>
  <AssetItemOrder>24</AssetItemOrder>
```

```

</App>
<App action="A" id="44">
  <Years from='1994' to='1995'/>
  <Make id="54"/>      <!-- Ford -->
  <Model id="688"/>   <!-- Mustang -->
  <SubModel id="39"/> <!-- GT -->
  <EngineBase id=" 143"/>      <!-- V8-302cid 5.0L -->
  <Note>Dual Exhaust</Note>
  <Note>If Welded Replace All Required Parts</Note>
  <Qty>1</Qty>
  <PartType id="10023"/> <!-- Tail Pipe -->
  <Position id="12"/> <!-- Right -->
  <Part>350524</Part>
  <AssetName>704</AssetName>
  <AssetItemOrder>24</AssetItemOrder>
</App>
...
<DigitalAsset>
  <DigitalFileInformation AssetName="704" action="A" LanguageCode="EN">
    <FileName>704_LIN_NUL.jpg</FileName>
    <AssetDetailType>LIN</AssetDetailType>
    <FileType>JPG</FileType>
    <Representation>A</Representation>
    <FileSize>123456</FileSize>
    <Resolution>72</Resolution>
    <ColorMode>RGB</ColorMode>
    <Background>WHI</Background>
    <OrientationView>NUL</OrientationView>
    <AssetDimensions UOM="PX">
      <AssetHeight>1500</AssetHeight>
      <AssetWidth>1500</AssetWidth>
    </AssetDimensions>
    <AssetDescription>Line Art Diagram</AssetDescription>
    <FilePath>\Mfg\704_LIN_NUL.jpg</FilePath>
    <URI>http://www.mfg.com/Images/704_LIN_NUL.jpg</URI>
  </DigitalFileInformation>
</DigitalAsset>

```

```
<FileDateModified>2013-01-31</FileDateModified>
<EffectiveDate>2013-01-31</EffectiveDate>
<ExpirationDate>2013-12-31</ExpirationDate>
<Country>US</Country>
</DigitalFileInformation>
</DigitalAsset>
<Footer>
<RecordCount>50</RecordCount>
</Footer>
</ACES>
```

### Key Points:

1. AssetName is the “logical” name of the image and is a key reference to the DigitalFileInformation with the same value for the AssetName attribute.
2. AssetItemOrder is the logical order of the items within the Asset, specified by the data provider.
3. Apps and Assets are not directly related. The receiver can choose how/whether to integrate them, using the AssetName.
4. Asset heading fitment details have been propagated to the Apps (Years, Make, Model, Submodels, Engine, Note)
5. Unavailable parts are included, in order to provide context for available parts
6. Asset graphical details have been applied to the Apps (Position, outlined components)