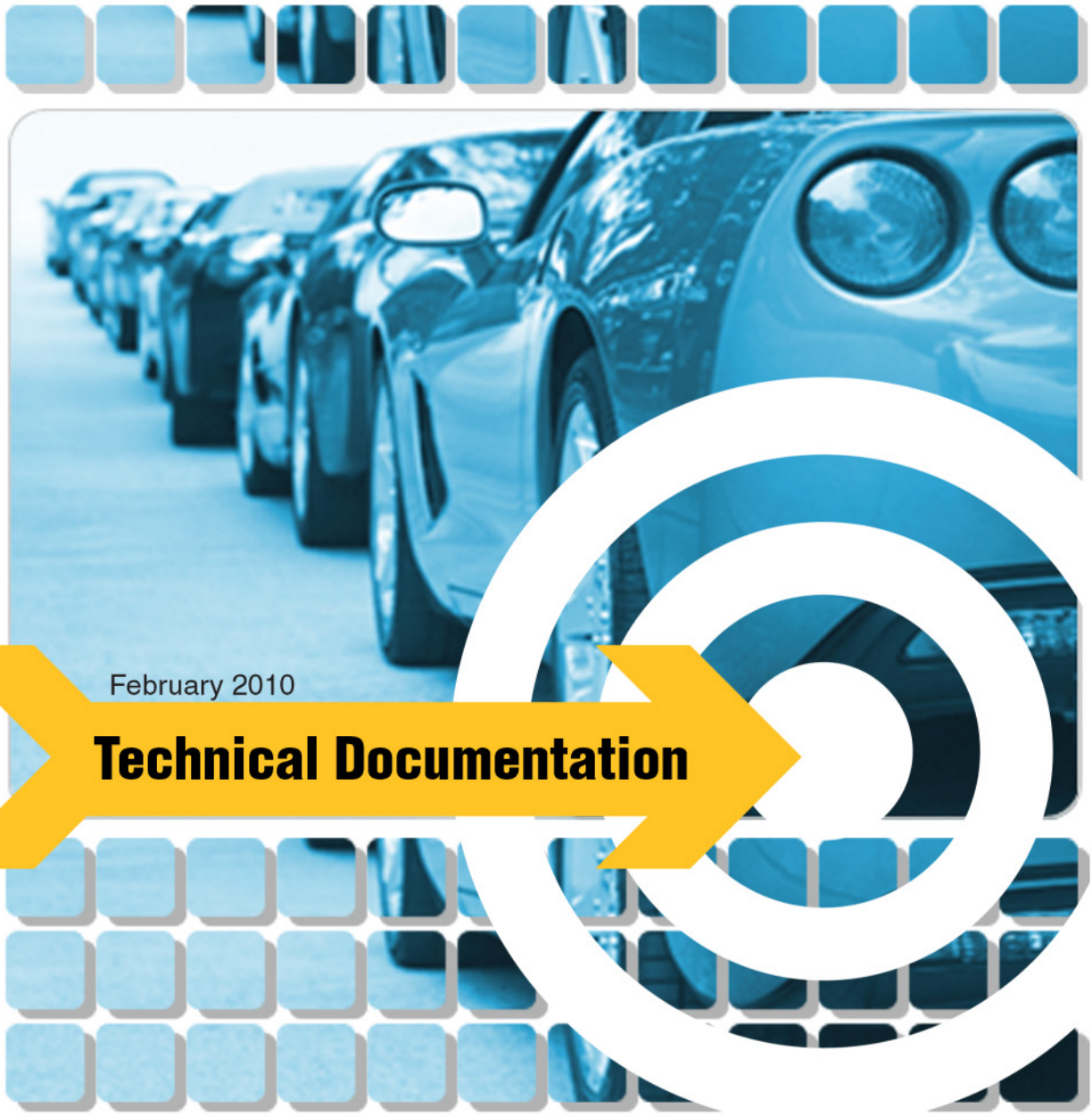
chrome®
The Automotive Information Professionals™

Chrome Construct Version 3



February 2010

Technical Documentation

TEL: 503 963 6300 | 800 936 8906
FAX: 503 963 6310

700 NE Multnomah St, 15th Floor
Portland, OR 97232

www.chrome.com

TABLE OF CONTENTS

Chrome Construct Developer's Guide

This document is organized into three main sections:

A. Technical Overview – This section provides information on migration considerations, a summary of operations, and instructions on how to connect to the Chrome Construct service.
B. Solutions Overview – This section describes how-to steps for implementing common components such as a vehicle configurator.
C. Interface Definitions – This section provides a detailed description of each operation and type supported by the web service including a description and example of each attribute.

The purpose of this document is to provide you with details of the Chrome Construct feature set and the application architecture.

Overview

Chrome Construct allows developers to build applications which allow users to:

- View information for base and configured vehicles including:
  - Vehicle name and additional basic information such as market class, body type, and drivetrain
  - Consumer friendly model and style names.
  - Vehicle pricing, including Base MSRP, Base Invoice (US only), and Destination Charge.
  - Option descriptions, codes, and pricing.
  - Standard Equipment.
  - Technical specifications which reflect the vehicle's current configuration.
  - Available Interior and Exterior Vehicle Colors including color swatches as images and as RGB Hex values.
  - Vehicle Images (stock photos).
  - Consumer Information, such as warranty, national rebates and incentives, recalls, and crash test ratings (when available) available both as editorial text and structured text to enable field level access to content such as incentive dollar amounts and percentages.
  - Categories (generic equipment descriptions).
  - Chrome vehicle reviews.
- Configure vehicles providing feedback on option availability and price changes.
- Compare the advantages and disadvantages of two or more vehicles, returned in natural-language terms.
- Compare equipment and technical specifications of two or more vehicles side-by-side.
- Automatically equip vehicles similarly to provide more accurate comparisons.
- Customize vehicle selection.
  - Request vehicles that are already configured with the base equipment selected.
  - Constrain vehicle selection by zip code
  - Select vehicles using Chrome standard names or consumer friendly names.
- Search Chrome's entire new vehicle data set using robust criteria based search criteria.
  - The ability to "find comparable" vehicles based on a reference vehicle.
  - Ability to constrain vehicle searches by ZIP Code.
  - Ability to search by categories, technical specifications, prices, vehicle type, etc.
  - Search by market class, drivetrain, number of passenger doors, or body style.

The service is backed by the same technology that enables Chrome's entire professional and consumer facing internet based products. With seamless automatic daily data updates, this product provides all of the flexibility of an API without the overhead of hosting one.

A. Technical Overview

The following section provides basic instructions regarding how to connect to the Chrome Construct service.

Using Web Services

Chrome Construct can be accessed by any application that supports web services. Because web services differ by implementation, this document is written to provide information related specifically to the functionality and data provided by this service.

In order to facilitate your use of the service, a web application sample has been provided in the following languages: Java, C# .NET, VB .NET, Coldfusion MX 7, and PHP.

The Chrome Construct web service implementation adheres to industry best practices for web service interoperability. In addition to complying with published standards listed below, we have independently verified our compatibility with the following development environments: Java, C# .NET, VB .NET, Coldfusion MX 7, and PHP.

Web Service Standards Conformance

Chrome Construct is implemented as a web service that conforms to WS-I Basic Profile 1.0, WSDL 1.1, and SOAP 1.1. These standards are further described below:

- **Web Services Interoperability Basic Profile 1.0** - A published description of what standards and technologies are required for interoperability between web service implementations on different software and operating system platforms (http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html).
- **WSDL 1.1 (Web Services Description Language)** - An XML document that describes a web service, including its name, operations, parameters for those operations, and the location of where to send requests (http://www.w3.org/TR/wsdl).
- **SOAP 1.1 (Simple Object Access Protocol)** - A standard for XML messaging and the mapping of data types so that applications adhering to these standards can communicate with each other (http://www.w3.org/TR/2000/NOTE-SOAP-20000508).

Operation Overview

Chrome Construct is a web service implemented in Java atop Chrome's core configuration and comparison libraries. The web service is designed to be scalable and to allow versatility in data presentation and workflow. Chrome Construct is scalable in that all transactions are stateless. This means that as users perform activities, such as selecting optional equipment, each response returns all of the information necessary to continue the transaction.

This service is versatile in that it allows developers to choose which elements are returned on a request to view a style or toggle any options. For example, if on a web page, a designer wishes to only show the vehicle's name and consumer information, the web service request can be filtered to show only these results, thereby limiting the amount of data being returned by each call. This filtering of results is easily enabled or disabled at runtime as part of the request parameter.

The service provides access to several operations related to vehicle descriptions. Below is the core list of services available for version 3. Each of these operations is elaborated upon in the interface definitions section of this document, including a detailed list of parameters and return types. The core operations are:

Operation Summary

Name	Returns
getModelYears	Provides a list of all available model years for use in building a vehicle selector.
getDivisions	Provides a list of all divisions for use in building a vehicle selector.
getSubdivisions	Returns all available subdivisions for use in building a vehicle selector.
getModelsByDivision	Returns all models by division, for use in building a vehicle selector.
getModelsBySubdivision	Returns all models by subdivision, for use in building a vehicle selector.
getConsumerFriendlyModelNames ByDivision	Returns the consumer friendly names of all models by division, for use in building a vehicle selector.
getConsumerFriendlyModelNames BySubdivision	Returns the consumer friendly names of all models by subdivision, for use in building a vehicle selector.
getStylesByConsumerFriendly ModelNameAndDivision	Returns all styles by consumer friendly model name and division, for use in building a vehicle selector.
getStylesByConsumerFriendly ModelNameAndSubdivision	Returns all styles by consumer friendly model name and subdivision, for use in building a vehicle selector.
getStyles	Provides a list of all styles for a given model for use in building a vehicle selector. The returned configuration state can be used for initiating a vehicle comparison transaction.
getConfiguration	Using a configuration state returned from the vehicle selector or from a previously saved configuration, provides access to the complete details of the style, including information regarding the vehicle, such as pricing, options, standards, and tech specs.
getConfigurationByStyleId	Using a Chrome style ID, provides access to the complete configuration of a style, including information regarding the vehicle, including pricing, options, standards, and tech specs.
getConfigurationByAutoBuilder StyleId	Using a Chrome AutoBuilderStyle ID, provides access to the complete configuration of a style, including information regarding the vehicle, including pricing, options, standards, and tech specs.
getStyleFullyConfigured	Using a configuration state returned from the vehicle selector, provides access to the complete details of the style with the base equipment pre-selected.
getStyleFullyConfiguredByStyleId	Using a Chrome Style ID, provides access to the complete details of the style with the base equipment pre-selected.
getStyleFullyConfiguredByAuto BuilderStyleId	Using a Chrome AutoBuilder Style ID provides access to the complete details of the style with the base equipment pre-selected.
materializeConfigurationState	Restores a previously saved configuration.
toggleOption	Toggles an option and returns the state of the vehicle, including refreshed vehicle information.
selectColor	Selects interior/exterior color information and adds this information to the state of the vehicle.
autoEquipStyles	Provides a means of equivalently equipping one or more styles based on a source style.
getMarketClasses	Returns all available market classes. Included as a reference for use with filtered vehicle selection and searching.
getDataVersions	Returns the data version information for each available locale.

Name	Returns
getCategoryDefinitions	Returns all categories for use in filtering side-by-side comparison results and searching.
getTechnicalSpecification Definitions	Returns all technical specifications for use in filtering side-by-side comparison results and searching.
getOptionKinds	Returns all available option kinds. Included as a reference for identifying types of options such as packages, engines, etc.
getAdvantageComparisonRuleSet Names	Returns all available advantage based comparison rule set names.
compareAdvantages	Returns natural language advantages and disadvantages.
compareSideBySide	Returns a side-by-side list of categories and technical specifications.
searchStyles	Provides a means of executing a search of new vehicle styles based on a robust set of search criteria.
validateSearch	Returns all search criterion errors. Included as a reference for developers to validate that the search criteria is well-formed before executing.
searchModels	Provides a means of executing a search of new vehicle models based on a robust set of search criteria.
getSearchCriterionDescriptors	Returns the search criterion descriptors. Included as a reference for developers when determining valid values for use in executing a search.

Programming Tips

The following tips are included as guidelines for programming tips related to your Chrome Construct implementation.

- Toolkits interpret the WSDL differently. The operation and type definitions in this document are based on the WSDL definition. Depending on your language and WDSL toolkit, the names of operations and even the data types may vary. For example, in PHP all values declared as Boolean in the WSDL are set as strings in the PHP implementation. As such it is suggested that you reference documentation specific to your web service implementation as well as the samples provided within this document.
- In PHP, where an array contains only a single item, the NuSOAP library converts these from an array to a single instance. Chrome's sample contains a helper function for handling this conversion.
- Also in PHP, complex types in an array cannot be initialized as null. All parameters need to be specified as true or false.

Migration Considerations

The new version of the Chrome Construct WSDL can be accessed from the following location:

https://platform.chrome.com/AutomotiveConfigCompareService/AutomotiveConfigCompareService3?WSDL

The migration from interface version 2 to interface version 3 is a major upgrade and will require code changes. These interface changes were required in order to introduce new functionality and improve the support of across multiple client platforms.

Major changes include:

- The configuration and comparison services are now combined in a single WSDL. As such it no longer requires that the configuration be serialized in order to pass configuration state between these services.

- In order to provide consistency across selection and search and to improve performance, Style is now a lightweight structure. Previously Style contained all content; this is now replaced with a Configuration structure which contains everything including the new Style. This change enables an intermediate list of styles to be displayed prior to the user selecting a single style. Previously one would be required to fully instantiate each style.
- Some structures have been flattened to make it easier to implement in languages such as PHP.
- The Chrome IDs are now Chrome's globally unique identifiers for subdivision, model and style. AutoBuilder Style IDs are still included for backward compatibility. Historical IDs have been removed.
- Color selections are now ID based. Previously colors were selected by code. In some circumstances codes were not yet available, which impacted accuracy of the selection. With the new ID structure, this problem has been corrected.
- Many operations which previously returned a StyleState now return a ConfigurationState with access to more content on the style include name and base pricing. This will remove the need to make a second call to the web service to access this level of content.
- The setup for include parameters when retrieving a style has changed to reflect changes in the overall structure.

New features include:

- All vehicle selection operations are now filterable by additional criteria such as zip code.
- There is a new search API that enables searching by criteria such as body type, equipment, technical specifications, and MSRP range to name a few.
- Vehicles can be pre-loaded with base equipment already selected.
- Option toggles can now be set to only return delta responses instead of the fully updated style. This still supports returning the full style for backward compatibility.
- There are new operations to support a vehicle selector based on Chrome's consumer friendly names.
- Consumer information can be accessed in a structured format. For example, direct access to the dollar amount on an incentive.
- There is now access to Chrome's automotive review content – AutoBriefs.
- Advantage based comparison can now support an unlimited number of comparison vehicles.
- There is a new operation for auto-equipping styles to match a source style. This enables more accurate comparisons.
- Color swatches are now available as RGB HEX values, in addition to color swatch images.
- Options now carry unique type filters which can be used to group mutually exclusive options in radio groups.

Please refer to the Migration Guide for additional information and details.

Contacting Client Support

Client Support is available by phone toll-free at (800) 937-3661, Monday through Friday, from 6:00 a.m. to 5:00 p.m. Pacific Time, or you can reach Client Support by email at support@chrome.com. This team can help you with product support, billing questions, and other inquiries.

Connecting to the Chrome Construct Service

The following section provides instructions for connecting to the service.

In order to use the Chrome Construct Service, you must supply a valid Chrome User ID and password, the URL of the WSDL defining the service, and the server namespace or package name. Chrome's order fulfillment department sent an email with a User ID and password to Chrome's contact person at your organization. Use the ID and password to initialize the `AccountInfo.accountNumber` and `AccountInfo.accountSecret` elements described in the AccountInfo section.

This document provides instructions for accessing the web services via the following methods:

- In Java, using JAX-RPC
- In Microsoft .NET, using a web reference
- In PHP, using NuSOAP
- In Coldfusion

Connecting to the Service from JAX-RPC

To call the Chrome Construct Service from JAX-RPC, you must create the stubs needed to communicate with the service and specify the service URL and namespace in the configuration file, config-wsdl.xml.

Stub Creation Command

```
%JWSDP_HOME%\jaxrpc\bin\wscompile -gen:client -f:unwrap -keep -d ./lib config-wsdl.xml
```

JAX-RPC Configuration File (config-wsdl.xml)

Note: Location must be on one line. Here the line is wrapped in order to present the information within this document:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration
  xmlns="http://java.sun.com/xml/ns/jax-rpc/ri/config">
    <wsdl

location="https://platform.chrome.com/AutomotiveConfigCompareService/AutomotiveConfigCompareService3?WSDL
      packageName="com.chrome.kp.configcompare3"/>

</configuration>
```

Note: The package name listed here matches the code samples. Package names are configurable.

Connecting to the Service from .NET for C#, VB, or ASP

To call the Chrome Construct Service from a Microsoft Visual Studio .NET environment, create a new project or open an existing project. Follow the steps below to specify the service URL and namespace.

1. From the Project menu, select Add Web Reference.
2. In the URL text box, type

   `https://platform.chrome.com/AutomotiveConfigCompareService/AutomotiveConfigCompareService3?WSDL`

   Visual Studio automatically discovers and displays the Automotive Description Service methods. The host name appears in the Web Reference Name text box.
3. To declare the host name that you would like to reference, append "." and AutomotiveConfigCompareService3 to the host name listed in the Web Reference Name text box, resulting in, for example:

   "`com.chrome.kp.configcompare3`"

Connecting With PHP

No resources specific to Chrome Construct are required to connect via PHP. However, a SOAP implementation for PHP must be used. The provided sample code uses the NuSOAP library (available at **http://sourceforge.net/projects/nusoap**). The sample code makes a connection by creating a new "SoapClient" object with the URL of the Chrome Construct Web Service. Note that NuSOAP has a WSDL cache feature that will speed performance by not requiring that the SoapClient be created for each request. This feature is illustrated in the sample provided.

Connecting With ColdFusion 7 and Greater

To connect to ColdFusion 7 and Greater:

1. Open the ColdFusion administrator website.
2. Go to the "Web Services" panel.
3. Add the URL to Chrome's web service and assign a name for reference within your implementation:
   `https://platform.chrome.com/AutomotiveConfigCompareService/AutomotiveConfigCompareService3?WSDL`
4. Reference the service by the name assigned in step 3. This will access the precompiled web service stubs.

B. Solutions Overview

Chrome Construct provides operations that enable you to search for, view, configure, and compare new vehicles. This document provides details on all of the interfaces of the service with instructions for implementing common solutions.

Using the Chrome Construct Service

Specifying a Locale

Locale is used to determine what vehicle data the Chrome Construct service will operate with. Currently Chrome Construct supports US vehicles in English and Canadian vehicles in both Canadian English and French. Your Chrome Construct account must be licensed for each country that you wish to have access to. Each call you make requires an AccountInfo object. To specify the locale, set the "locale" property in AccountInfo with the properties set as specified below:

1. For US vehicles:
    a) Set the "country" property to "US".
    b) Set the "language" property to "en".
2. For Canadian vehicles:
    a) Set the "country" property to "CA".
    b) Set the "language" property as specified below:
        1) For Canadian English, set the "language" property to "en".
        2) For French, set the "language" property to "fr".

Selecting a Style

Chrome Construct includes operations for building two different types of vehicle selectors. The first type is a standard selector, which provides for selecting by year, division or subdivision, model and style. The second type is a consumer friendly selector, which provides for selecting by year, division or subdivision, and consumer friendly model name and style name.

Both types of selectors provide selection by either division or subdivision. Divisions correspond to vehicle brands such as Ford, Chevrolet, Buick, etc. Subdivisions provide an additional level of granularity below division such as Ford Cars, Chevy Pickups, etc.

A consumer friendly selector is different than a standard selector in the way it presents the Model Name and/or Style Name to the user. Manufacturers may have their own internal (non-marketed) names for models and styles with which the average consumer may not be familiar. Using Chrome Construct a selector can be created that hides those unfamiliar names and instead presents marketed names which an average consumer would recognize. For example, Subaru markets their Legacy Wagon (internal name) as the Outback. Use the target audience of your application to determine which type of selector to implement.

Each of the operations that accept a FilterRules parameter provides the ability to further customize your vehicle selector. Using filter rules you can limit what vehicle data is returned from the service by order availability (Fleet/Retail), market class, vehicle type, postal code, and MSRP.

Each type of filter specified in the FilterRules will be applied, in addition to any others that may be specified. At the very minimum, order availability must be specified, while all other filter rule elements may be left unspecified. To get a list of market classes available on which to filter use the getMarketClasses operation. For a list of vehicle types, see the VehicleType interface definition. When filtering on MSRP, you may specify a minimum price, maximum price, or both (see the MoneyRange interface definition for details).

To implement a standard selector:

1. Create a new ModelYearsRequest and set the properties of the request. See the ModelYearsRequest interface definition for details.
2. Call getModelYears with the ModelYearsRequest to get an array of available model years. The user should be able to select from the displayed available model years.
3. Do one of the following depending, on whether you want to select by division or subdivision:
   a) If selection by Division is desired:
      1) With the model year selected by the user, create a DivisionsRequest object and set the parameters.
      2) Call getDivisions with the DivisionsRequest object to get an array of Division objects. Each Division object has a name that can be displayed to the user and an ID that is used to identify the division in the request to get models.
      3) With the selected division ID, create a ModelsByDivisionRequest object and set the parameters.
      4) Call getModelsByDivision with the ModelsByDivisionRequest object to get an array of Models. Each Model object has a name that can be displayed to the user and a model ID that is used to identify the model in the request to get styles.
   b) If selection by Subdivision is desired:
      1) With the model year selected by the user, create a SubdivisionsRequest object and set the parameters.
      2) Call getSubdivisions with the SubdivisionsRequest object to get an array of Subdivision objects. Each Subdivision object has a name that can be displayed to the user and an ID that is used to identify the subdivision in the request to get models.
      3) With the selected subdivision ID, create a ModelsBySubdivisionRequest object and set the parameters.
      4) Call getModelsBySubdivision with the ModelsBySubdivisionRequest object to get an array of Models. Each Model object has a name that can be displayed to the user and a model ID that is used to identify the model in the request to get styles.
4. With the selected model ID, create a StylesRequest object and set the parameters.
5. Call getStyles with the StylesRequest object to get an array of Style objects. Each Style object provides a sampling of information about the style that can be displayed to the user. For instance style name, style ID, base prices, body type, and a URL to an image of the style are just some of the information available. See the Style interface definition for a full list.

   The Style object contains the configuration state. The configuration state is used when calling getConfiguration (and getConfigurationByStyleId, or getConfigurationByAutoBuilderStyleId) or getStyleFullyConfigured (and getStyleFullyConfiguredByStyleId, or getStyleFullyConfiguredByAutoBuilderStyleId), which provides the full set of information about a style including (but not limited to) options, standard features, paint, and technical specifications. See the section on Viewing a Style for more details.

To implement a consumer friendly selector:

1. Create a new ModelYearsRequest and set the properties of the request. See the ModelYearsRequest interface definition for details.
2. Call getModelYears with the ModelYearsRequest to get an array of available model years. The user should be able to select from the displayed available model years.
3. Do one of the following depending on whether you want to select by division or subdivision:
   a) If selection by Division is desired:
      1) With the model year selected by the user, create a DivisionsRequest object and set the parameters.
      2) Call getDivisions with the DivisionsRequest object to get an array of Division objects. Each Division object has a name that can be displayed to the user and an ID that is used to identify the division in the request to get models.
      3) With the selected division ID, create a ConsumerModelNamesByDivisionRequest object and set the parameters.
      4) Call getConsumerFriendlyModelNamesByDivision with the ConsumerModelNamesByDivisionRequest object to get an array of consumer friendly model names. Each name can be used to identify the model in the request to get styles.

5) With the selected consumer friendly model name, create a StylesByConsumerModelNameAndDivisionRequest object and set the parameters.
6) Call getStylesByConsumerFriendlyModelNameAndDivision with the StylesByConsumerModelNameAndDivisionRequest to get an array of Style objects.

b) If selection by Subdivision is desired:
1) With the model year selected by the user, create a SubdivisionsRequest object and set the parameters.
2) Call getSubdivisions with the SubdivisionsRequest object to get an array of Subdivision objects. Each Subdivision object has a name that can be displayed to the user and an ID that is used to identify the subdivision in the request to get models.
3) With the selected subdivision ID, create a ConsumerModelNamesBySubdivisionRequest object and set the parameters.
4) Call getConsumerFriendlyModelNamesBySubdivision with the ConsumerModelNamesBySubdivisionRequest object to get an array of consumer friendly model names. Each name can be used to identify the model in the request to get styles.
5) With the selected consumer friendly model name, create a StylesByConsumerModelNameAndSubdivisionRequest object and set the parameters.
6) Call getStylesByConsumerFriendlyModelNameAndSubdivision with the StylesByConsumerModelNameAndSubdivisionRequest to get an array of Style objects.

4. Each Style object provides a sampling of information about the style that can be displayed to the user. For instance, consumer friendly style name, style ID, base prices, body type, and a URL to an image of the style are just some of the information available. See the Style interface definition for a full list.

   The Style object contains the configuration state. The configuration state is used when calling getConfiguration, getStyleFullyConfigured, getStyleFullyConfiguredByStyleId, or getStyleFullyConfiguredByAutoBuilderStyleId, which provides the full set of information about a style including (but not limited to) options, standard features, paint, and technical specifications. See the section on Viewing a Style for more details.

Creating Filter Rules

FilterRules allow you to limit the results of a request by certain criteria. There are five different criteria by which you can to filter:

1. orderAvailability - This is the only FilterRule that is required parameter. This will filter the results to either Fleet or Retail vehicles.
2. postalCode - When set, the results will be filtered to vehicles available in the specified postal code (US only).
3. marketClassIds - When set, the results will be filtered to vehicles with market class IDs that match those specified in the array.
4. vehicleTypes - When set, the results will be filtered to vehicles with vehicle types that match those specified in the array.
5. msrpRange - When set, the results will be filtered to vehicles with an MSRP that falls within the range specified.

The application of the FilterRules is determined at the Style level, but the results can be seen at higher levels. For instance, when getting Divisions with FilterRules where an msrpRange of 10000 to 15000 is specified, only Divisions that contain a Style with an MSRP that falls within that range will be returned.

Note: Each of the criteria specified in FilterRules are applied in addition to (rather than either/or) the other criteria specified.

Viewing a Style

Whether you are building an application that provides vehicle information or a vehicle configuration tool, the first step will most likely be to present the user with information about the vehicle, such as style pricing, optional equipment, standard equipment, etc.

The style information is returned in the form of a Configuration object. There are multiple ways to retrieve a configuration. The method you use will depend on what information you have to start with, as well as

what you intend to do with that information. It can be retrieved by using a configuration state, a Style ID, or an AutoBuilder Style ID. If you have a Style object (see Searching for Vehicles) you have all three of these available to you. If a configuration state is used, the style information will be retrieved starting from the exact configuration that is specified in that state. If a Style ID or an AutoBuilder Style ID is used, the style information will be retrieved starting from a completely blank state (i.e., with no equipment selected).

If Chrome Construct is being used with one of Chrome's new vehicle data products, the Style ID can be obtained from the following sources:

Product Name	File Name	Column Name
NVD v2	Styles.txt	Style ID
Fleet NVD (v3)	Styles.txt	Style ID

There are two main ways to get a configuration. The standard way returns the configuration without any changes to its state. This can be used when loading configuration states (such as from a database) that have already been previously configured or for presenting style information without any configured state. The other way will (when possible) return the configuration in a fully configured (orderable) state. This is used for presenting style information for a consumer presentation, where the base equipment is already selected. Neither method will prevent the option of further configuration (see Configuring a Style).

All methods for getting a configuration accept a ReturnParameters object which provides a way to limit the amount of information returned about the style. Depending on how you structure your application and what information you need to display, this can impact response times. See the section on Limiting Return Values for more detail.

Requesting a configuration:

To request a configuration using a configuration state:

1. With the configuration state, create a ConfigurationRequest and set the parameters. See the ConfigurationRequest interface definition for details.
2. Call getConfiguration with the ConfigurationRequest to get a Configuration object.

To request a configuration using a Style ID:

1. With the Style ID, create a ConfigurationByStyleIdRequest and set the parameters. See the ConfigurationByStyleIdRequest interface definition for details.
2. Call getConfigurationByStyleId with the ConfigurationByStyleIdRequest to get a Configuration object.

To request a configuration using an AutoBuilder Style ID (Note: This method is provided for backward compatibility only, otherwise, please use Style ID):

1. With the AutoBuilder Style ID, create a ConfigurationByAutoBuilderStyleIdRequest and set the parameters. See the ConfigurationByAutoBuilderStyleIdRequest interface definition for details.
2. Call getConfigurationByAutoBuilderStyleId with the ConfigurationByAutoBuilderStyleIdRequest to get a Configuration object.

Requesting a configuration fully configured:

To request a configuration using a configuration state:

1. With the configuration state, create a FullyConfiguredRequest and set the parameters. See the FullyConfiguredRequest interface definition for details.
2. Call getConfiguration with the FullyConfiguredRequest to get a Configuration object.

To request a configuration using a Style ID:

1. With the Style ID, create a FullyConfiguredByStyleIdRequest and set the parameters. See the FullyConfiguredByStyleIdRequest interface definition for details.
2. Call getConfigurationByStyleId with the FullyConfiguredByStyleIdRequest to get a Configuration object.

To request a configuration using an AutoBuilder Style ID (Note: This method is provided for backward compatibility only, otherwise, please use Style ID):

1. With the AutoBuilder Style ID, create a FullyConfiguredByAutoBuilderStyleIdRequest and set the parameters. See the FullyConfiguredByAutoBuilderStyleIdRequest interface definition for details.
2. Call getConfigurationByAutoBuilderStyleId with the FullyConfiguredByAutoBuilderStyleIdRequest to get a Configuration object.

Using Consumer Information

Consumer Information is available in two different formats in Chrome Construct:

1. Consumer Information, such as rebates, crash test ratings. recall notices, and warranties are available as a single string if the includeConsumerInfo parameter is set to true in ReturnParameters. See ConsumerInformation for details.
2. StructuredConsumerInformation provides the same information as the ConsumerInformation described above. StructuredConsumerInformation breaks the data up into separate elements that allow you to control which elements are displayed. It also provides the ability to reliably determine the independent values of each element. For instance, if a consumer wished to determine the number of years a vehicle was warrantied for, it could be done by getting the StructuredConsumerInformation with the typeName of "Warranty" and the value property of the StructuredConsumerInformationItem that contains the name "Basic Years". This information is available if includeStructuredConsumerInfo is set to true in the ReturnParameters. See StructuredConsumerInfo and StructuredConsumerInfoItem for details.

Displaying Model and Style Names

There are two ways to display model and style names with Chrome Construct:

1. The standard detailed Chrome model and style names, which are compliant with the manufacturers' names, and
2. Consumer friendly model and style descriptions for vehicles. The consumer friendly names more closely represent how a vehicle is marketed to a consumer, and can be used in lieu of the standard Chrome model and style names.

Consumer friendly names can be used in the following ways:

a. Vehicle selectors – consumerFriendlyModelNames and consumerFriendlyStyleNames can be used as an alternate method for consumers for select and search for vehicle information. The Style object will return the requested values.

b. You can also use the consumerFriendlyDrivetrain and consumerFriendlyBodyType fields to group like styles, such as grouping returned vehicles by “Four Wheel Drive” and/or “Sport Utility”.

Using Editorial Content (Chrome Reviews)

EditorialContent provides review content for the vehicle at the model level. To return Editorial Content, set the includeEditorialContent property in ReturnParameters object to true. Each vehicle review includes vehicle strengths, changes, the value the vehicle brings to the consumer, and an overview. For additional information, see EditorialContent.

Note: A vehicle may have multiple reviews from different sources, so use EditorialContentSource to retrieve the source information.

Limiting Return Values

When getting style information and toggling options, the return results can be limited to only return data that is relevant to your application. For example, on a web page that just displays a style description, style pricing, and standard equipment, you may want to suppress the return of all other data items by setting the respective include attribute to false or null for the entire tree when creating the request. The advantage of limiting the amount of information returned is increased performance, since less data needs to be marshaled and sent over the network.

Below is a list of all of the data items that can be limited in the return value of a Configuration:

- Standard Equipment
- Optional Equipment and Descriptions
- Special Equipment Options
- Colors – All Colors/Only Colors Valid with Current Configuration
- Editorial Content
- Consumer Information
- Configuration Check List
- Additional Images
- Technical Specifications
- Filtered Technical Specification Titles

See the interface definition for ReturnParameters for more information.

Configuring a Style

Chrome Construct allows vehicles to be configured using the toggleOption and selectColor interfaces.

The typical workflow for vehicle configuration is to display a list of options available for selection and allow the user to choose options. The initial list is retrieved using the operations described in the section on Viewing a Style. As the user chooses an option, a call is made to the toggleOption operation. This operation returns a new Configuration instance that contains the updated ConfigurationState, as well as the updated option availability, pricing and technical specification values. The ConfigurationState itself contains all information required to identify the vehicle including the Style ID, fleet/retail mode, SEO visibility setting, option selections and color selections. To initiate the next toggleOption or selectColor transaction, pass in the ConfigurationState from the previous call.

Toggling Options and Refreshing Dynamic Data

Options can be selected through the toggleOption interface. The toggleOption interface sends a request to the service to select an option. The service will look at the current ConfigurationState that is passed in as part of the request to determine if the option selection is valid. The response to the toggle includes a status, the updated configuration state for use in the next toggle, the updated style, and conflict information if a conflict occurred. The toggleOption interface will return the updated Configuration to eliminate the need to make a subsequent call to getStyleFullyConfigured. This updated Configuration contains the ConfigurationState that will be passed to the next toggleOption request.

The toggle option works as follows:

1. Create a new ToggleOptionRequest.
2. Set the parameters of the request as specified in the interface definition section. Note that the ChromeOptionCode being set is the option code that the user is requesting to be toggled.
3. If this is the first toggle, set the ConfigurationState to the ConfigurationState returned from getConfiguration or from the ConfigurationState of the Configuration returned from getStyleFullyConfigured. If this is a subsequent toggle or a conflict resolving toggle, set the ConfigurationState to the ConfigurationState returned as part of the Configuration referenced in the ToggleOptionResponse from the previous toggleOption call.
4. Once all parameters have been bound, pass the ToggleOptionRequest to toggleOption.
5. This will return a ToggleOptionResponse that includes the new Configuration and information regarding the toggle itself. If the resulting option is successful, the ToggleOptionResponseStatus will be “None” and no conflict resolving options will be returned. See “Handling Toggle Conflicts” for information on how to handle conflicts.
6. Refresh all dynamic data items displayed to the user using the Configuration returned as part of the ToggleOptionRequest. The style price, option state, option price, and technical specifications may change after each toggle. If you are displaying any of these data items, refresh the data from the returned style whenever an option is toggled.

7. Display the option availability to the user. After each option toggle, the availability of all options is updated. This availability is represented as the ConfigurationState. Use the ConfigurationState to indicate to the user if the option is now selected, unselected, included, required, excluded, or upgraded. Note that the toggleOption interface will support the toggling of an option in any state including "excluded".

To improve performance and limit the amount of data returned with each ToggleOptionRequest, set returnDeltaConfiguration to true in the toggleOption operation. This will return only the options and attributes that were changed with the last toggleOption. Since the response will contain only the delta information, it is expected that the client application is able to persist the style information and have the ability to update it.

Handling Toggle Conflicts

In the event that an option toggle results in a conflict, here are the steps for handling the conflict:

1. If the ToggleOptionResponseStatus is "UserChoiceNeeded":
   a. Display to the user the conflict resolving options just as you would on the configuration page using the option code to identify the appropriate option description to display.
   b. The conflict resolving option is selected by calling toggleOption again using the ConfigurationState from the Configuration returned from the last ToggleOptionResponse.
   c. This process should be continued until the OptionToggleStatus is "None".
2. If the ToggleOptionResponseStatus is "UserConfirmationNeeded" or "UserChoiceNeeded", and the affected options are returned, the options specified in additionAffectedChromeOptionCodes and deletionAffectedChromeOptionCodes should be shown to the user. These are options that were affected by the toggleOption action and should be presented to the user to confirm.
3. If the ToggleOptionResponseStatus is "None", then no further action is needed.

To implement the capability to cancel the action of a toggleOption, retain the configuration state that was originally passed as a parameter to the toggleOption request. If the toggleOption is confirmed by the user, set the configuration state to the new configuration state returned as part of the ToggleOptionResponse. If the action is canceled, continue with the original configuration state from prior to the toggle.

Building a Configuration Checklist

The configuration checklist provides groups of options that need to be selected from in order for the vehicle configuration to be considered orderable.

To build a checklist:

1. On the Configuration, iterate the ConfigurationCheckListItem array.
2. Each ConfigurationCheckListItem object contains the option codes that would satisfy that group and an attribute called satisfy that returns true when at least one option in the group is selected. Using the options from Configuration, displays the options to the user.
3. Once all option groups have been satisfied, the fullyConfigured attribute of the ConfigurationState for the Configuration will return true.

Note: If your intention is to pre-configure vehicles upon initial load, please see the interface definitions for getStyleFullyConfigured, get StyleFullyConfiguredByStyleId, and getStyleFullyConfiguredByAutoBuilderStyleId operations.

Selecting Colors

Chrome Construct supports selecting colors for the purpose of saving the state of color selections. Color selections will not affect pricing or the selection of options. Color selections are made by calling selectColor.

ColorCombinations are returned as part of the Configuration. Based on the setting of the ReturnParameters.includeInvalidColors specified as part of the ReturnParameters, the color combinations will include either all color combinations or only combinations available with the current configuration.

Each color combination contains an attribute called valid. When the color combination is valid, the color combination is currently allowed as configured. When the color combination is invalid, the color combination is available on the vehicle but not as currently configured. Even when a color combination is not valid, that color combination can still be selected to allow the user's choice to be reflected in the configuration state. Color combination selections are treated as mutually exclusive of each other. The last selection will replace any prior selections.

To select a color:

1. Create a SelectColorRequest. See the interface definition for parameters to set. Note that the selection is made by passing in the combinationColorId, the secondaryColorId (if present), and the auxiliaryChoiceColorId (if present).
2. Pass the SelectColorRequest to the selectColor interface.
3. The returned Configuration will reflect the color selections in the ConfigurationState. If you iterate the ColorCombinations, the selected attribute of the selected ColorCombination will be true.

Note: A secondary color ID and auxiliary color ID can be selected only if the parent combination color ID is selected.

To deselect a color combination:

- Pass a zero or null value in place of the color combination ID. This will effectively return you to an unselected state for all colors.

To deselect an auxiliary or secondary color:

- Pass a zero or null value in place of the auxiliary color ID or the secondary color ID. This will return you to an unselected state for the item.

Serializing Configuration State

The ConfigurationState has an attribute called serializedValue. The serializedValue is an XML representation of the configuration state. This serializedValue can be used to initialize the ConfigurationState.

The serialized configuration state can be used for the following purpose:

- Implementing save/retrieve—the configuration state can be persisted and used at a later point in the application to retrieve the configuration details.
- Pass configuration state as a parameter in a web application—in cases where it is not desirable to pass the configuration state as an object, the XML string can be passed instead.

Using the Serialized Value:

1. Call materializeConfigurationState to pass in your serialized value.
2. Call any method that takes a ConfigurationState.

Side-by-Side Comparison

Side-by-side comparison evaluates the categories (generic equipment descriptions) and technical specifications of two or more vehicles. By default, the side-by-side comparison returns all categories and all technical specifications that are available to the vehicles included in the comparison.

To initiate a side-by-side comparison:

1. Create a new array of ConfigurationState objects that contain all of the style states of the styles that will be compared.
2. Create a new SideBySideComparisonRequest.

3. On the SideBySideComparisonRequest, set the parameters of the request as described in the interface definition.
4. Call compareSideBySide with the SideBySideComparisonRequest.
5. Use the SideBySideComparisonResult to access the comparison groups and items for display.

Optionally, to limit by type (category or technical specification):

1. To limit the results of the comparison to just technical specifications, set includeCategoryComparisons to false.
2. To limit the results of the comparison to just categories, set includeTechnicalSpecificationComparisons to false.

Optionally, to limit to specific categories or technical specification values:

1. To limit the results to specific technical specification values, set the includeTechnicalSpecificationTitleIds int array to the technical specification title IDs that should be returned.
2. To limit the results to specific categories, set the includeCategoryIds int array to the category IDs that should be returned.

Note: To find available categories and technical specifications, use the getCategoryDescriptions and getTechnicalSpecificationValues operations, respectively. If the technical specification title is not available for any of the vehicles being compared, that item will be omitted from the result set even if explicitly requested.

Using the Side-By-Side Comparison Response

The side-by-side comparison response will contain groups of comparable items. Each group is a set of technical specifications or categories that are logically grouped together. The group has a group name that can be used for display purposes. Within each group there are 1..n items where an item represents a single technical specification ID or category ID.

For a more detailed explanation of each operation and type, refer to the interface definition section of this document.

Advantage Based Comparison

Advantage based comparisons return comparison information in the form of natural language descriptions.

Initiating an advantage based comparison:

1. Create a new AdvantageComparisonRequest.
2. Set the pivot configuration style.
3. Set the array of target comparison configuration styles of the vehicles you would like to compare against the pivot configuration.
4. On the AdvantageComparisonRequest, set the parameters as described in the interface definition.
5. Call compareAdvantages with the AdvantageComparisonRequest request.

See ReturnParameters for what information is returned. See the section on Viewing a Style for directions on how to get a configuration state. Note: A comparison can be performed on any configuration state.

Using Auto-Equip

Since each vehicle has its own standard features and optional equipment, it can sometimes be difficult to do a valid/useful comparison between vehicles, without first configuring each to a similar state. For instance, if you wanted to compare a style that comes standard with a V6 engine to a style that has a V6 option but comes standard with a 4-cylinder engine. With auto-equip you provide one style as the "source" style and an array of styles as the "target" styles. Auto-equip will take each of the "target" styles and automatically equip them (as closely as possible) to be similar to the "source" style. In the example stated

above, this would mean that each style passed as a "target" would automatically be configured with a V6 engine (if available).

To use auto-equip:

1. Choose a configuration state for the others to be made similar to. This may be any configuration state retrieved from any Style object. This is referred to as the source.
2. Choose one or more configuration states that you wish to be made similar to the source. This may be any configuration state(s) retrieved from any Style object(s). These are referred to as the targets.
3. Create a new AutoEquipRequest using the selected source and target configuration states. See the AutoEquipRequest interface definition for details.
4. Call the autoEquipStyles method with the AutoEquipRequest to get an array of ConfigurationState objects. There will be one for each of the targets that were defined previously (in step 2) each of which will be configured as closely as possible to the source configuration state. These configuration states may then be used for any other method of the service that accepts configuration states, including the comparison methods.

Searching for Vehicles

Chrome Construct provides the ability to search for vehicles against a wide array of customizable criteria. While this may be one of the more complicated features to use, it is also one of the most useful. Using search you can build applications that allow a user to find vehicles with and/or without specific features and attributes. This allows for building applications that target consumers who have a good idea about what they want in a vehicle, but aren't too sure about what brand or model they want. In addition, it provides a means for creating customized features to your applications. See Finding Comparable Vehicles for an example of using search to implement this feature.

About Search Criterion Objects

There are three different objects used for defining the criteria for searches. They are SearchCriterion, AndCriterion, and OrCriterion. Search allows for an unlimited number and combination of these objects. SearchCriterion (the most basic criterion) allows you to specify whether or not a vehicle must have something. For example you could create a SearchCriterion to search for all vehicles that have a V6 engine. OrCriterion allows you to specify an array of SearchCriterion and will evaluate whether or not at least one of the SearchCriterion applies. For instance, you could create an OrCriterion to search for all vehicles that either have a V6 engine or an 8 cylinder engine. The AndCriterion allows you to specify an array of SearchCriterion and will evaluate whether or not all of the SearchCriterion applies. For instance, you could create an AndCriterion to search for all vehicles that have both a V6 engine and an 8 cylinder engine (available as different options).

Defining Search Criterion Object Properties

The SearchCriterion object, whether used directly or as part of an OrCriterion or AndCriterion, requires you to define four things: a token name, a criterion type, an importance, and the value (or range of values) to search for. Below is an explanation of each property of the SearchCriterion object:

1. name - This is the search token name and defines which attribute of a vehicle that you are searching for. See the interface definition for SearchTokenName for a list of searchable attributes or retrieve a list dynamically by using the getSearchCriterionDescriptors method. The token name should be the first property you define, since it affects the other search criterion properties.
2. type - There are five different search criterion types (see SearchCriterionType). These types determine how you'll need to define the value(s) to search for. The type you use will be specified by the search token name you choose. See the interface definition for SearchTokenName to lookup the type for the token name you've chosen. This information is also available dynamically via the getSearchCriterionDescriptors method of the Chrome Construct service.
    a) Boolean - When creating a search criterion with this type, set the value property of the SearchCriterion to either "true" or "false". These types of search tokens typically have a name containing the word "has", for instance "hasPowerSteering". This will help you determine which way to set the value. If you want to search for a vehicle which has power steering, then set the value to "true", otherwise set it to "false".

    Note: Do not set the min and max properties on the SearchCriterion for this type.

b) String - This type is for searching for a specific value. When creating a search criterion with this type, set the value property of the SearchCriterion to the specific value you are searching for. To determine the possible values, see the interface definition for SearchTokenName and find the token name you've chosen. For some token names, the possible values are also available dynamically via the getSearchCriterionDescriptors method.

Note: Do not set the min and max properties on the SearchCriterion for this type.

c) NumberRange - This type is for searching for a number that falls within a certain range. When creating a search criterion with this type, set the min and/or max properties of the SearchCriterion as desired. Setting only the min will mean "at least" the number you specify. Setting only the max will mean "at most" the number you specify. Setting both will mean "no less than" the min and "no more than" the max or the numbers you specify.

Note: Do not set the value property on the SearchCriterion for this type.

d) MoneyRange - This type is for searching for a price that falls within a certain range. When creating a search criterion with this type, set the min and/or max properties of the SearchCriterion as desired. Setting only the min will mean "at least" the price you specify. Setting only the max will mean "at most" the price you specify. Setting both will mean "no less than" the min and "no more than" the max or the prices you specify.

Note: Do not set the value property on the SearchCriterion for this type.

e) TechnicalSpecificationRange - This type is for searching for a technical specification value that falls within a certain range. When creating a search criterion with this type, set the min and/or max properties of the SearchCriterion as desired. Setting only the min will mean "at least" the number you specify. Setting only the max will mean "at most" the number you specify. Setting both will mean "no less than" the min and "no more than" the max of the numbers you specify.

Note: Do not set the value property on the SearchCriterion for this type.

3. importance - There are only two different search importance types (see SearchImportanceType): MustHave and MustNotHave. While it may be obvious what the terms mean, thorough care should be used in determining which one should be selected. For instance, if searching for a Boolean type token of "hasPowerWindows" with a value of "false" (meaning vehicles without power windows) and you set the search importance to MustNotHave, your search would return only vehicles with power windows. For this reason, we suggest always using an importance of MustHave for Boolean type searches.
4. value - This property must be set for Boolean and String type search criterions. It represents the value of the attribute you are searching for.
5. min - This property is used for the NumberRange, MoneyRange, and TechnicalSpecificationRange types of search criterions. It represents the minimum value of the attribute you are searching for.
6. max - This property is used for the NumberRange, MoneyRange, and TechnicalSpecificationRange types of search criterions. It represents the maximum value of the attribute you are searching for.

The OrCriterion object requires you to define two things: an array of search criterion and an importance. Below is an explanation of each property of the OrCriterion object:

1. criteriaArray - This is an array of SearchCriterion objects. The search will evaluate every vehicle against each criterion in this array. If any of the SearchCriterion in the OrCriterion array applies to the vehicle, it will be considered a match. For simplicity, we suggest always using the same importance on each SearchCriterion in this array (i.e., all MustHave or all MustNotHave). This will help in determining the importance to set for the OrCriterion itself.
2. importance - There are only two possible values for importance (see SearchImportanceType) which are MustHave and MustNotHave. While it may be obvious what the terms mean, thorough care should be used in determining which one should be selected. For instance if there were two SearchCriterion in the criteriaArray, one as a MustHave-hasAirConditioningFront set to true and one as MustHave-hasAirConditioningRear set to true, then setting the importance on the OrCriterion to MustHave would mean that the search will find vehicles that have air conditioning in either the front and/or the rear. However, setting the importance on the OrCriterion to MustNotHave would mean that the search will find only vehicles that do not have air conditioning in the front or the rear.

The AndCriterion object requires you to define two things: an array of search criterion and a search token name. Below is an explanation of each property of the AndCriterion object:

1. criteriaArray - This is an array of SearchCriterion objects. The search will evaluate every vehicle against each criterion in this array. If every SearchCriterion in the array applies to the vehicle, it will be considered a match. Every SearchCriterion in the array must have the same search token name as

the AndCriterion does. In addition, only String and NumberRange type criterions are allowed in this array.

2. name - This is the search token name and is what defines which attribute of a vehicle that you are searching for. Every SearchCriterion in the criteria array must have the same search token name. See the interface definition for SearchTokenName for a list of searchable attributes, or retrieve a list dynamically by using the getSearchCriterionDescriptors method.

Performing Searches

To do a search, a SearchServiceRequest object is needed. This is the object that will contain any criteria you want for your search and define any limitations on the results. The request accepts an array for each of the three different criterion objects (SearchCriterion, AndCriterion, and OrCriterion). At least one of these objects must be populated with at least one criterion. However, any amount and combination is allowed beyond that. The search will gather all of the criterion objects from each of the three different arrays. Only vehicles that meet every criterion defined will be returned by the search. There is only one exception to that rule, which will be explained in the note below. The following explanation of each property of the SearchServiceRequest object:

1. criteriaArray - Set this property with an array of all of the SearchCriterion you want to search against. To be clear, these are only standalone SearchCriterion that you've defined, not ones that are used in an OrCriterion or AndCriterion. The search will ensure that each vehicle returned meets every criterion specified in this array.

   Note: One exception for the searching with this array, is when you've defined multiple SearchCriterion with the same SearchTokenName. In this case the search will interpret this as an OR condition. However, for such cases we suggest simply using an OrCriterion instead. If you truly meant that the vehicle should have both/all, you'll need to use an AndCriterion instead.

2. orCriteriaArray - Set this property with an array of all of the OrCriterion you want to search against. The search will ensure that each vehicle returned meets every criterion specified in this array.

3. andCriteriaArray - Set this property with an array of all of the AndCriterion you want to search against. The search will ensure that each vehicle returned meets every criterion specified in this array.

4. filterTBD - This property only has an affect when you are searching with any TechnicalSpecificationRange criterion types. In some cases there may be technical information on a vehicle that is currently unknown (i.e., To be determined or TBD). When searching for vehicles on technical specifications the search will automatically include those vehicles, since it is not known whether or not it matches the criteria you have specified. However, this property allows you to turn that feature off (by setting the value to true) so those vehicles are not returned. For example, if you were searching for "fuelEconomyCity" with a range of 20 to 30 and you had filterTBD set to true, you would only get vehicles for which the fuel economy is available and it fell within the range. If you had it set to false, you would those as well as any vehicles that did not have any fuel economy defined.

5. filterByPostalCode - This property tells the search whether or not to limit the vehicles returned by the postal code specified by the postalCode property. If you wish to limit the vehicles by postal code, set this property to true.

6. postalCode - This property allows you to define a postal code to limit the search by. If you wish to do so, set this property to the postal code desired and make sure to set the filterByPostalCode property to true.

7. maxNumResults - The property defines the maximum results the search should return. The lower you set this number, the better performance you'll get on your search. The maximum this can be set to is 100. Not setting this property, setting it to less than zero, or setting it to greater than 100 will result in the server using the default value of 100. When searching for styles, this number applies to the number of styles returned. When searching for models it applies to the number models returned.

To search for styles:

1. Create a new SearchServiceRequest and set the properties of the request. This is the object that will contain the search criterion discussed above.

2. With the SearchServiceRequest create a new SearchStylesRequest and set the properties of the request. See the SearchStylesRequest interface definition for details.

3. Call the searchStyles method with the SearchStylesRequest to get an array of Style objects. You can use these styles just like any other retrieved from Chrome Construct. See Viewing a Style or Configuring a Style for more details.

To search for models:

1. Create a new SearchServiceRequest and set the properties of the request. This is the object that will contain the search criterion discussed above.
2. Create a SearchModelsRequest and set the properties of the request. See the SearchModelsRequest interface definition for details.
3. Call the searchModels method with the SearchModelsRequest to get an array of ModelSearchResult objects. Each of the result objects contains a Model object as well as an array of Style objects from that Model that match the search criteria. The model contains some basic information about the vehicle that can be presented to the user, such as division name, model name, and an ID that can be used to uniquely identify the model. See the Model interface definition for more details. The Style objects can be used just like any other retrieved from Chrome Construct. See Viewing a Style or Configuring a Style for more details. A typical workflow when searching for models would be to present the model information first to the user and allow them to select one. Then once a model has been selected, the styles for that model can be presented to the user.

Validating a Search

Chrome Construct provides a method that will validate whether a search request is properly formed. Using any SearchServiceRequest object that you've created in preparation for executing a search, you can call the validateSearch method. This method will check each search criterion in the request and validate four things:

1. It will make sure the SearchTokenName defined is supported.
2. It will verify that the criterion type defined is the correct type for the SearchTokenName defined.
3. It will also verify that any range search types have valid min and/or max values.
4. Finally, it will verify that any range search types have min and/or max values that are not outside of the allowable range.

After completing the validation, the method will return a list of errors (see SearchCriterionError) found within the search request. If there are no errors returned, it was determined that your search request was valid.

NOTE: A valid (correctly constructed) search does not mean that the search will return any vehicles. It is possible to construct a valid search with criteria that do not apply to any vehicles, and therefore may return no results.

Using Search Criterion Descriptors

Chrome Construct provides a way to dynamically retrieve information about searchable vehicle attributes. The getSearchCriterionDescriptors method will return an array of SearchCriterionDescriptor, one for each SearchTokenName. Each descriptor will define the SearchCriterionType of the search token. If the type is TechnicalSpecificationRange, then a minimum and maximum value will also be specified, as well as the unit of measure for the technical specification. If the type is String, for some search tokens (see the interface definition for SearchTokenName), a list of possible values will be provided. Those values are returned as SearchValue objects each of which has an ID and a value. Use the value for display to the user and use the ID in your search criterion when searching.

Finding Comparable Vehicles

One of the many possible uses you can make of the search functionality of Chrome Construct is to find comparable vehicles to one you've chosen and/or configured. Building this feature into your application allows your users the capability to examine other options for the type of vehicle they are interested in. For instance, if a user knows they like the Toyota Camry, but would like to know if there are other vehicles similar in features, equipment, and/or pricing, a find comparable feature would save them from either creating their own complicated search or traversing a selector to examine every possible vehicle. Below is an example how to create a find comparable feature using search.

Since each application may have a different concept of "comparable", this is purely a guideline and may be customized for your own purposes. The following will only outline the criteria necessary for building the search. For help on running the search, see Searching for Vehicles.

To find comparable vehicles:

1. First, you'll need information about the vehicle you want to find comparable to. For this example you'll need a Configuration object with technical specifications included. If you only have a Style or ConfigurationState object, see Viewing a Style for information on obtaining a Configuration object.
2. Create an array of SearchCriterion objects containing the following:
   a. For the same model year or newer, create a SearchCriterion with the following property settings:
      - name = SearchTokenName.year
      - importance = SearchImportanceType.MustHave
      - type = SearchCriterionType.NumberRange
      - min = (set this to the value of configuration.style.modelYear)
   b. For the same market class, create a SearchCriterion with the following property settings:
      - name = SearchTokenName.marketClassId
      - importance = SearchImportanceType.MustHave
      - type = SearchCriterionType.String
      - value = (set this to the value of configuration.style.marketClassId)
   c. For the same number of doors, create a SearchCriterion with the following property settings:
      - name = SearchTokenName.numberOfDoors
      - importance = SearchImportanceType.MustHave
      - type = SearchCriterionType.String
      - value = (set this to the value of configuration.style.passengerDoors)
   d. For the same passenger capacity, create a SearchCriterion with the following property settings:
      - name = SearchTokenName.passengerCapacity
      - importance = SearchImportanceType.MustHave
      - type = SearchCriterionType.TechnicalSpecificationRange
      - min = (find the TechnicalSpecification in configuration.technicalSpecifications with a titleId of 8, then set this to the technicalSpecification.value)
      - max = (set this to the same value as min)
   e. For a similar wheelbase (within 10% range):
      1) First determine whether this criterion is applicable. This can be done by examining the market class. Check to see if the configuration.style.marketClassId matches any of these values: 1, 2, 3, 4, 5, 6, 9, 12, 14, 16, 18, 65, 66, or 99. These are market class IDs that represent light and medium duty trucks, some vans, and specialty vehicles. This information can be determined by using the getMarketClasses method of the Chrome Construct service. If the ID matches one of those listed, then a wheelbase criterion is applicable.
      2) If determined to be applicable, then create a SearchCriterion with the following property settings.
         - name = SearchTokenName.wheelbase
         - importance = SearchImportanceType.MustHave
         - type = SearchCriterionType.TechnicalSpecificationRange
         - min = (find the TechnicalSpecification in configuration.technicalSpecifications with a titleId of 301, then set this to the technicalSpecification.value multiplied by 0.9)
         - max = (find the TechnicalSpecification in configuration.technicalSpecifications with a titleId of 301, then set this to the technicalSpecification.value multiplied by 1.1)

f. For similar MSRP (within 10% range), create a SearchCriterion with the following property settings:

- name = SearchTokenName.msrp
- importance = SearchImportanceType.MustHave
- type = SearchCriterionType.MoneyRange
- min = (set this to the value of configuration.configuredTotalMsrp multiplied by 0.9)
- max = (set this to the value of configuration.configuredTotalMsrp multiplied by 1.1)

g. To prevent the same exact model from appearing in search results, create a SearchCriterion with the following property settings.

- name = SearchTokenName.modelId
- importance = SearchImportanceType.MustNotHave
- type = SearchCriterionType.String
- value = (set this to the value of configuration.style.modelId)

3. Create an array of OrCriterion objects containing the following:

a. To limit which divisions are searched:

1) First determine which divisions you wish to search against. This could either be hard-coded in your application, or a list could be presented to the user to choose from. A list of divisions and their corresponding IDs are available through the getDivisions method of the Chrome Construct service. The search is done using the division ID, so make sure to capture that information. If you want all divisions to be searched, you could simply skip this OrCriterion altogether.

2) Once a list of division IDs is gathered, then create one SearchCriterion for each division with the following property settings:

- name = SearchTokenName.divisionId
- importance = SearchImportanceType.MustHave
- type = SearchCriterionType.String
- value = (set this to a division ID)

3) Create an OrCriterion with the following property settings:

- importance = SearchImportanceType.MustHave
- criteriaArray = (set this to an array containing each of the divisionId SearchCriterion you created in the previous step).

b. For similar body types:

1) Get the bodyTypeId from each BodyType object in configuration.style.bodyTypes

2) Create one SearchCriterion for each body type ID with the following property settings:

- name = SearchTokenName.bodyType
- importance = SearchImportanceType.MustHave
- type = SearchCriterionType.String
- value = (set this to a body type ID)

3) Create an OrCriterion with the following property settings:
    - importance = SearchImportanceType.MustHave
    - criteriaArray = (set this to an array containing each of the bodyType SearchCriterion you created in the previous step)

4. You now have all the criteria necessary to do a find comparable search. See Searching for Vehicles for details on how to execute the search.

C. Chrome Construct Service Interface Definitions

This section defines each of the operations and types available through the Chrome Construct Service.

Operations

1. getModelYears

This operation is used to get all model years of vehicles for the specified filter rules. This operation would be used in building a vehicle selector. The years returned by this operation are intended to be used as parameters to the getDivisions, getSubdivisions, and other operations, such as getConsumerFriendlyModelNamesByDivision.

INPUT: ModelYearsRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: IntArrayElement

Element Name	Type	Example	Description
i	Array of int	2006, 2007, 2008	A list of filtered model years.

2. getDivisions

This operation is used to get all divisions for the specified filter rules and model year. This operation would be used in building a vehicle selector. The divisions returned as part of the Division object is intended to be used as a parameter to getModels.

INPUT: DivisionsRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
modelYear	int	2007	Specifies the model year to filter to.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: DivisionArrayElement

Element Name	Type	Example	Description
division	Array of Division	See Division	A list of filtered divisions for the model year.

3. getSubdivisions

This operation is used to get all subdivisions for the specified filter rules and model year. This operation would be used in building a vehicle selector. The subdivisions returned as part of the Subdivision object is intended to be used as a parameter to getModelsBySubdivision.

INPUT: SubdivisionsRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
modelYear	int	2007	Specifies the model year to filter to.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: SubdivisionArrayElement

Element Name	Type	Example	Description
subdivision	Array of Subdivision	See Subdivision	A list of filtered subdivisions for the model year.

4. getModelsByDivision

This operation is used to get all models for the specified filter rules, model year, and division. This operation would be used in building a vehicle selector. The DivisionID returned as part of the Model object is intended to be used as a parameter to getModelsByDivision.

INPUT: ModelsByDivisionRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
modelYear	int	2007	Specifies the model year to filter to.
divisionId	int	8	Specifies the ID of the division to filter to. Use this ID or the subdivision ID, depending on the type of selector being implemented.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: ModelArrayElement

Element Name	Type	Example	Description
model	Array of Model	See Model	A list of filtered models for the model year, and division.

5. getModelsBySubdivision

This operation is used to get all models for the specified filter rules, model year, and subdivision. This operation would be used in building a vehicle selector. The subdivisions returned as part of the Model object is intended to be used as a parameter to getModelsBySubdivision.

INPUT: ModelsBySubdivisionRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
modelYear	int	2007	Specifies the model year to filter to.
subdivisionId	int	4661	Specifies the ID of the subdivision to filter to.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: ModelArrayElement

Element Name	Type	Example	Description
model	Array of Model	See Model	A list of filtered models for the model year, division, and subdivision.

6. getConsumerFriendlyModelNamesByDivision

This operation is used to get consumer friendly model names for the specified model year and division. The Consumer Friendly Model Name more closely represents how the vehicle is marketed to the consumer, and can be used in lieu of the Model Name. This operation would be used in building a vehicle selector to getStyleFullyConfiguredByConsumerFriendlyModelNameAndDivision.

INPUT: ConsumerModelNamesByDivisionRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
modelYear	int	2007	Specifies the model year to filter to.
divisionId	int	8	Specifies the ID of the division to filter to.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: StringArrayElement

Element Name	Type	Example	Description
s	Array of String	Avalanche, Aveo, Cobalt, Colorado, Corvette, Equinox, Express Cargo Van, Express Commercial Cutaway, Express Passenger, HHR, Impala…	All filtered models, listed by consumer friendly model names.

7. getConsumerFriendlyModelNamesBySubdivision

This operation is used to get consumer friendly model names for the specified model year and subdivision. The Consumer Friendly Model Name more closely represents how the vehicle is marketed to the consumer, and can be used in lieu of the Model Name. This operation would be used in building a vehicle selector to getStylesByConsumerFriendlyModelNameAndSubdivision.

INPUT: ConsumerModelNamesBySubdivisionRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
modelYear	int	2007	Specifies the model year to filter to.
subdivisionId	int	4661	Specifies the ID of the subdivision to filter to.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: StringArrayElement

Element Name	Type	Example	Description
s	Array of String	Avalanche, Colorado, Silverado 1500, Silverado 1500 Classic, Silverado 1500 Classic Hybrid, Silverado 1500HD Classic, Silverado 2500HD, Silverado 2500HD Classic, Silverado 3500 Classic, Silverado 3500HD, Silverado SS Classic	All consumer friendly model names within the subdivision.

8. getStylesByConsumerFriendlyModelNameAndDivision

This operation is used to get all styles for the specified filter rules, model year, and division. The Consumer Friendly Model Names are returned from getConsumerFriendlyModelNamesByDivision.

INPUT: StylesByConsumerModelNameAndDivisionRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
modelYear	int	2007	Specifies the model year to filter to.
divisionId	int	8	Specifies the ID of the division to filter to.
consumerFriendlyModelName	String	Silverado 1500	This is the consumer friendly model name.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: StyleArrayElement

Element Name	Type	Example	Description
style	Array of Style	See Style	All styles (by consumer friendly name) within the division and model.

9. getStylesByConsumerFriendlyModelNameAndSubdivision

This operation is used to get all styles for the specified filter rules, model year, and subdivision. The Consumer Friendly Model Names are returned from getConsumerFriendlyModelNamesBySubdivision.

INPUT: StylesByConsumerModelNameAndSubdivisionRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
modelYear	int	2007	Specifies the model year to filter to.
subdivisionId	int	4661	Specifies the ID of the subdivision to filter to.
consumerFriendlyModelName	String	Silverado 1500	This is the consumer friendly model name.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: StyleArrayElement

Element Name	Type	Example	Description
style	Array of Style	See Style	All styles (by consumer friendly name) within the subdivision and model.

10. getStyles

This operation is used to get all styles for the specified filter rules and model ID. This operation would be used in building a vehicle selector.

INPUT: StylesRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
modelId	int	16615	The model ID of the model.
filterRules	FilterRules	See FilterRules	The rules by which the response is filtered.

OUTPUT: StyleArrayElement

Element Name	Type	Example	Description
style	Array of Style	See Style	Returns the style objects of available styles for the specified model ID.

11. getConfiguration

This operation should be used to get a configuration object, which contains all of the information available for a style, such as technical specifications, colors, configuration state, etc. You would use this when you have a ConfigurationState from a prior transaction and you don't want to pre-select options.

INPUT: ConfigurationRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
configurationState	ConfigurationState	See ConfigurationState	The configuration state of the style to retrieve.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.

OUTPUT: ConfigurationElement

Element Name	Type	Example	Description
configuration	Configuration	See Configuration	The style information populated to the depth specified in the ReturnParameters.

12. getConfigurationByStyleId

This operation should be used to get a configuration object, which contains all of the information available for a style, such as technical specifications, colors, configuration state, etc. You would use this when you have a Style ID and you don't want to pre-select options.

INPUT: ConfigurationByStyleIdRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
orderAvailability	OrderAvailability	See OrderAvailability	Specifies whether or not to include vehicles in fleet mode or retail mode.
styleId	int	285716	Unique identifier (primary key) assigned by Chrome. Style ID does not change once it is assigned to a vehicle.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.

OUTPUT: ConfigurationElement

Element Name	Type	Example	Description
configuration	Configuration	See Configuration	The style information populated to the depth specified in the ReturnParameters.

13. getConfigurationByAutoBuilderStyleId

This operation should be used to get a configuration object, which contains all of the information available for a style, such as technical specifications, colors, configuration state, etc. You would use this when you have an AutoBuilder Style ID and you don't want to pre-select options. Note: AutoBuilderStyleId is provided for backward compatibility only.

INPUT: ConfigurationByAutoBuilderStyleIdRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
orderAvailability	OrderAvailability	See OrderAvailability	Specifies whether or not to include vehicles in fleet mode or retail mode.
autoBuilderStyleId	String	w2007k10m21t1	The AutoBuilder Style ID for the style.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.

OUTPUT: ConfigurationElement

Element Name	Type	Example	Description
configuration	Configuration	See Configuration	The style information populated to the depth specified in the ReturnParameters.

14. getStyleFullyConfigured

This operation is used to retrieve configuration information for a style. This operation can be used for fetching a base style or a previously configured style. If the ConfigurationState passed contains option and color selections, the resulting Configuration will have the options and colors selected. When configuring a vehicle, toggleOption and selectColor will return the same style information as getStyleFullyConfigured. As such, when using toggleOption or selectColor, it is not necessary to call getStyleFullyConfigured again.

Turning on the toggleOptionResponse is preferred, as it allows you to handle option conflicts that may occur when attempting to fully configure a style.

INPUT: FullyConfiguredRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
configurationState	ConfigurationState	See ConfigurationState	The configuration state of the style to retrieve.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.

OUTPUT: ToggleOptionResponse

Element Name	Type	Example	Description
toggleOptionResponse	ToggleOptionResponse	See ToggleOptionResponse	This provides an updated ConfigurationState.

15. getStyleFullyConfiguredByStyleId

This operation is used to get a fully configured style by Style ID.

INPUT: FullyConfiguredByStyleIdRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
orderAvailability	OrderAvailability	See OrderAvailability	Specifies whether or not to include vehicles in fleet mode or retail mode.

Element Name	Type	Example	Description
styleId	String	285716	Unique identifier (primary key) assigned by Chrome. Style ID does not change once it is assigned to a vehicle.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.

OUTPUT: ToggleOptionResponse

Element Name	Type	Example	Description
toggleOptionResponse	ToggleOptionResponse	See ToggleOptionResponse	This provides an updated ConfigurationState.

16. getStyleFullyConfiguredByAutoBuilderStyleId

This operation is used to get a fully configured style by AutoBuilder Style ID. Note: AutoBuilderStyleId is provided for backward compatibility only.

INPUT: FullyConfiguredByAutoBuilderStyleIdRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
orderAvailability	OrderAvailability	See OrderAvailability	Specifies whether or not to include vehicles in fleet mode or retail mode.
autoBuilderStyleId	String	w2007k10m21t1	The AutoBuilder Style ID for the style.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.

OUTPUT: ToggleOptionResponse

Element Name	Type	Example	Description
toggleOptionResponse	ToggleOptionResponse	See ToggleOptionResponse	This provides an updated ConfigurationState.

17. materializeConfigurationState

This operation is used to populate a configuration object using the serialized value of a configuration. Use this method to populate all of the attributes of a ConfigurationState when only the serialized value is known.

INPUT: ConfigurationStateRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
serializedValue	String	See ConfigurationState	The serialized value of a ConfigurationState object.

OUTPUT: ConfigurationStateElement

Element Name	Type	Example	Description
configurationState	ConfigurationState	See ConfigurationState	The style information populated to the depth specified in the ReturnParameters object.

18. toggleOption

This operation is used to configure a style. Toggling an option will validate that the option selection is valid, force on any options required or included by the selected option, exclude options not allowed, and update all dynamic data including pricing, technical specifications, and color availability. When an option selection is not valid, toggleOption returns a response that can be displayed to the user to provide steps to resolve the conflict.

INPUT: ToggleOptionRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
configurationState	ConfigurationState	See ConfigurationState	The configuration state of the style.
chromeOptionCode	String	Z83	The option code that the user has selected.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.
returnDeltaConfiguration *	Boolean	TRUE	Returns only the options and attributes (such as tech specs, prices, etc.) that changed with the last option toggle.

* If you want to achieve faster performance, then set the returnDeltaConfiguration element to TRUE. See note on dynamic data (*DD) in the Configuration type section for what information is returned when this element is set to TRUE. To improve performance gains, no aggregates are returned. Depending upon how you are implementing your application, set it to FALSE to store all the data on a server and then it is updated, else set it to TRUE to display everything and then refresh it.

OUTPUT: ToggleOptionResponse

Element Name	Type	Example	Description
toggleOptionResponse	ToggleOptionResponse	See ToggleOptionResponse	This provides an updated ConfigurationState.

19. selectColor

This operation is used to add color selections to the configuration state. Color availability is affected by option selections; however, color selections do not affect option availability or pricing. Adding the color selections to the style's state means the configuration state can be restored with the user's color selections intact.

INPUT: SelectColorRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
configurationState	ConfigurationState	See ConfigurationState	The configuration state of the style to retrieve.
combinationColorId	String	DV-167551041 7J1943219436	This is the combined ID of the primary exterior color and the interior color.
secondaryColorId	String	S2-1882558260L21958927315X556112443	The ID of the secondary exterior color code selected.
auxiliaryChoiceColorId	String	1247100866217-172855086687-1987300828	The ID of the exterior trim and/or stripe color selected.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.

OUTPUT: ConfigurationElement

Element Name	Type	Example	Description
configuration	Configuration	See Configuration	AN updated Configuration containing the color selection changes.

20. autoEquipStyles

This operation is used to configure a set of styles to have similar equipment as your primary vehicle, providing a way to compare the vehicle feature by feature.

INPUT: AutoEquipRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
sourceConfiguration State	ConfigurationState	See ConfigurationState	The configuration state of the style that the target styles are to be configured similar to
targetConfiguration States	Array of ConfigurationState	See ConfigurationState	The configuration states for each style that is to be configured similar to the source style.

OUTPUT: ConfigurationStateArrayElement

Element Name	Type	Example	Description
configurationState	Array of ConfigurationState	See ConfigurationState	The updated configuration state for each of the target configuration states.

21. getMarketClasses

This operation is used to retrieve the market classes of the vehicles as specified by the EPA (Environmental Protection Agency). See Appendix C - Market Classes for a list of valid types.

INPUT: MarketClassesRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.

OUTPUT: MarketClassArrayElement

Element Name	Type	Example	Description
array	Array of MarketClass	See MarketClass	All available market classes.

22. getDataVersions

This operation is used to get the AutoBuilder data version information. This information may be helpful in debugging issues when talking to Chrome Client Support.

INPUT: DataVersionsRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.

OUTPUT: DataVersionArrayElement

Element Name	Type	Example	Description
dataVersion	Array of DataVersion	See DataVersion	All available data build numbers.

23. getCategoryDefinitions

This operation returns a list of all categories (generic descriptions). Categories are assigned to standard equipment and optional equipment. Use this lookup to retrieve descriptions of all possible categories that may be used to describe style equipment.

INPUT: CategoryDefinitionsRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.

OUTPUT: CategoryDefinitionsArrayElement

Element Name	Type	Example	Description
array	Array of CategoryDefinition	See CategoryDefinition	All available categories.

24. getTechnicalSpecificationDefinitions

This operation returns a list of all technical specification titles. Technical specification titles will vary depending on vehicle type, exterior, cargo options, and interior configuration. Use this lookup to retrieve descriptions of all possible technical specifications that may be returned within a style's technical specification data.

INPUT: TechnicalSpecificationDefinitionsRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.

OUTPUT: TechnicalSpecificationDefinitionArrayElement

Element Name	Type	Example	Description
array	Array of TechnicalSpecification Definition	See TechnicalSpecification Definition	All available technical specification definitions.

25. getOptionKinds

This operation returns a list of all option kinds. Use this lookup to retrieve descriptions of all option kinds that may be assigned to an option.

INPUT: OptionKindsRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.

OUTPUT: OptionKindArrayElement

Element Name	Type	Example	Description
optionKind	Array of OptionKind	See OptionKind	All option kinds that can appear on an option.

26. getAdvantageComparisonRuleSetNames

This operation is used to get the rule set names for advantage-based comparisons. Rule files offer flexibility in customized rule sets. These files are managed by Chrome, and are available to anyone with access to the system.

INPUT: AdvantageComparisonRuleSetNamesRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.

OUTPUT: StringArrayElement

Element Name	Type	Example	Description
s	Array of String	chromerules	All available rule set names.

27. compareAdvantages

This operation is used to compare advantages and disadvantages between the primary vehicle and any additional vehicles being compared. Vehicles cannot be compared across locales (US, CA).

INPUT: AdvantageBasedComparisonRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
ruleSetName	String	chromerules	This is the name of the rule set.
pivotConfiguration State	ConfigurationState	See ConfigurationState	Configuration state of the pivot (primary) vehicle.
comparison ConfigurationStates	Array of ConfigurationState	See ConfigurationState	Configuration states of the comparison vehicles.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.

OUTPUT: AdvantageBasedComparison

Element Name	Type	Example	Description
pivotConfiguration	**Configuration**	See Configuration	The Configuration of the pivot (primary) vehicle.
comparisons	**Array of AdvantageComparison**	See AdvantageComparison	The AdvantageComparisons of the comparison vehicles.

28. compareSideBySide

This operation is used to compare technical specifications and category items between the primary vehicle and any additional vehicles being compared. Vehicles cannot be compared across locales (US, CA).

INPUT: SideBySideComparisonRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
comparison ConfigurationStates	Array of ConfigurationState	See ConfigurationState	Configuration states of the comparison vehicles.
includeCategory Comparisons	Boolean	TRUE	Indicates whether category comparisons are included in the comparison. FALSE returns no categories for comparison, TRUE uses the configuration state array to get category IDs.
filteredCategoryIds	Array of int	1082, 1081	All filtered category IDs. If the value is null, all category IDs will be returned. See Appendix A – Categories.
includeTechSpec Comparisons	Boolean	TRUE	Indicates whether technical specifications are included in the comparison. FALSE returns no technical specifications for comparison, TRUE uses the configuration state array to get tech spec IDs.
filteredTechSpec TitleIds	Array of int	26, 27	All filtered technical specification IDs. If the value is null, all tech spec IDs will be returned.
returnParameters	ReturnParameters	See ReturnParameters	Used to indicate which attributes of the style to fill in the return set. Returning only what is needed improves performance.

OUTPUT: SideBySideComparison

Element Name	Type	Example	Description
comparison Configurations	**Array of Configuration**	See Configuration	The Configuration of the pivot (primary) vehicle.
comparisonGroups	**Array of SideBySideComparison Group**	See SideBySideComparison Group	The SideBySideComparisonGroups of the comparison vehicles.

29. searchStyles

This operation is used to search for styles with specific features and/or available equipment.

INPUT: SearchStylesRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.

Element Name	Type	Example	Description
orderAvailability	OrderAvailability	See OrderAvailability	Specifies whether or not to include vehicles in fleet mode or retail mode.
searchRequest	SearchServiceRequest	See SearchServiceRequest	The SearchServiceRequest is a collection of search criteria used for executing a search.

OUTPUT: StyleArrayElement

Element Name	Type	Example	Description
style	Array of Style	See Style	All styles matching the search criteria.

30. validateSearch

This operation is used to validate whether the search criteria being used is valid or not. If any of the search criteria are invalid, you will receive a SearchCriterionError with a description of the error.

INPUT: ValidateSearchRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
searchRequest	SearchServiceRequest	See SearchServiceRequest	The SearchServiceRequest is a collection of search criteria used for executing a search.

OUTPUT: SearchCriterionErrorArrayElement

Element Name	Type	Example	Description
array	Array of SearchCriterionError	See SearchCriterion Error	All applicable search criterion errors.

31. searchModels

This operation is used to search for models that contain styles with specific features and/or available equipment.

INPUT: SearchModelsRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.
orderAvailability	OrderAvailability	See OrderAvailability	Specifies whether or not to include vehicles in fleet mode or retail mode.
searchRequest	SearchServiceRequest	See SearchServiceRequest	The SearchServiceRequest is a collection of search criteria used for executing a search.

OUTPUT: ModelSearchResultArrayElement

Element Name	Type	Example	Description
result	Array of ModelSearchResult	See ModelSearchResult	All models meeting the search criteria.

32. getSearchCriterionDescriptors

This operation is used to show the properties of the search criteria. Each criteria has a token name, type, a min and max value (if the type is a MoneyRange, NumberRange, or TechnicalSpecificationRange), and unit (tech spec search criteria).

INPUT: SearchCriterionDescriptorRequest

Element Name	Type	Example	Description
accountInfo	AccountInfo	See AccountInfo	Account information for authentication.

OUTPUT: SearchCriterionDescriptorArrayElement

Element Name	Type	Example	Description
array	Array of SearchCriterionDescriptor	See SearchCriterior Descriptor	The search criterion descriptors.

Complex Types

In this section, the types are arranged in alphabetical order.

AccountInfo

Customer account information is required for authorization purposes.

Name	Type	Example	Description
accountNumber	String	123456	The account number issued by Chrome to identify the account to be authenticated.
accountSecret	String	password	The account password to protect the account holder.
locale	Locale	See Locale	The country and language requested.
sessionId	String	A98400sMFD8	Set to the ID of the session to link together activities of a single user for the purpose of click stream logging.

AdditionalImage

This type contains information about any additional images that may be available for a given vehicle style. Additional images may include interior and alternate exterior views, and may not be available for all vehicles.

Name	Type	Example	Description
description	String	*	Returns a description of the type of media being returned.
url	String	*	Returns the URL of the media.
mimeType	String	*	The mime type for the image. This can be used to determine the type of content being returned.

* Reserved for future use.

AdvantageBasedComparison

This is the return value of an advantage based comparison. The pivot configuration represents the style that all of the natural language descriptions are centered around. The comparison styles represent the styles to which the pivot configuration is being compared.

Name	Type	Example	Description
pivotConfiguration	Configuration	See Configuration	Style representing the primary vehicle.

Name	Type	Example	Description
comparisons	Array of AdvantageComparison	See Advantage Comparison	Contains the collection of natural language advantages and disadvantages for the vehicles being compared, such as pricing, images, etc.

AdvantageComparison

This is the return value of an advantage based comparison.

Name	Type	Example	Description
comparison Configuration	Configuration	See Configuration	This is the configuration being compared.
comparisonItems	Array of AdvantageComparison Item	See Advantage ComparisonItem	Contains the collection of natural language advantages and disadvantages for the vehicles being compared, such as pricing, images, etc.

AdvantageComparisonItem

Each AdvantageComparisonResultItem represents the result of a rule evaluation from a specified rule set. This is returned as part of the AdvantageComparisonResult.

Name	Type	Example	Description
comparisonResult Type	ComparisonResultType	See ComparisonResult Type	Defines whether the item compared is an advantage, disadvantage, or tie as compared with the other style.
comparisonValue	String	Continuously Variable Transmission not available, Automatic Transmission available and Manual Transmission standard	Contains the values being compared.
difference	String	2	Returns the difference between the values being evaluated. For example, if the rule evaluates MSRP, this would be the difference between the MSRP of the pivot style and the MSRP of the other style.
naturalLanguage Description	String	Automatic Transmission standard, versus Manual Transmission standard	Returns the natural language description specified by the comparison rule that generated this result.

Name	Type	Example	Description
pivotValue	String	Continuously Variable Transmission not available, Automatic Transmission standard and Manual Transmission not available	Returns the value of the rule being evaluated for the pivot style.
ruleDescription	String	transmissionType	Returns the name of the rule used to generate this result.
units	String		Returns the unit of measure of the item being compared. When this is a technical specification, this would indicate if the number references a weight, height, width, etc.

AndCriterion

This allows user to combine multiple values with importance of a specified search token. The same search token name must be used for each of the search criterion in the array. Only NumberRange and String criterion types can be used in an AndCriterion. This interface would be used when user seeks "AND" behavior of the assigned token values during a search. This is used when searching for a range of related criteria for BodyType, such as "2dr Car" and "Convertible".

Name	Type	Example	Description
name	SearchTokenName	See SearchTokenName	The name of the Search Criterion.
criteriaArray	Array of SearchCriterion	See SearchCriterion	The array of Search Criterion.

AuxiliaryColor

The AuxiliaryColor represents an exterior trim color (e.g., Wheel Flares) that comes with the current color combination. Not all vehicles carry auxiliary colors. The auxiliary color is not selectable.

Name	Type	Example	Description
title	String	Paint Stripe	Indicates the exterior portion of the vehicle that contains the color being described.
name	String	Red	The manufacturer's color name.
condition	String	((-ZY1)&(__A)&(__C)),((-ZY1)&(__A)&(!__C))	The condition that must be satisfied for the auxiliary color to be present.
valid	Boolean	TRUE	Indicates that the auxiliary color is valid with the current configuration.

AuxiliaryChoiceColor

The AuxiliaryChoiceColor represents an exterior trim color that the user must select from within a color combination. This is primarily used for stripes. Not all vehicles carry auxiliary choice colors. Only one auxiliary choice color can be selected at a given time.

Name	Type	Example	Description
manufacturersCode	String	87A	The manufacturer's color code for the auxiliary color. Note: Some of the codes carried may not be the true manufacturer's invoice code.
title	String	Stripe Color	Indicates the exterior portion of the vehicle that contains the color being described.
name	String	Pewter Gray	The manufacturer's color name.
genericColors	Array of GenericColor	See GenericColor	Returns all of the generic descriptions for this auxiliary choice color. Generic color descriptions standardize color descriptions across manufacturers.
colorId	String	1247100866217-172855086687-1987300828	This is the ID of the auxiliary color choice.
condition	String	((-ZY1)&(__A)&(__C)),((-ZY1)&(__A)&(!__C))	The condition that must be satisfied for the auxiliary choice color to be present.
valid	Boolean	TRUE	Indicates that the auxiliary choice color is valid with the current configuration.
selected	Boolean	TRUE	Indicates whether the current color choice is selected. Selected color choices are stored in the ChromeStyleKey.Returns TRUE if selected.

BodyType

The body type defines the general description of the vehicle body. For example, 2 DR car, sport utility, convertible. Each style can have one or more body type with exactly one primary body type description.

Name	Type	Example	Description
bodyTypeId	int	3	This is a unique identifier of the body type.
bodyTypeName	String	Regular Cab Pickup	Returns the primary body type.
primary	Boolean	TRUE	This indicates whether or not the body type is the primary body type.

Category

This type represents the generic equipment description for the standard equipment item or optional equipment item that it represents. Standard equipment and optional equipment may have zero or more categories. See Appendix A – Categories.

Name	Type	Example	Description
categoryFlag	CategoryFlag	See CategoryFlag	This defines the relationship of the category to the option or standard.
categoryId	int	1150	This is the identifier for the category.

CategoryDefinition

This type represents the generic description of a standard or optional equipment item. See Appendix A - Categories for a list of valid values.

Name	Type	Example	Description
categoryId	int	1002	This is the identifier for the category. Allows for the comparison of similar equipment on two different vehicles.
categoryName	String	Passenger Air Bag	This is the generic equipment description.

ColorCombination

This type represents combinations of exterior and interior colors available for the style. Within each combination, there may be one or more secondary colors and auxiliary colors. The availability of a secondary color, auxiliary choice color, or auxiliary color is driven by option selections. When the ReturnParameters are set to include invalid colors, all combinations possible for the style are returned even if those colors are not orderable as configured. When the ReturnParameters are set to not include invalid colors, only active combinations, secondary exterior colors, and auxiliary colors are returned.

Name	Type	Example	Description
primaryExteriorColor	PrimaryExteriorColor	See PrimaryExterior Color	This field is a Manufacturer-defined or Chrome-defined field. Note: Some of the codes carried may not be the true manufacturer's invoice code. Returns the primary exterior color within this combination within this style.
secondaryExterior Colors	Array of SecondaryExteriorColor	See SecondaryExterior Color	This field is a Manufacturer-defined or Chrome-defined field. A secondary exterior color may be a lower body color or a top color.
interiorColor	InteriorColor	See InteriorColor	Manufacturer's Interior Color Code if not blank. Returns the interior color for this combination within this style.

Name	Type	Example	Description
auxiliaryColors	Array of AuxiliaryColor	See AuxiliaryColor	All auxiliary colors where the user does not have a selection choice.
auxiliaryChoiceColors	Array of AuxiliaryChoiceColor	See AuxiliaryChoiceColor	All auxiliary colors where the user can choose the color.
colorId	String	DV-1675510417J19432 19436	This is the ID of the selected primary exterior color ID and the interior color ID.
condition	String	((-ZY1)&(__A)&(__C)), ((-ZY1)&(__A)&(!__C))	This is the condition that must be satisfied for the combination of interior and exterior color to be orderable. When the condition is met, the combination will be marked as valid. This is not intended for display. It can be used to programmatically determine options that affect configuration.
valid	Boolean	TRUE	Indicates whether the combination is allowed with the current configuration.
selected	Boolean	TRUE	Indicates if this color combination is currently selected.

ColorSelection

This type represents the selected colors for the current configuration of the style.

Name	Type	Example	Description
combinationColorId	String	DV-1675510417J19432 19436	This is a combination of the selected primary exterior color ID and the interior color ID.
secondaryColorId	String	S2-1882558260L21958 927315X556112443	This is the ID of the selected secondary exterior color.
auxiliaryChoiceColor Id	String	1247100866217-172855086687-1987300828	This is the ID of the selected auxiliary choice color.
validWithCurrent Configuration	Boolean	TRUE	Indicates whether or not the color selections are valid with the current configuration.

Configuration

This type contains all of the information regarding the style including options, standards, technical specifications, etc. The contents of the style are determined by the ReturnParameters passed in to the respective getStyleFullyConfigured, toggleOption, and selectColor operation.

Name	Type	Example	Description
style *DD	Style	See Style	This is the collection of style descriptions including the model year, division name, model name, body type, etc.
configuredOptions Msrp *DD	double	-945.00	This is the MSRP of the options as configured.
configuredOptions Invoice *DD	double	-784.35	This is the invoice price of the options as configured. Note the invoice price is only available for US vehicles through this service.
configuredTotalMsrp *DD	double	17,815.00	This is the total MSRP of the style as configured. This does not include the destination charge.
configuredTotal Invoice *DD	double	16,993.35	This is the total invoice price of the style as configured. This does not include the destination charge. Note the invoice price is only available for US vehicles through this service.
configuredPriceState	PriceState	See PriceState	This is an indicator as to whether the price is confirmed, unknown, or estimated.
standardEquipment Notes	Array of String	ALL STANDARDS ARE 2007 UNLESS OTHERWISE NOTED	Information regarding the standard equipment. The includeStandards flag needs to be set to true to get standardEquipmentNotes.
standardEquipment	Array of Standard	See Standard	This is the collection of standard equipment.
orderingNotes	Array of String	INVOICE PRICES TO FOLLOW	Information regarding the ordering of the vehicle. The includeOptions flag in Return Parameters needs to be set to true to return OrderingNotes.
options *DD	Array of Option	See Option	This is the collection of optional equipment including descriptions, pricing, and configuration state.
colorCombinations *DD	Array of ColorCombination	See ColorCombination	This is the collection of all available color combinations for the vehicle including exterior, interior, and trim colors.
configurationCheck ListItems	Array of ConfigurationCheckList Item	See Configuration CheckListItem	This is the check list of groups of options that need to be satisfied for the style configuration to be ordered.
consumerInformation	ConsumerInformation	See Consumer Information	This is the collection of consumer information text.
structuredConsumer Information	Array of StructuredConsumer Information	See Structured Consumer Information	Returns all structured consumer information.
technicalSpecifications *DD	Array of TechnicalSpecification	See Technical Specification	This is the collection of technical specifications based on the current configuration of the style.

Name	Type	Example	Description
editorialContent Sources	Array of EditorialContentSource	See EditorialContent Source	Returns all editorial content sources.
additionalImages	Array of AdditionalImage	See AdditionalImage	Returns all additional images.

*DD If the returnDeltaConfiguration in the toggleOption operation is set to TRUE, then these items are always returned. For options and colorCombinations, these items MAY be returned if their values changed as a result of the last option toggle. These are the items that are designated as dynamic fields.

ConfigurationCheckListItem

The ConfigurationCheckListItem represents a group of options where one option in the group must be selected for the group to be satisfied. The check list itself refers to the collection of ConfigurationCheckListItem objects returned from the Configuration. This list will vary per style. The typical items in a check list are engine, transmission, package codes.

Name	Type	Example	Description
itemName	String	Chassis Package	The name of the checklist item group. This describes the options.
chromeOptionCodes	Array of String	Z83	The option codes as a string. Using the description from the option, these options would be presented to the user for selection. The option would be selected using the toggleOption interface.
satisfied	Boolean	TRUE	Returns true if one of the options within the group is selected.

ConfigurationState

The ConfigurationState encapsulates the identifier for the style including settings for order availability, the enabling of special equipment, and all option and color selections.

Name	Type	Example	Description
dataVersion	dateTime	02/02/2007 15:02:54	The date and time the data was produced. Note: The format may vary depending upon your implementation.
styleId	int	285716	Unique identifier (primary key) assigned by Chrome. Style ID does not change once it is assigned to a vehicle.
fullyConfigured	Boolean	TRUE	Returns true if all items in the orderable vehicle checklist have been selected.
orderAvailability	OrderAvailability	See OrderAvailability	Specifies whether or not to include vehicles in fleet mode or retail mode.

Name	Type	Example	Description
specialEquipment Enabled	Boolean	FALSE	Indicates if special equipment is enabled. This setting affects displayed options, comparison results, and ordering rules (in configuration).
chromeOptionCode ToggleStream	Array of String	LU3, GU5	This is a list of option selections that the user made, in the order in which they were selected.
selectedChrome OptionCodes	Array of String	LU3, GU5	This is a list of all selected options.
selectedColor	ColorSelection	See ColorSelection	Indicates the state of the color selection.
serializedValue	String	<chromeStyleKey country="US" id="w2007k10m21t1" language="en" sessionId="64636a3:1139bdc9bcf:-7d99"><created>July 6, 2007 11:39:42 AM PDT</created><lastModified>July 6, 2007 11:39:42 AM PDT</lastModified><dataCreationTime>June 25, 2007 3:03:00 PM PDT</dataCreationTime><fleetMode>fleet</fleetMode><includeDioPrice>true</includeDioPrice></chromeStyleKey>	A string representation of the state.

ConsumerInformation

This type contains all of the consumer information available for the style.

Name	Type	Example	Description
crashTestRating	String	NHTSA CRASH TEST RESULTS: Frontal crash ratings: Driver: * * * * Passenger: * * * * * Side crash ratings: Driver: * * * * Rear Passenger: * * * * Results based on a 35 MPH frontal crash and 38.5 MPH side crash. Results are reported in a range of one to five stars, with five stars indicating the best crash protection for vehicles within the same weight class. This test used driver and passenger belts and airbags.	NHTSA crash test ratings for the style.

Name	Type	Example	Description
		NHTSA ROLLOVER RESISTANCE RATING: * * * * The Rollover Resistance Rating is an estimate of your risk of rolling over if you have a single vehicle crash. It does not predict the likelihood of that crash. The Rollover Resistance Rating utilizes a "fishhook" maneuver which is a series of abrupt turns at varying speeds to see how "top-heavy" a vehicle is. The more "top-heavy" the vehicle, the more likely it is to roll over. The lowest rated vehicles (1-star) are at least four times more likely to roll over than the highest rated vehicles (5-stars).	
rebate	String	One of the following incentives may apply to this vehicle. Regional incentives may vary. Cash Incentive: $1500 or financing Financing Incentive: 0.0% Expiration Date: 07/09/2007 Resource: GM.com 05/10/2007	The OEM's rebate information for the style with effective dates.
recall	String	NHTSA CAMPAIGN ID: 05V455000 Mfg's Report Date: 10/04/2005 Component: STEERING:HYDRAULIC POWER ASSIST:HOSE, PIPING, AND CONNECTIONS Potential Number Of Units Affected: 123592 Summary: CERTAIN PASSENGER VEHICLES MAY HAVE BEEN BUILT WITH A POWER STEERING HOSE THAT IS NOT TO SPECIFICATION. UNDER EXTREME STEERING MANEUVERS, SUCH AS TURNING THE STEERING WHEEL FULLY TO THE LEFT OR RIGHT WHILE BRAKING, THE HOSE MAY FRACTURE AND LEAK FLUID. Consequence: IF THIS WERE TO OCCUR, POWER STEERING ASSIST WOULD BE LOST AND INCREASED STEERING EFFORT WOULD BE REQUIRED. ON VEHICLES EQUIPPED WITH HYDRO-BOOST POWER BRAKES, IT COULD ALSO RESULT IN LOSS OF POWER BRAKE ASSIST AND INCREASED BRAKING EFFORT WOULD BE REQUIRED. IF THE POWER STEERING FLUID SPRAY WERE TO SPRAY ONTO HOT ENGINE PARTS,	Recall notices, if available.

Name	Type	Example	Description
		AN ENGINE COMPARTMENT FIRE COULD OCCUR. Remedy: DEALERS WILL INSPECT THE POWER STEERING HOSE(S) FOR TWO SUSPECT DATE CODES AND REPLACE THEM IF REQUIRED. THE RECALL WILL BEGIN ON OCTOBER 14, 2005. OWNERS SHOULD CONTACT CHEVROLET AT 1-800-630-2438, GMC AT 1-866-996-9463, BUICK AT 1-866-608-8080, CADILLAC AT 1-866-982-2339, ISUZU AT 1-800-255-6727, OR HUMMER AT 1-800-732-5493. Notes: GM RECALL NO. 05086. CUSTOMERS CAN ALSO CONTACT THE NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION'S VEHICLE SAFETY HOTLINE AT 1-888-327-4236 (TTY 1-800-424-9153), OR GO TO HTTP://WWW.SAFERCAR.GOV.	
warranty	String	Warranty = Basic: 3 Years/36,000 Miles Drivetrain: Gas Engine 5 Years/100,000 Miles Diesel Engine 5 Years/100,000 Miles Corrosion: 6 Years/100,000 Miles Roadside Assistance: 5 Years/100,000 Miles	The OEM's warranty information.

DataVersion

This is to the data version information that may be helpful to know when contacting Customer Support.

Name	Type	Example	Description
country	String	US	A valid ISO-3166 Country Code.
build	String	419	The build number of this version of the data.
date	dateTime	02/02/2007 15:02:54	The date and time the data was produced. Note: The format may vary depending upon your implementation.

Division

This type contains the division description information used in a vehicle selector.

Name	Type	Example	Description
divisionId	int	8	The ID of the division used as a parameter to the getModelsByDivision operation.
divisionName	String	Chevrolet	The division name used to display to the user.

EditorialContent

This type provides Chrome AutoBrief review information. See Appendix F – Editorial Content Types for a list of valid types.

Name	Type	Example	Description
description	String	MODEL_VALUE	Returns a description of the editorial content used to identify the type of content.
value	String	Refinement; interior appointments; fuel economy; tow rating; available luxury features	Returns the body of the editorial content. In this case, it the Chrome Reviews.
mimeType	String	text/plain	The mime type for editorial content.

EditorialContentSource

This type provides the name and associated content of the editorial content.

Name	Type	Example	Description
sourceName	String	Chrome Reviews	Returns the names of the edtiorial content sources provided for this style. The source name includes the provider of the data and the type of data. Use this name to retrieve an editorial content group.
editorialContents	Array of EditorialContent	See EditorialContent	Returns the editorial content in this collection from the specified source.

FilterRules

This operation is used as a parameter to filter or restrict the returned results.

Element Name	Type	Example	Description
orderAvailability	OrderAvailability	See OrderAvailability	Specifies whether or not to include vehicles in fleet mode or retail mode.

Element Name	Type	Example	Description
postalCode	String	97232	Specifies which postal code you want to restrict your results to. This is valid for the US only.
marketClassIds	Array of int	2	Specifies which market class IDs you want returned by the search.
vehicleTypes	Array of VehicleType	See VehicleType	Specifies which vehicle types you want returned by the search.
msrpRange	MoneyRange	See MoneyRange	Specifies the money range you want returned by the search.

GenericColor

A generic color is a standardized description of an exterior color used to describe colors across manufacturers.

Name	Type	Example	Description
colorName	String	Silver	Indicates if this generic color best describes the manufacturer's color when more than one generic color applies. For example, Champagne may be described as "White" or "Beige". If the vehicle color is closer to beige, beige would be the primary color.
primary	Boolean	TRUE	Returns true if this generic color best represents the manufacturer's color

InteriorColor

This type represents the interior color within the color combination.

Name	Type	Example	Description
manufacturersCode	String	88V	The manufacturer's color code for the interior color.
name	String	Dark Titanium	The manufacturer's color name for the interior color.
genericColors	Array of GenericColor	See GenericColor	The generic color descriptions for the manufacturer color.

Locale

Applications can use Locale to tailor information for the language and location of its users. Currently the supported locales include "en, US", "en, CA", and "fr, CA". Vehicles cannot be compared across locales (US, CA). Note: Locales are licensed separately.

Name	Type	Example	Description
country	String	US	A valid ISO-3166 Country Code.
language	String	en	A valid ISO-639-1 Language Code.

MarketClass

The market class defines the market class of the vehicle as specified by the EPA. A market classification indicates the type of vehicle, general size, and some indicators of body styles. Each Chrome style is linked to one market class. See Appendix C - Market Classes.

Name	Type	Example	Description
id	int	3	This is the unique identifier for the market class.
name	String	2WD Standard Pickup Trucks	Returns the market class description.

Model

This type contains the model description information used in a vehicle selector.

Name	Type	Example	Description
modelYear	integer	2007	The model year of the model.
divisionId	int	8	The division ID of the model.
divisionName	String	Chevrolet	The division name of the model.
subdivisionId	int	10	The subdivision ID of the model.
subdivisionName	String	Chevy Pickups	The subdivision name of the model.
modelId	int	16615	The year make model ID.
modelName	String	Silverado 1500	The name of the model for display to the user.
lastModifiedDate	dateTime	08/30/2007 08:12:14	The date the model information was last updated. Note: The format may vary depending upon your implementation.
effectiveDate	dateTime	11/22/2007 07:44:26	Effective date of the later of Price Sheet, Order Guide, or major bulletin revising either. Note: The format may vary depending upon your implementation.

ModelSearchResult

This type list is used to return all styles available for a specific model.

Element Name	Type	Example	Description
model	Model	See Model	The name of a model.
styles	Array of Style	See Style	All styles available for the specified model.

MoneyRange

This type is used to specify a price range.

Element Name	Type	Description
minimumPrice	double	This is the minimum price for the search range.
maximumPrice	double	This is the maximum price for the search range.

Option

This type represents the option including the description, pricing, and configuration state.

Name	Type	Example	Description
chromeOptionCode	String	__V	This is the Chrome option code that is unique within a style. This is the option code that is passed as a parameter to toggleOption.
oemOptionCode	String	BG9	Returns the manufacturer's option code for this option. This may be null if the equipment does not carry a manufacturer's code.
headerId	int	11160	This is the ID for the display header.
headerName	String	ENGINE	This is the logical group name for optional equipment provided for display purposes.
optionKindId	int	6	This is a classification used to identify an option. Its primary purpose is for identifying options that can satisfy groups within the complete vehicle checklist used for configuring vehicles.
descriptions	Array of OptionDescription	See OptionDescription	This is a description of the option. Returns the concatenation of the primary name, extended description, rule logic description, implied logic description and price note.
msrp	double	-$784.35	Returns the MSRP of the option.

Name	Type	Example	Description
invoice	double	-$945.00	Returns the invoice price of the option. Note the invoice price is only available for US vehicles through this service.
priceState	PriceState	See PriceState	This is an indicator as to whether the price is actual, unknown, or estimated.
priceVaries	Boolean	FALSE	Indicates that this option's price could vary based on the selection of other options.
priceReason	String	null	This is this the condition for which the price applies. May be null (default price) or a logical expression consisting of option codes and Boolean expressions.
categories	Array of Category	See Category	Returns all available categories.
specialEquipment	Boolean	FALSE	Indicates that the option is a special equipment option.
selectionState	OptionSelectionState	See OptionSelection State	Indicates the state of the option based on the current option selections.
uniqueTypeFilter	String	Engine	This can be used to group mutually exclusive options into radio groups.

OptionDescription

This type represents the description of the option.

Name	Type	Example	Description
description	String	TOWING PACKAGE	This is the description of the option.
type	OptionDescriptionType	See OptionDescription Type.	Describes if the description is the primary option name, or the extended option description.

OptionKind

This type represents the description of the option kind. The option kind is an attribute of an option. It is primarily used to identify items within the same configuration check list but can be used to identify classifications of options. See Appendix E - OptionKinds for a list of valid values.

Name	Type	Example	Description
optionKindId	int	6	This is a classification used to identify an option. Its primary purpose is for identifying options that can satisfy groups within the complete vehicle checklist used for configuring vehicles. For some custom applications it can be used to identifying an option as a package.
optionKindName	String	Engine	This is the description of the option kind.

OrCriterion

The OrCriterion allows user to add multiple criteria where OR behavior requires among criterions (i.e. this will be used when user seeks "OR" behavior for a given set of tokens during a search).

Name	Type	Example	Description
importance	SearchImportanceType	See SearchImportance Type	This is the Search Importance Type.
criteriaArray	Array of SearchCriterion	See SearchCriterion	This is the search criteria.

PrimaryExteriorColor

This type represents the primary exterior color within a color combination.

Name	Type	Example	Description
manufacturersCode	String	16U	The manufacturer's color code.
rgbHexCode	String	929292	RGB Hex representation of the primary exterior color.
swatchUrl	String	http://media.hg.carbook.com/autoBuilderData/colorSwatches/acdip.gif	The URL for the exterior color swatch for the primary exterior color.
name	String	Graystone Metallic	The manufacturer's color name.
genericColors	Array of GenericColor	See GenericColor	The generic color description for the manufacturer color.

ReturnParameters

This type represents the request parameter that determines how populated the returned style should be. Choosing to return only data that you plan on using will increase performance.

Name	Type	Example	Description
includeStandards	Boolean	TRUE	Set this flag to true to include standard Equipment.

Name	Type	Example	Description
includeOptions	Boolean	TRUE	Set this flag to true to include options.
includeOption Descriptions	Boolean	TRUE	Set this flag to true to include option descriptions.
includeSpecial EquipmentOptions *	Boolean	TRUE	Set this flag to true to include special equipment options.
includeColors	Boolean	TRUE	Set this flag to true to include colors.
includeInvalidColors	Boolean	TRUE	Set this flag to true to include invalid colors.
includeEditorial Content	Boolean	TRUE	Set this flag to true to include editorial content.
includeConsumerInfo **	Boolean	FALSE	Set this flag to true to include consumer information as a single string.
includeStructured ConsumerInfo **	Boolean	See StructuredConsumerInformation and StructuredConsumerInformationItem	Set this flag to true to include structured consumer information items as separate elements, for example, if you wanted to use a rebate amount in a financing calculation.
includeConfiguration Checklist	Boolean	TRUE	Set this flag to true to include the configuration checklist.
includeAdditional Images	Boolean	TRUE	Set this flag to true to include additional images.
includeTechSpecs	Boolean	TRUE	Set this flag to true to include technical specifications.
filteredTechSpec TitleIds	Array of int	2000, 1101, 1100, 1099, 1098, 1097, 1096	Set this flag to true to include filtered tech spec title IDs. For example, return fuel economy only, instead of hundreds of items.

*If includeSpecialEquipmentOptions is set to TRUE, the Configuration State is superseded.

**Normally, you would set only one of the two Consumer Information elements to true, to include either one or the other.

SearchCriterion

Use the search criterion in an array with AndCriterion and/or OrCriterion to add filters to your search request. See Searching for Vehicles for detailed information.

Name	Type	Example	Description
name	SearchTokenName	See SearchTokenName	The name of the search token on which you wish to search.
importance	SearchImportanceType	See SearchImportanceType	The importance of the item in the search.

Name	Type	Example	Description
type	SearchCriterionType	See SearchCriterion Type	This is the type of search criterion.
value	String		The value can be Boolean, a NumberRange, MoneyRange, a String, or a TechnicalSspecificationRange, depending on the SearchTokenName or SearchCriterionType.
min	String	86.6	This is the minimum value for a NumberRange, MoneyRange, or TechnicalSpecificationRange.
max	String	308.0	This is the maximum value for a NumberRange, MoneyRange, or TechnicalSpecificationRange.

SearchCriterionDescriptor

This is where you specify what search criterion is included in the search. The name is limited to the list of SearchTokenName(s).

Name	Type	Example	Description
name	SearchTokenName	See SearchTokenName	This is the name of the search token you wish to include in the search.
type	SearchCriterionType	See SearchCriterion Type	This is the type of search criterion you wish to include in the search.
min	String	86.6	This is the minimum value for a range that comes from the data.
max	String	308.00	This is the maximum value for a range that comes from the data.
unit	SearchUnitType	See SearchUnitType	A text abbreviation of the unit of measure. Identifies the unit of measure for the specification information. For example, "lbs". This method returns the integer value of the type. There are three supported types, NA (0), SAE (1) and METRIC (2). .
values	SearchValue	See SearchValue	The parameters specified in the SearchValue.

SearchCriterionError

This type returns an error message to the user with an array of errors, such as out of range, etc.

Name	Type	Example	Description
name	SearchTokenName	See SearchTokenName	This is the name of the search token included in the search.

Name	Type	Example	Description
type	SearchErrorType	See SearchErrorType	This is the error type.
error	String	Invalid type encountered. Expected: Technical SpecificationValue, Found: NumberRange	This is the description of the error.

SearchValue

This type contains the search item ID and value for a given Search Token Name, as returned by the SearchCriterionDescriptor.

Name	Type	Example	Description
id	String	1001	The ID of the search item. In this example, this is the Category ID that corresponds to Driver Air Bag.
value	String	Driver Air Bag	The name of the search item. In this example, the value corresponds to the Category Name.

SearchServiceRequest

Name	Type	Example	Description
criteriaArray	Array of SearchCriterion	See SearchCriterion	This is the array of search criterion included in the search.
orCriteriaArray	Array of OrCriterion	See OrCriterion	This is the array of OrCriterion included in the search.
andCriteriaArray	Array of AndCriterion	See AndCriterion	This is the array of AndCriterion included in the search.
filterTBD	Boolean	TRUE	In some cases there may be technical information on a vehicle that is currently unknown (i.e., To be determined or TBD). For example, if you were searching for "fuelEconomyCity" with a range of 20 to 30 and you had filterTBD set to true, you would only get vehicles for which the fuel economy is available and it fell within the range. If you had it set to false, you would get those as well as all vehicles that did not have any fuel economy defined.
filterByPostalCode	Boolean	TRUE	This method allows user to set the flag whether to include the postal code during search or not as a criteria.
postalCode	String	97232	This is the postal code to filter the results to.

Name	Type	Example	Description
maxNumResults	int	100	This is the maximum number of results returned. NOTE: The default is set to 100, in order to improve response time for large queries.

SecondaryExteriorColor

This type represents the secondary exterior color available for the style. Where the secondary color is included for a combination, that color generally represents a lower body color or top color. Check the valid attribute to determine if the secondary exterior color is available with the current configuration. Only one secondary exterior color can be selected. Not all vehicles carry secondary exterior colors.

Name	Type	Example	Description
manufacturersCode	String	HG	The manufacturer's color code.
rgbHexCode	String	BBB49A	RGB Hex representation of the secondary exterior color.
swatchUrl	String	http://media.hq.carbook.com/autoBuilderData/colorSwatches/d9slf.gif	The URL for the exterior color swatch for the secondary exterior color. This URL is an image of the same color that is represented as the RGB Hex code.
name	String	Smokestone Metallic	The manufacturer's color name.
genericColors	Array of GenericColor	See GenericColor	The generic color descriptions for the manufacturer color.
colorId	String	DV-1675510417J1943219436HG-1033234775	This is the ID for the secondary exterior color. This is provided to enable selection of the exterior color.
condition	String	((954)&(!54L))	This is the condition that must be satisfied for the secondary exterior color to be available. This is included for reference only.
valid	Boolean	TRUE	Indicates that this secondary exterior color is valid with the current configuration.
selected	Boolean	FALSE	Indicates that the secondary exterior color is selected in the current configuration.

SideBySideComparison

All styles returned by the side-by-side comparison are clustered into logical groupings. These groups can be used for display and must be used to retrieve the comparison result.

Name	Type	Example	Description
comparison Configurations	Array of Configuration	See Configuration	All configurations being compared.

comparisonGroups	Array of SideBySideComparison Group	See SideBySide ComparisonGroup	All side by side comparison groups being compared.

SideBySideComparisonGroup

All categories and technical specifications returned by the side-by-side comparison are clustered into logical groupings. These group names can be used for display and must be used to retrieve the comparison result.

Name	Type	Example	Description
groupName	String	Accessories	The name of the logical grouping.
comparisonItems	Array of SideBySideComparison GroupItem	See SideBySide ComparisonGroup Item	The list of items being compared.

SideBySideComparisonGroupItem

This type provides access to the comparison result values for each category and technical specification in the side by side comparison. Comparison information will be returned in logical groups that can be displayed to the user such as "Accessories".

Name	Type	Example	Description
comparisonFeatureId	int	1176	Returns the ID of the item being compared. If this is a category, the ID is the category ID. If this is a technical specification, the ID is the technical specification title ID.
comparisonFeature Type	ComparisonFeatureType	See ComparisonFeature Type	Indicates whether this is a technical specification or a category.
comparisonValues	Array of String	Not Available, Not Available, Available	If the value is a category, one of the following values will be returned: ▪ Not Available - the category is not standard or optional. ▪ Available - the category is available as an option and is currently not selected. ▪ Selected - the category is optional and selected. ▪ Standard - the category is part of the standard equipment for the vehicle. If the value is a technical specification, the value will be the technical specification value of the style as configured. Note: The base style may return different values than a fully configured style
featureDescription	String	Passenger Vanity Mirror	This is a description of the feature being compared. See Appendix A - Categories.

Standard

This type represents a standard equipment items on a style.

Name	Type	Example	Description
headerId	int	1343	This is the ID for the display header.
headerName	String	SAFETY	This is the logical group name for standard equipment provided for display purposes. This is the collection of generic equipment descriptions for the standard.
description	String	Daytime Running Lamps, with automatic exterior lamp control	This is the complete description of the standard including equipment description and header.
categories	Array of Category	See Category	Returns all available categories.
upgrade	Boolean	FALSE	Indicates if the item was an upgrade from the base style.

StructuredConsumerInformation

This type represents the types (rebate, warranty, crash test results, and recalls) and the associated values of Chrome's structured consumer information. See Appendix G – Structured Consumer Information Types for a list of valid values.

Name	Type	Example	Description
typeName	String	Warranty	This is the consumer information type.
items	Array of StructuredConsumer InformationItem	See Structured Consumer InformationItem	This is the array of items associated with the consumer information type.

StructuredConsumerInformationItem

This type represents the name, value, and condition note of the structured consumer information item. See Appendix G – Structured Consumer Information Types for a list of valid values.

Name	Type	Example	Description
name	String	Drivetrain Years	The name of the normalized consumer information label.
conditionNote	String	Diesel Engine	The specific conditions to which the label's value applies. Condition notes only apply to warranty information.
value	String	3	The value of the consumer information name.

Style

The Style contains all IDs, description and configuration information for a style, including model year, division name, etc. This type may be used in a vehicle selector.

Name	Type	Example	Description
modelYear	integer	2007	The model year of the style.
divisionId	int	8	Unique ID assigned to a division.
divisionName	String	Chevrolet	The name of the division.
subdivisionId	int	4661	Unique identifier assigned to a vehicle subdivision.
subdivisionName	String	Chevy Pickups	Within a division, this is the name of the group of vehicles by type.
modelId	int	16615	The year make model ID of the model.
modelName	String	Silverado 1500	The name of the model for display to the user.
styleId	int	285716	Unique identifier (primary key) assigned by Chrome. Style ID does not change once it is assigned to a vehicle. This is the preferred identifier to use in Chrome Construct.
autoBuilderStyleId	String	w2007k10m21t1	The AutoBuilder Style ID of the style. Used for backward compatibility.
styleName	String	2WD Reg Cab 119.0" Work Truck	The name of the style for display to the user. Returns the full style name. The full style name includes the trim name and other distinguishing descriptions such as the body type, wheelbase, drivetrain, and/or number of passenger doors.
styleNameWithoutTrim	String	2WD Reg Cab 119.0"	Returns the style name. The style name is the full style name without the trim.
trimName	String	Work Truck	Returns the trim name. This is the OEM's branded description of the vehicles within a model group.
manufacturerModelCode	String	CC10703	The manufacturer's code for the model.
baseMsrp	double	17860.00	Returns the base MSRP (Manufacturer's Suggested Retail Price) for this style.
baseInvoice	double	16877.70	Returns the base invoice price for this style.
destination	double	900.00	Returns the destination charge.

Name	Type	Example	Description
trueBasePrice	Boolean	FALSE	TRUE if vehicle can be purchased for Base Price + Destination Charge.
priceState	PriceState	See PriceState	Indicates whether the price is the actual price or unknown.
bodyTypes	Array of BodyType	See BodyType	Returns all body types that apply to this style.
passengerDoors	int	2	Only applies to number of full-size independently operating passenger doors.
marketClassId	int	3	Returns the market class ID of the vehicle. The market class is the EPA classification of the vehicle.
marketClassName	String	2WD Standard Pickup Trucks	Returns the market class description. The market class is the EPA classification of the vehicle.
stockPhotoUrl	String	http://media.hq.carbook.com/autoBuilderData/stockPhotos/9493.jpg	This is the URL to a jpg within the dataset. This is typically the Chrome stock photo of the vehicle's exterior. Images are available for the last 10 years forward for US vehicles, and 2000 and forward for Canadian vehicles.
consumerFriendly ModelName	String	Silverado 1500	This is a model name that more closely represents how a vehicle is marketed to customers. It can be used in lieu of the Model Name listed in the Models table.
consumerFriendly StyleName	String	2WD Reg Cab 119.0" Work Truck	This is a consumer friendly style name can be used as an alternate to the Chrome Style Name.
consumerFriendly Drivetrain	String	Rear Wheel Drive	This is the standard drivetrain for the vehicle, used for grouping like styles.
consumerFriendly BodyType	String	Regular Cab Pickup - Short Bed	This is a simplified description of the body type used for grouping like styles.
configurationState	ConfigurationState	See ConfigurationState	The configuration state of the style to retrieve.

Note: The Configuration State attributes may be null when the ReturnParameters passed in to the operation suppress the respective item.

Subdivision

This type contains the subdivision description information used in a vehicle selector.

Name	Type	Example	Description
subdivisionId	int	4661	The ID of the subdivision used as a parameter to the getModels operation.
subdivisionName	String	Chevy Pickups	The subdivision name used to display to the user.

TechnicalSpecification

This type represents the value of a technical specification title for a style.

Name	Type	Example	Description
groupId	int	6	This is the ID for the display header.
groupName	String	Powertrain	This is the logical group name for technical specifications provided for display purposes.
headerId	int	14	This is the ID for the display header.
headerName	String	Mileage	This is the name of the display header.
titleId	int	27	This is the ID for mapping to the title name.
titleName	String	EPA Fuel Economy Est – Hwy	This is the Tech Specification Title.
value	String	21	This is the value for the technical specification for the style.
measurementUnit	String	MPG	A text abbreviation of the unit of measure. Identifies the unit of measure for the specification information. See SearchUnitType.

TechnicalSpecificationDefinition

The TechnicalSpecificationDefinition provides access to the performance data and technical specifications for a specific style. The technical specification data is dynamic. The technical specification values may change as configuration changes. As such, when configuring a vehicle, it is important to retrieve the technical specifications if the option selections have changed since the last retrieve.

Name	Type	Example	Description
titleId	int	10	This is the unique identifier for the title.
name	String	Base Curb Weight	This is the description of the technical specification title.
unitOfMeasure	String	lbs	This is the unit that the technical specification is returned as. See SearchUnitType.

ToggleOptionResponse

When toggling an option, this response notifies you if there is an option conflict and updates the configuration state.

Name	Type	Example	Description
status	ToggleOptionResponse Status	See ToggleOption ResponseStatus	This is the result of the toggle. This indicates whether or not the toggle was successful or not. If the toggle is not successful, the status will indicate the resolution path.
originatingChrome OptionCode	String	PCY	This is the option that caused the conflict. This could differ from the option that was toggled if multiple logic rules are triggered.
originatingOptionAn Addition	Boolean	TRUE	Indicates whether the originating option selection was to add or delete.
additionAffected ChromeOptionCodes	Array of String	LY5	These are the options that were added as a result of the originating option toggle.
deletionAffected ChromeOptionCodes	Array of String	UE0	These are the options that were deleted as a result of the originating option toggle.
conflictResolving ChromeOptionCodes	Array of String	LY2, LY5, LMG	These are the options to choose from to resolve the conflict.
requiresToggleTo Resolve	Boolean	TRUE	Indicates if this conflict must be resolved by selecting one of the options specified by the conflict resolving option codes.
configuration	Configuration	See Configuration	This is the style after the option has been toggled. This contains the updated pricing and option states. Use the ConfigurationState from this style to initiate subsequent toggleOption and selectColor requests.

Simple Types - Enumerations

CategoryFlag

The CategoryFlag describes the relationship of the category to the equipment that it describes.

Name	Description
Deleted*	Used primarily for options, this indicates that the category is no longer applicable when set. Indicates that the equipment described by this category is removed when the option that carries this flag is selected. If a vehicle comes standard with cloth seats, a leather seat trim option will have a category of cloth with a "DELETE" flag. In this example, this indicates that the selection of leather overrides standard cloth seats.

Name	Description
Package	Used to indicate the equipment represented by the category is included in the option as part of a package. This means that there is other equipment included by the option as well. Indicates that the category description applies to one of the items described in a standard or option. For example, a sport package may include leather seats.
Clean	Indicates that the category is the item described by the standard or option.

*In most circumstances where a list of generic descriptions is required, categories with the deleted flag should be ignored. This is useful to indicate when the selection of an option causes a generic description to be overridden. For example, if cloth seats are standard and leather seats are optional, there would be a category with a delete flag on the leather seat option indicating that cloth seats are deleted.

ComparisonFeatureType

This is the result of a side by side comparison.

Name	Description
Category	Indicates that a category is being compared between the vehicles.
TechnicalSpecification	Indicates that a Technical Specification is being compared between the vehicles.

ComparisonResultType

This is the result of an advantage based comparison as specified by the rule set.

Name	Description
Advantage	Indicates that the pivot style is better than the other style based on the advantage based rule definition.
Disadvantage	Indicates that the pivot style is worse than the other style based on the advantage based rule definition.
Tie	Deprecated. Indicates that the pivot style is equivalent to the other style based on the advantage based rule definition.
CannotCalculate	Deprecated. Indicates that one of the vehicles had insufficient data to determine the comparison. This will occur most often with technical specifications and warranties when the information is TBD during initial release of a vehicle.

OptionDescriptionType

Options carry more than one type of description. The Option Description Type indicates each type.

Name	Description
PrimaryName	This is the short description of the option.
Extended	This is a description of the contents of the option. In the case of an option package, this description describes the package contents.
EnforcedLogic	This is a description of the rules that will be enforced upon selection of the option.
UnenforcedLogic	This is a description of minor equipment changes that are not represented by options and are not enforced.
PriceNote	These are notes regarding the option's price.

OptionSelectionState

The OptionSelectionState defines the state of an option after toggleOption has been called. Toggling an option may affect the state of 0 or more other options. When toggling an option, use the state of each option to indicate the affect of the toggle to the user.

Name	Description
Unselected	Indicates that the option is not selected.
Selected*	Indicates that the option is selected explicitly by the user.
Included*	Indicates that this option is included by another option. For example, the option may be included by a package. This supersedes the selected state.
Required*	Indicates that this option is required by another selection. This supersedes the selected state.
Excluded	Indicates that another option has excluded this option. This can still be selected by the user in a UI, but they will need to de-select the option that caused the exclusion. This supersedes the selected state.
Upgraded	Indicates that this option has been replaced by another option that is considered an upgrade. This supersedes the selected state.

*When an option is selected, included, or required that equipment will be on the vehicle.

Note: If the user selects an option and a subsequent option selection alters the state to included, required, excluded or upgraded, the option state will be updated respectively. If the event that the option that triggered, the state change is de-selected, the option will revert to the selected state to preserve the user's choice.

OrderAvailability

OrderAvailability indicates whether the style should be configured as a fleet vehicle or a retail vehicle. This setting may affect equipment availability. Note that while most styles are available in fleet or retail mode, some styles are carried as fleet only or retail only. Requesting a fleet only vehicle in retail mode will create an exception.

Name	Description
Fleet	Specify fleet when retail only styles and options should be hidden.
Retail	Specify retail when fleet only styles and options should be hidden.

PriceState

This is an indicator as to whether the price is the actual price, an estimated price or unknown. Early in a model year, a style may be available before initial pricing is released. Use this price state to determine how to appropriately display the state of the pricing information to the user. In consumer facing applications, it is customary to display the price as N/A. In a professional facing application, the price may be displayed with a note indicating that price is not the actual price.

Name	Description
Actual	Indicates the pricing is the actual price of the style or option.
Estimated	Indicates that a price is available but that price is estimated. Estimates may come from prior model year or from another style within the model.
Unknown	Indicates that no pricing is available for the style or option.

SearchCriterionType

This type returns the search criterion type information. Searches should always use the ID value for the search criterion.

Name	Description
String	This criterion requires a string input. For example: ▪ SearchTokenName = division ▪ SearchImportanceType = MustHave ▪ value – “Honda”
Boolean	This criterion must contain the string TRUE or FALSE. Other examples of Boolean search criterion types include, for example: ▪ SearchTokenName = hasTargaTop ▪ returnDeltaConfiguration = true ▪ SearchImportance = MustHave When building a criterion, set the name of the Criterion to this token and the value to Boolean.TRUE. Note: When constructing a MUST_NOT_HAVE search set the Boolean to TRUE and the Importance to MUST_NOT_HAVE.
NumberRange	This criterion represents a number range (min / max) for use in building search criteria for numeric values such as model year. For example: ▪ SearchTokenName = year ▪ SearchImportance = MustHave ▪ min = “2006” ▪ max = “2008”
MoneyRange	This criterion represents a value range, such as MSRP or invoice price. For example: ▪ SearchTokenName = msrp ▪ SearchImportance = MustHave ▪ Min = “20000.0” ▪ max = “30000.0”
TechnicalSpecificationRange	This criterion contains a minimum and maximum technical specification value range. For example: ▪ SearchTokenName = GroundClearance(Front) ▪ SearchImportance = MustHave ▪ min = “15.0” ▪ max = “20.0”

SearchErrorType

These are the error messages that you may receive when conducting a search.

Enumeration Value	Description
InvalidToken	The SearchTokenName supplied is invalid.
InvalidType	The Type supplied is invalid.
InvalidValue	The range of values entered is invalid.
OutOfRange	The value entered is beyond the maximum or below the minimum range.

SearchImportanceType

This type is used to specify 'must have' or 'must not have' when searching for equipment or options.

Enumeration Value	Description
MustHave	When specified as the Importance, only styles with the Criterion will be returned.
MustNotHave	When specified as the Importance, styles with the Criterion will not be returned.

SearchTokenName

SearchTokens provides the base set of searchable criteria for building a new vehicle search request. These tokens range from equipment, technical specification items, and configuration elements. Each constant in SearchTokens is a searchable token that is used in building a Criterion. Each Criterion is arranged by SearchCriterionType, which may be a NumberRange, String, MoneyRange, Boolean, or TechnicalSpecificationRange.

Enumeration Value	Description
Boolean Search Criterion Types	
hasAirBagDriver	Use this token to search for vehicles with driver-side airbags.
hasAirBagPassenger	Use this token to search for vehicles with passenger airbags.
hasAirbagPassengerShutoff	Use this token to search for vehicles with a passenger airbag shutoff switch.
hasAirBagRear	Use this token to search for vehicles with rear airbags.
hasAirConditioningFront	Use this token to search for vehicles with front air conditioning.
hasAirConditioningRear	Use this token to search for vehicles with rear air conditioning.
hasAlarmSystem	Use this token to search for vehicles with an alarm.
hasAllWheelDrive	Use this token to search for vehicles with all wheel drive.
hasAmFmStereo	Use this token to search for vehicles with an AM/FM radio.
hasAutoOffHeadLight	Use this token to search for vehicles with Auto Off Headlights.
hasCassette	Use this token to search for vehicles with a cassette player.
hasCdChanger	Use this token to search for vehicles with a CD changer.
hasCdPlayer	Use this token to search for vehicles with a CD player.
hasChildLocks	Use this token to search for vehicles with child locks.
hasClimateControl	Use this token to search for vehicles with climate control.
hasContinuouslyVariableTrans	Use this token to search for vehicles with continuously variable transmission.
hasConvertibleHardtop	Use this token to search for vehicles with a convertible hardtop.
hasCruiseControl	Use this token to search for vehicles with cruise control.
hasDualAirConditioningControls	Use this token to search for vehicles with dual air conditioning controls.
hasEntertrainmentSystem	Use this token to search for vehicles with an entertainment system.
hasFixedSunRoof	Use this token to search for vehicles with fixed sun roof.
hasFourWheelDrive	Use this token to search for vehicles with four wheel drive.
hasFrontWheelDrive	Use this token to search for vehicles with front wheel drive.
hasHeatedDriverSeat	Use this token to search for vehicles with heated driver seat.
hasKeylessEntry	Use this token to search for vehicles with keyless entry.
hasManualSlidingSunRoof	Use this token to search for vehicles with manual sliding sun roof.
hasMoonRoof	Use this token to search for vehicles with a moon roof.

Enumeration Value	Description
hasNavigationSystem	Use this token to search for vehicles with a navigation system.
hasPanoramicRoof	Use this token to search for vehicles with panoramic roof.
hasPopupSunRoof	Use this token to search for vehicles with a popup sun roof.
hasPowerLocks	Use this token to search for vehicles with power locks
hasPowerSlidingSunRoof	Use this token to search for vehicles with power sliding sun roof.
hasPowerSteering	Use this token to search for vehicles with power steering.
hasPowerWindows	Use this token to search for vehicles with power windows.
hasPremiumSound	Use this token to search for vehicles with a premium sound system.
hasRearWheelDrive	Use this token to search for vehicles with rear wheel drive.
hasRearWindowDefogger	Use this token to search for vehicles with a rear window defogger.
hasSatelliteRadio	Use this token to search for vehicles with satellite radio.
hasSlidingSunRoof	Use this token to search for vehicles with a sliding sun roof.
hasSuperCharger	Use this token to search for vehicles with a super charger.
hasTargaTop	Use this token to search for vehicles with a targa top.
hasTiltSteeringWheel	Use this token to search for vehicles with a tilt steering wheel.
hasTractionControl	Use this token to search for vehicles with traction control.
hasTTop	Use this token to search for vehicles with a T-Top.
hasTurboCharger	Use this token to search for vehicles with a turbo charger.
NumberRange Search Criterion Type	
crashTestRating	This is the value of the NHTSA crash test rating – a min/max range of 1 to 5 stars (*).
year	Use this token to search for vehicles by model year.
MoneyRange Search Criterion Type	
destinationCharge	Use this token to search for vehicles by destination charge.
fleetIncentiveAmount	Use this token to search for vehicles by fleet incentive amount range.
invoicePrice	Use this token to search for vehicles by invoice price.
msrp	Use this token to search for vehicles by MSRP.
String Search Criterion Type Types marked with an asterisk (*) have a dynamic list of possible values with their searchable IDs available via getSearchCriterionDescriptors.	
airbagSideType *	Use this token to search for vehicles by side airbag type.

ID	Side Airbag Type
bodyFront	Side Airbag - Front Body
headOnlyFront	Side Airbag - Front Head Only
bodyRear	Side Airbag - Rear Body
headOnlyRear	Side Airbag- Rear Head Only

bodyType *	Use this token to search for vehicles by body type. See Appendix B – Body Types for a list of valid values.

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td>brakeType *</td><td>Use this token to search for vehicles by brake type.

<table><tr><th>ID</th><th>Brake Type</th></tr><tr><td>ABS 4-WHEEL</td><td>ABS 4-Wheel</td></tr><tr><td>ABS REAR</td><td>ABS Rear</td></tr><tr><td>DISC 4-WHEEL</td><td>Disc 4-Wheel</td></tr><tr><td>DRUM 4-WHEEL</td><td>Drum 4-Wheel</td></tr><tr><td>FRONT DISC/ REAR DRUM</td><td>Front Disc/ Rear Drum</td></tr></table></td></tr>
<tr><td>cargoClassification *</td><td>Use this token to search for vehicles by cargo classification type.

<table><tr><th>ID</th><th>Cargo Classification Type</th></tr><tr><td>1</td><td>Pickup</td></tr><tr><td>2</td><td>Passenger Van, Utility</td></tr><tr><td>3</td><td>Cargo Van</td></tr><tr><td>4</td><td>Step-Van, Comm. Cargo Van</td></tr><tr><td>5</td><td>Extended Chassis-Cab</td></tr><tr><td>6</td><td>Coupe, Sedan, Convertable</td></tr><tr><td>7</td><td>Wagon, Tracker, Jeep</td></tr><tr><td>8</td><td>Hatchback</td></tr><tr><td>9</td><td>Cutaway & Stripped Chassis</td></tr></table></td></tr>
<tr><td>categoryId</td><td>Use this token to search for vehicles by category ID. See Appendix A – Categories (Generic Equipment) for a list of valid values. A list of categories and their IDs is also available using the getCategoryDefinitions method.</td></tr>
<tr><td>differential *</td><td>Use this token to search for vehicles by differential type.

<table><tr><th>ID</th><th>Differential Type</th></tr><tr><td>LOCKING FRONT</td><td>Locking Front</td></tr><tr><td>LOCKING REAR</td><td>Locking Rear</td></tr></table></td></tr>
<tr><td>divisionId</td><td>Use this token to search for vehicles by Division ID. As an example, to search styles for division Kia, pass division ID 22 as token value. A list of divisions and their IDs is available using the getDivisions method.</td></tr>
</table>

Enumeration Value	Description
engineCylinders *	Use this token to search for vehicles by number of engine cylinders.

ID	Engine Cylinder Type
3	3
4	4
5	5
FLAT	Flat 6
STRAIGHT	Straight 6
V6	V6
8	8
10	10
12	12

engineType *

Use this token to search for vehicles by engine type.

ID	Engine Type
BI-FUEL/ELECTRIC	Bi-Fuel/Electric
DIESEL	Diesel
ELECTRIC	Fuel System-Electric
PROPANE	Fuel System-Propane
GASOLINE	Gasoline
NATURAL_GAS	Natural Gas

exteriorClassification *

Use this token to search for vehicles by exterior type or classification.

ID	Exterior Classification Type
1	Pickups, Chassis-Cabs
2	Vans, Utilities
3	Step-Van, Comm. Cargo Van
4	Comm. Chassis-Cabs
5	Coupe, Sedan, Hatchback, Convertible
6	Wagon, Tracker, Jeep

frontSeatType *

Use this token to search for vehicles by front seat type.

ID	Front Seat Type
BENCH - FIXED"	Bench - Fixed
BENCH - SPLIT	Bench - Split
BUCKET	Bucket

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td>interiorClassification *</td><td>Use this token to search for vehicles by interior classification type.

<table><tr><th>ID</th><th>Interior Type</th></tr><tr><td>1</td><td>1 Row of seats</td></tr><tr><td>2</td><td>2 Rows of seats</td></tr><tr><td>3</td><td>3 Rows of seats</td></tr><tr><td>4</td><td>4 Rows of seats</td></tr><tr><td>5</td><td>5 Rows of seats</td></tr><tr><td>6</td><td>Stripped Chassis</td></tr></table></td></tr>
<tr><td>marketClassId</td><td>Use this token to search for vehicles by market class ID. When building a criterion, set the name of the criterion to this token. See Appendix C – Market Classes for a list of valid values. A list of market classes and their IDs is also available using the getMarketClasses method.</td></tr>
<tr><td>manufacturersStyle Code</td><td>Use this token to search by manufacturer's style code.
Note: When building a criterion, set the name of the criterion to this token and the value to the manufacturer's model code. Given that the manufacturer's model code is not unique across divisions, the Division should also be specified as part of the search request.</td></tr>
<tr><td>modelId</td><td>Use this token to search for vehicles by model ID. A list of models and their IDs is available using either getModelsByDivision or getModelsBySubdivision method.</td></tr>
<tr><td>numberOfDoors *</td><td>Use this token to search for vehicles by number of doors.

<table><tr><th>ID</th><th>Number of Doors</th></tr><tr><td>2</td><td>2</td></tr><tr><td>3</td><td>3</td></tr><tr><td>4</td><td>4</td></tr><tr><td>5</td><td>5</td></tr></table></td></tr>
<tr><td>powerSeatType *</td><td>Use this token to search for vehicles by power seat type.

<table><tr><th>ID</th><th>Power Seat Type</th></tr><tr><td>POWER DRIVER + PASSENGER</td><td>Power Driver and Passenger</td></tr><tr><td>POWER DRIVERS</td><td>Power Drivers</td></tr></table></td></tr>
<tr><td>primaryExteriorColor *</td><td>Use this token to search for vehicles by primary generic exterior color name. See Appendix D – Generic Colors for a list of valid values.</td></tr>
<tr><td>seatTrim *</td><td>Use this token to search for vehicles by seat trim.

<table><tr><th>ID</th><th>Seat Trim Type</th></tr><tr><td>CLOTH</td><td>Cloth</td></tr><tr><td>LEATHER</td><td>Leather</td></tr><tr><td>VINYL</td><td>Vinyl</td></tr></table></td></tr>
</table>

Enumeration Value	Description
specialtyWheels *	Use this token to search for vehicles by specialty wheel type.

ID	Specialty Wheel Type
ALUMINUM	Aluminum
CHROME	Chrome
WIRE WHEEL COVERS	Wire Wheel Covers

subDivisionId	Use this token to search for vehicles by subdivision ID. As an example, to search styles for “Kia Utility Vehicles”, pass subdivision ID “87” as the token value. A list of subdivisions and their IDs is available using the getSubdivisions method.
tireFront *	Use this token to search for vehicles by front tire type.

ID	Description
ALL SEASON	All Season
ALL TERRAIN	All Terrain
HIGHWAY	Highway
ON/OFF ROAD	On/Off Road
PERFORMANCE	Performance
TOURING	Touring

tireRear *	Use this token to search for vehicles by front tire type. See Tire Types listed above under tireFront.
tireSpare *	Use this token to search for vehicles by spare tire type.

ID	Description
compact	Compact Spare
fullSize	Full Sized Spare

transmissionType *	Use this token to search for vehicles by transmission type.

ID	Transmission Type
AUTOMATIC	Automatic
MANUAL	Manual

vehicleClassification *	Use this token to search for vehicles by vehicle classification type.

ID	Vehicle Classification
0	Car
1	Jeep
2	Light Duty Truck
3	Medium Duty Truck

Enumeration Value	Description
TechnicalSpecificationRange Search Criterion Type For any of these types, you can use the getSearchCriterionDescriptors method to get a minimum and maximum value as well as the unit of measure.	
baseCurbWeight	Use this token as a number range to search vehicles by base curb weight, where the number corresponds to the weight in lbs (US), or kilograms (Canada). The tech spec is represented as whole number in lbs.
cargoVolumeRearSeatDown	Use this token with a number range to search vehicles by cargo volume w/rear seat down, where the number corresponds to cubic feet represented as a decimal.
cargoVolumeRearSeatUp	Use this token to search vehicles by cargo volume w/rear seat up, where the number corresponds to cubic feet represented as a decimal.
cargoVolumeTotal	Use this token to search vehicles by total cargo volume, where the number corresponds to cubic feet represented as a decimal.
engineDisplacement	Use this token to search vehicles by engine displacement. Specified as a number range where the number corresponds to the displacement in liters. In Chrome's data this may be represented as 6.5L/350 or 6.5/350, 6.5L, 6.5, or 6.5 (350).
fuelEconomyCity	Use this token to search vehicles by fuel economy in the city. Specified as a number range as miles per gallon (US), or Liters per 100 km (Canada). Values are from the EPA in US and EnerGuide in Canada.
fuelEconomyCombined	Specified as a number range where the number corresponds to the average MPG of tech spec title IDs 26 and 27. This is represented as miles per gallon (US), or Liters per 100 km (Canada). Values are from the EPA in US and EnerGuide in Canada.
fuelEconomyHwy	Use this token to search vehicles by fuel economy on the highway. Specified as a number range as miles per gallon (US), or Liters per 100 km (Canada). Values are from the EPA in US and EnerGuide in Canada.
grossVehicleWeightRatingCap	Use this token as a number range to search vehicles by Gross Vehicle Weight Rating Cap (GVWR), where the number corresponds to the weight in lbs (US), or kilograms (Canada). The tech spec is represented as whole number in lbs. Number may contain a "," as in 10,000.
groundClearanceFront	Use this token as a number range to search vehicles by ground clearance front, where the number corresponds to inches (US, or in millimeters (Canada). Represented as a decimal.
groundClearanceMinimum	Use this token as a number range to search vehicles by minimum ground clearance, where the number corresponds to inches (US, or in millimeters (Canada). Represented as a decimal.
groundClearanceRear	Use this token as a number range to search vehicles by ground clearance rear, where the number corresponds to inches (US, or in millimeters (Canada). Represented as a decimal.
heightOverall	Use this token as a number range to search vehicles by overall height, where the number corresponds to inches (US, or in millimeters (Canada). Represented as a decimal.
lengthOverall	Use this token as a number range to search vehicles by overall length, where the number corresponds to inches (US, or in millimeters (Canada). Represented as a decimal.
maxPayload	Use this token as a number range to search vehicles by max payload, where the number corresponds to the weight in lbs of the GVWR cap – curb weight. The tech spec is represented as whole number in lbs (US), or kilograms (Canada). Number may contain a "," as in 10,000.
maxTrailerWeight	Use this token with a number range to search vehicles by weight distributing hitch - max trailer weight.
maxWidthWoMirrors	Use this token with a number range to search vehicles by max width without mirrors, where the number corresponds to inches represented as a decimal.

Enumeration Value	Description
passengerCapacity	Use this token as a number range to search vehicles by passenger capacity, where the number corresponds to the number of people represented as a whole number.
passengerVolume	Use this token to search vehicles by passenger volume, where the number corresponds to cubic feet represented as a decimal.
saeNetHP	Use this token to search vehicles by SAE Net Horsepower. Specified as a number range where the number corresponds to the net horsepower. This tech spec is represented as HP @ RPM "200 @ 4900".
saeNetTorque	Use this token to search vehicles by SAE Net Torque. Specified as a number range where the number corresponds to the net torque. This tech spec is represented as torque @ RPM "200 @ 4900".
stepUpHeightFront	Use this token with a number range to search vehicles by step up height – front, where the number corresponds to inches represented as a decimal.
trunkVolume	Use this token with a number range to search vehicles by trunk volume, where the number corresponds to cubic feet represented as a decimal.
wheelbase	Use this token as a number range to search vehicles by wheelbase, where the number corresponds to inches (US, or in millimeters (Canada). 7Represented as a decimal.

SearchUnitType

These are descriptions of the unit values that may be searched upon. 'n/a' is used where technical specification token does not have any unit to define, for example, the "passenger car" token does not have any unit. The unit is required to construct the technical specification value.

Enumeration Value	Description
n/a	The unit information is not available.
pounds	Pounds are units of weight in the SAE system.
inches	Inches are units of length in the SAE system
miles	Miles are units of distance in the SAE system.
feet	Feet are units of length in the SAE system.
cubic feet	Cubic Feet are units of volume in the SAE system.
horsepower	Horsepower are units of power in the SAE system.
miles per gallon	MPG (Miles Per Gallon) are units of fuel economy in the SAE system.
foot pounds	Foot Pounds are units of torque in the SAE system.
cubic inches	Cubic Inches are units of volume in the SAE system.
kilograms	Kilograms are units of weight in the METRIC system.
millimeters	Millimeters are units of length in the METRIC system.
meters	Meters are units of length in the METRIC system.
kilometers	Kilometers are units of distance in the METRIC system.
liters	Liters are units of volume in the METRIC system.
liters per 100 km	Liters per 100 KM (Liters per 100 Kilometers) are units of fuel economy in the METRIC system.

ToggleOptionResponseStatus

The ToggleOptionResponseStatus encapsulates the response of an option toggle with the complex option state returned. The ToggleResponse is used to determine if the toggle is successful and if not what steps need to be taken by the user to complete the toggle.

Name	Description
None	Indicates that there is no conflict with the selected option, and that the option toggle was successful.
UserConfirmation Needed	Indicates that the option toggle caused the selection and/or de-selection of options which must be confirmed by the user and accepted.
UserChoiceNeeded	Indicates that the option toggle caused a conflict where the user must choose among a list of options to resolve the conflict.
InvalidOption	Indicates that the option code provided did not match any of the known options.
FatalError	Indicates that the option toggle failed because of an internal error.

Appendices

The following appendices include listings of data IDs and names.

Appendix A – Categories (Generic Equipment)

Below is a list of all generic equipment descriptions as of the publication of this document. These descriptions are common across all styles. While the text description may be updated, the category ID will not be repurposed. New categories are added periodically. A list of categories and their IDs is available using the getCategoryDefinitions method.

Category ID	Category Name
1001	Driver Air Bag
1002	Passenger Air Bag
1003	Rear Air Bag
1004	Passenger Air Bag On/Off Switch
1005	Side Air Bag
1006	Rear Body Air Bag
1007	Side Head Air Bag
1008	Rear Head Air Bag
1009	Climate Control
1010	Dual Zone A/C
1011	A/C
1012	Rear A/C
1013	Alarm
1014	AM/FM Stereo
1015	Cassette
1016	CD Changer
1017	CD Player
1018	4-Wheel ABS
1019	ABS
1020	4-Wheel Disc Brakes
1021	4-Wheel Drum Brakes
1022	Front Disc/Rear Drum Brakes
1033	Cruise Control
1034	Rear Defrost
1035	Locking Front Differential
1036	Locking Rear Differential
1037	Third Passenger Door
1038	Fourth Passenger Door
1039	Child Safety Locks
1040	Four Wheel Drive
1041	All Wheel Drive
1042	Front Wheel Drive
1043	Rear Wheel Drive
1044	Dual Rear Wheels
1045	10 Cylinder Engine
1046	12 Cylinder Engine

Category ID	Category Name
1047	3 Cylinder Engine
1048	4 Cylinder Engine
1049	5 Cylinder Engine
1050	Straight 6 Cylinder Engine
1051	V6 Cylinder Engine
1052	8 Cylinder Engine
1053	Supercharged
1054	Turbocharged
1055	Front Floor Mats
1056	Rear Floor Mats
1057	Propane Fuel
1058	Diesel Fuel
1059	Gasoline Fuel
1060	Auto-On Headlights
1061	Daytime Running Lights
1062	Keyless Entry
1063	Power Door Locks
1064	Heated Mirrors
1065	Power Driver Mirror
1066	Navigation System
1067	Moonroof
1068	Sunroof
1069	Power Tilt/Sliding Sunroof
1070	T-Top
1071	Targa Roof
1072	Running Boards
1073	Additional Rear Seat
1074	Power Driver Seat
1075	Power Passenger Seat
1076	Pass-Through Rear Seat
1077	Cloth Seats
1078	Leather Seats
1079	Vinyl Seats
1080	Fixed Bench Seat
1081	Split Bench Seat
1082	Bucket Seats
1083	Manual Steering
1084	Power Steering
1085	Cell Phone
1086	Cell Phone Hookup
1087	Adjustable Steering Wheel
1088	Tires - Front All-Season
1089	Tires - Front All-Terrain
1090	Tires - Front Highway
1091	Tires - Front On/Off Road
1092	Tires - Front Performance

Category ID	Category Name
1093	Tires - Rear All-Season
1094	Tires - Rear All-Terrain
1095	Tires - Rear Highway
1096	Tires - Rear On/Off Road
1097	Tires - Rear Performance
1098	Compact Spare Tire
1099	Full Size Spare Tire
1100	Traction Control
1101	3-Speed A/T
1102	4-Speed A/T
1103	5-Speed A/T
1104	6-Speed A/T
1105	4-Speed M/T
1106	5-Speed M/T
1107	6-Speed M/T
1108	7-Speed M/T
1123	Aluminum Wheels
1124	Chrome Wheels
1125	Wire Wheel Covers
1126	Power Windows
1127	Intermittent Wipers
1128	Tires - Front Touring
1129	Tires - Rear Touring
1130	A/T
1131	M/T
1132	Sun/Moon Roof
1133	Stepside Pickup Box
1134	Continuously Variable Trans
1135	Flat 6 Cylinder Engine
1136	Premium Sound System
1138	Convertible Hardtop
1139	Air Suspension
1140	Side Stance / Highrider Suspension
1141	Quad Bucket Seats
1142	Night Vision
1143	Panoramic Roof
1144	Dual Moonroof
1145	Rollover Protection System
1146	8-Speed M/T
1147	9-Speed M/T
1148	10-Speed M/T
1149	Satellite Radio
1150	MP3 Player
1151	Fog Lamps
1152	Hybrid Fuel
1153	Power Passenger Mirror

Category ID	Category Name
1154	Heated Exterior Driver Mirror
1155	Heated Exterior Passenger Mirror
1156	Heated Driver Seat
1157	Heated Passenger Seat
1158	Privacy Glass
1159	Variable Speed Intermittent Wipers
1160	Rain Sensing Wipers
1161	Steering Wheel Controls
1162	Rear Seat Audio Controls
1163	Equalizer
1164	Auxiliary Pwr Outlet
1165	Rotary engine
1166	Vehicle Anti-Theft System
1167	Electric Fuel System
1168	HID headlights
1169	Auto-Off Headlights
1172	Luggage Rack
1173	Electrochromic rearview mirror
1174	Integrated Turn Signal Mirrors
1175	Driver Vanity Mirror
1176	Passenger Vanity Mirror
1177	Driver Illuminated Vanity Mirror
1178	Passenger Illuminated Visor Mirror
1179	Mirror Memory
1180	Rear Parking Aid
1181	Adjustable Pedals
1182	Rear Seat Heat Ducts
1183	Manual Tilt/Sliding Sunroof
1184	Fixed Sunroof
1185	Sunroof Sunshield
1186	Sunroof Wind Deflector
1187	Front Reading Lamps
1188	Rear Reading Lamps
1189	Driver Lumbar
1190	Passenger Lumbar
1191	Seat Memory
1192	Leather Wrapped Steering Wheel
1193	Active Suspension System
1194	Rear Spoiler
1195	Auto Transmission w/Manual Override
1196	Transmission Overdrive Switch
1197	Emergency Trunk Release
1198	Remote Trunk Release
1199	Trailer Hitch Receiver
1200	Front Tow Hooks
1201	Rear Tow Hooks

Category ID	Category Name
1202	Tire Pressure Monitoring System
1203	Trip Computer
1204	Universal Garage Door Opener
1205	Wheel Locks
1206	Sliding Rear Window
1207	Wheel Covers
1208	Steel Wheels
1209	Natural Gas Fuel
1210	7-Speed A/T
1211	Wireless Cell Phone Hookup
1212	Onboard Hands-Free Communications System
1213	Flexible Fuel Capability
1215	Entertainment System
1216	Bed Liner
1217	Gasoline/Propane Bi-Fuel
1218	Gasoline/Natural Gas Bi-Fuel
1219	Canopy
1220	8-Speed A/T
1221	Remote Engine Start
1222	Power Third Passenger Door
1223	Power Fourth Passenger Door
1224	Back-Up Camera
1225	Power Liftgate
1226	High Output
1227	Electronic Stability Control
1228	Brake Assist

Appendix B – Body Types

Below is a list of all body types associated to Chrome styles. This is the list of body types as of the publication of this document. A style may have one or more body types with one body type listed as primary.

Body Type ID	Body Type
1	2dr Car
2	4dr Car
3	Regular Cab Pickup
4	Extended Cab Pickup
5	Crew Cab Pickup
6	Convertible
7	Station Wagon
8	Long Bed
9	Short Bed
10	Regular Cab Chassis-Cab
11	Extended Cab Chassis-Cab

12	Crew Cab Chassis-Cab
13	Sport Utility
14	Mini-van, Passenger
15	Mini-van, Cargo
16	Full-size Passenger Van
17	Full-size Cargo Van
18	Specialty Vehicle
19	Stepside Bed

Appendix C – Market Classes

Market Class ID	Market Class Description
1	2WD Small Pickup Trucks
2	4WD Small Pickup Trucks
3	2WD Standard Pickup Trucks
4	4WD Standard Pickup Trucks
5	2WD Light Duty Chassis-Cab Trucks
6	4WD Light Duty Chassis Cab Trucks
9	Medium Duty Chassis Cab Trucks
12	2WD Special Purpose Vehicles
14	4WD Special Purpose Vehicles
16	2WD Sport Utility Vehicles
18	4WD Sport Utility Vehicles
20	Two-seater Passenger Car
21	2-door Mini-Compact Passenger Car
22	2-door Sub-Compact Passenger Car
23	2-door Compact Passenger Car
24	2-door Mid-Size Passenger Car
25	2-door Large Passenger Car
42	4-door Sub-Compact Passenger Car
43	4-door Compact Passenger Car
44	4-door Mid-Size Passenger Car
45	4-door Large Passenger Car
53	Small Station Wagon
54	Mid-Size Station Wagon
55	Large Station Wagon
61	Mini-Van (Passenger)
62	2WD Minivans
63	4WD Minivans
65	Large Passenger Vans
66	Cargo Vans
99	Commercial Vehicles

Appendix D – Generic Colors

ID	Generic Color Description
BLACK	Black
BLUE	Blue
BROWN	Brown
COPPER	Copper
GOLD	Gold
GRAY	Gray
GREEN	Green
MAROON	Maroon
NON-COLOR	Non-Color
ORANGE	Orange
PURPLE	Purple
RED	Red
SILVER	Silver
TAN	Tan
TEAL	Teal
WHITE	White
YELLOW	Yellow

Appendix E – Option Kinds

Option Kind ID	Option Kind
0	NO OPTION KIND
1	Body Code
2	Model Option
3	GVWR
4	Rear Wheel Config
5	Emission System
6	Engine
7	Transmission
8	Axle Ratio
9	Front Tires
10	Rear Tires
11	Spare Tires
12	Tires
13	Side Door Type
15	Decor Level
16	PEG
19	Quick-Order Package
20	P.E.P.
21	Option Package

Option Kind ID	Option Kind
26	Seating Arrangement
27	Seat Type
28	Seat Trim
29	Paint Type
30	Striping Option
32	Air Cond Option
33	Radio
34	Rear Defogger
37	Top Color
39	Payload Package
40	Differential
41	Wheel Type
43	Chassis Package
44	Roof Type
45	Ambulance Package
46	Glass Package
47	Region
49	9W_ and 9V_ Paints
52	Retail Amenity Delete
55	Owner Follow-up
56	Upfitter Package
59	Regional Dest Chrg Adj
60	Gas Guzzler Tax
65	Required Option
66	2nd Row Seat Type
67	3rd Row Seat Type
97	Ford Special Fleet Account Credit
98	New Jersey Cost Surcharge
99	Deletion
100	Fleet Incentive
103	VQ_ Fleet Options
104	Bid Assistance
106	Fleet Only
109	Fleet Net Invoice
110	Vehicle Application
111	Brake System
112	Exhaust System
113	Air Cleaner
115	Battery
116	Alternator
117	Front Axle
118	Front Suspension
119	Rear Axle
120	Rear Suspension
121	Wheelbase
122	Frame Type

Option Kind ID	Option Kind
123	Front Wheels
124	Rear Wheels
125	Fuel Tank
126	Driver Seat
127	Passenger Seat
129	Front Tire Tread
130	Rear Tire Tread
131	Interior Trim
132	Front Tire Brand
133	Rear Tire Brand
136	Air Compressor
140	Airbags
141	Side Mirrors
142	Block Heater
143	Power Take Off
144	Rear Shock Absorber
145	Rear Stabilizer Bar
148	Dealer Installed Options
149	Rear Seat
150	Bumper
152	Radiator
153	Bolted or Riveted Suspension

Appendix F – Editorial Content Types

Description
Model Strengths
Model Changes
Model Value
Model Overview

Appendix G – Structured Consumer Information Types

Type Name (ID)	Name
Rebate (1)	
	Incentive Note
	Cash Incentive Minimum
	Cash Incentive Maximum
	Cash Incentive Bonus Cash
	Cash Incentive Note
	Cash Finance Rule

Type Name (ID)	Name
	Financing Incentive Min
	Financing Incentive Max
	Financing Bonus Cash
	Financing Incentive Note
	Expiration Date
	Resource Name
	Resource Date
Crash Test (4)	
	Frontal Driver
	Frontal Passenger
	Side Driver
	Side Rear Passenger
	Crash Test Note
	NHTSA Rollover Rating
	Rollover Note
Recall (3)	
	NHTSA CAMPAIGN ID
	Mfg's Report Date
	Component
	Potential Number of Units Affected
	Summary
	Consequence
	Remedy
	Notes
	Dates of Manufacture
	Models Involved
	Manufacturer Recall No.
Warranty (2) See the list of Conditions below for valid values.	
	Warranty Note
	Basic Years
	Basic Miles/km
	Basic Note
	Drivetrain Years
	Drivetrain Miles/km
	Drivetrain Note
	Corrosion Years
	Corrosion Miles/km
	Corrosion Note
	Rear Axle Years
	Rear Axle Miles/km
	Rear Axle Note
	Frame Rail Years
	Frame Rail Miles/km
	Frame Rail Note
	Hybrid Component Years

Type Name (ID)	Name
	Hybrid Component Miles/km
	Hybrid Note
	Emissions Years
	Emissions Miles/km
	Emissions Note
	Roadside Assistance Years
	Roadside Assistance Miles/km
	Roadside Assistance Note

Condition (applies to Warranty items only)

Condition Type	Condition Note
Drivetrain	
	Gas Engine
	Diesel Engine
	Gas 8100 Engine
	Vortec 8100 Engine
	Duramax 6600 Engine
	Duramax 7800 Engine
	Caterpillar Engine
	Caterpillar C7 Diesel Engine
	Manual Transmission
	Allison Basic
	Allison 1000
	Allison 2000
	Allison 2400
	Allison 2500
	Allison 3000MD
	Allison HS
	Allison RS
	Allison 1000MH
	Allison 2200MH
	Allison 1000EVS
	Allison 2200EVS
	Allison 3000EVS
	Allison 3500EVS
	Allison 3050
	Allison 3060
	Allison 3066
	Allison 3550
	Allison 3560
	Allison 3500MD
	AISIN
	Turbo Diesel
	Rear Axle
	Frame Rails and Cross Members

Condition Type	Condition Note
Corrosion	
	Rust-Through
	Cab Corrosion-Perforation
Emissions	
	Gas Engine
	Diesel Engine
	Short Term California
	Long Term California