



10/24/2018

# EAS – Rating Procedures



Karthik Velusamy

## 1.1 Get everything request decode

Get everything request will return all the products and rates mapped to the dealer, the request would be in the below format,

```
EXECUTE [dbo].[geteverything]
    @ai_program_id = 20284
    ,@ai_dealer_number= 'VW405021'
    ,@as_vin= '3VW4T7AUXGM017959'
    ,@ai_odometer= 100
    ,@ad_sale_date = '2017-05-17'
    ,@ad_inservice_date= '2016-11-05'
    ,@ab_is_after_sale = 'N'
    ,@ad_vehicle_purchase_date= '2017-05-17'
    ,@as_product_type= '*'
    ,@ai_vehicle_year= '2016'
    ,@as_vehicle_make= 'VOLKSWAGEN'
    ,@as_vehicle_model= 'GOLF GTI'
    ,@as_vehicle_trim= 'S'
    ,@as_vehicle_condition = 'NEW'
    ,@as_finance_type = 'LEASE'
    ,@as_groupcode = null
    ,@as_correlation_id = null
    ,@as_vendor = null
    ,@as_debug = 'PROD'
go
```

**@ai\_program\_id:** Program id is the value to which the dealer is associated. Program id can be obtained from the below table,

```
Select * from program; -- program_id
```

**@ai\_dealer\_number:** cms\_dealer\_number for the dealer must be provided. Dealer number can be obtained from the below table,

```
Select * from v_dealer; -- cms_dealer_number
```

**@as\_vin:** Provided VIN number is checked with the VIN pattern from the Vin details table. Vin pattern is obtained from the below vin number using the below,

If the provided VIN number is 17 digits, remove the 9<sup>th</sup> digit from the right and then remove the last 6 digits.

**@ai\_odometer:** Provide the odometer value. Based on the odometer value sku eligibility is based on that odometer value.

**@ad\_sale\_date:** Contract sale date must be mentioned for this parameter.

**@ad\_inservice\_date:** In service date for the vehicle must be provided for the parameter. It's the date from which the vehicle is on road. (Similar to RC book.)



**@ab\_is\_after\_sale:** if the contract is being brought during the time of vehicle purchase.

- a. This flag is sent as 'N', if this is the time of vehicle purchase is provided as 'Y' in the portal.
- b. This flag is sent as 'Y', if this is the time of vehicle purchase is provided as 'N' in the portal.

**@ad\_vehicle\_purchase\_date:** Vehicle purchase date must be mentioned for this parameter.

**@as\_product\_type:** Product type for the product code can be provided, generally '\*' is provided.

Select \* from ref\_product\_code; -- Product code & product type id.

Select \* from ref\_product\_type; -- Product type.

**@as\_vehicle\_condition:** Vehicle condition can be provided with the below values,

1. New – New vehicle.
2. CPO – Certified Pre-Owned.
3. Used – Pre-Owned.
4. DEMO – Demo vehicle.

**@as\_finance\_type:** Finance type can be provided with the below values,

1. Finance.
2. Cash
3. Lease

If the provided VIN number did not decode, then the below parameters are used,

**@ai\_vehicle\_year:** Vehicle year must be provided. This value depends on the vehicle age for the sku eligibility conditions.

**@as\_vehicle\_make:** Manufacturer of the vehicle is provided as the parameter.

**@as\_vehicle\_model:** Model is provided as the parameter.

**@as\_vehicle\_trim:** If trim value is available, it is provided as the parameter.

### 1.1.1 Get everything procedure

Based on the above input values, get everything procedure returns the pricing for the product & product plan codes. The procedure works in the below order,

1. VIN number is decoded to obtain the VIN pattern, regarding the process to decode a VIN number please check the above section.
2. Based on the VIN pattern, warranty details are obtained from VIN\_DETAILS table. Please note that the obtained VIN pattern should have a record in VIN\_DETAILS table.
3. VIN information is obtained from Data one or auto check and similar websites.
4. Based on the new code changes for ECOMSERV=53, if the provided VIN number has a record in VCI\_VEHICLE\_RETAIL\_SALES table, in-service date is replaced with the value in the



table. Please note that for those VIN number, the in-service date cannot be manually provided.

5. For the provided VIN number, make, model and other information's are obtained.
6. The below input values determine the eligibility for in & out of warranty plans. (Based on the input values & information obtained from VIN\_DETAILS table.)
  - Contract sale date
  - In-service date
  - Odometer
7. Required temporary tables are created.
8. All the possible product plan sku's are obtained based on the products mapped for the dealer/program is obtained.
9. Classing information is obtained for the make/model & product codes. (Based on classing method.)
10. If the product is excluded for the make & model, the product plan sku's are deleted from the temporary table for the product.
11. Active rate system mapping for the dealer is obtained and inserted into a temporary table.
12. Based on the rate method, odometer & classifications, all available product plan sku price parameters are obtained and inserted into a temporary table.
13. Required information's for product plan sku are obtained.
14. Product plan sku's that doesn't meet the eligibility conditions are deleted. Eligibility conditions would be odometer, vehicle age, time months, vehicle and finance conditions.
15. Based on the above obtained information's (product plan sku/rate system & product plan sku price parameter), we can obtain the required pricing details. The pricing is obtained to the portal/API.
16. Retail cost is updated for Penske dealers.
17. Surcharge value for the product plan sku's are obtained and returned to portal/API.
18. Input values are inserted into econ\_log table.

### 1.1.2 Get everything dependent objects

The geteverything() rating procedure depends on the below tables,

- Agent\_Commission\_Details
- Dealer
- Dealer\_Program\_Association
- Dealer\_Product\_Code
- Dealer\_Product\_Plan\_Exception
- Econ\_Childproducts\_Log
- Econ\_Log
- Econ\_Products\_Log
- Econ\_Surcharges\_Log
- Program
- Program\_Vehicle\_Class\_Mapping



- Program\_Vehicle\_Class
- Program\_Classing\_Method
- Product\_Plan
- Product\_Plan\_Association
- Product\_Plan\_Sku
- Product\_Plan\_Sku\_Price
- Product\_Plan\_Sku\_Price\_Exception
- Product\_Plan\_Sku\_Price\_Header
- Product\_Plan\_Sku\_Price\_Parameter
- Product\_Plan\_Sku\_Price\_Version
- Product\_Plan\_Xref
- Plan\_Vsc\_Surcharge\_Eligibility
- Ref\_Amount\_Type
- Ref\_Product\_Code
- Ref\_Product\_Type
- Ref\_Rate\_Method
- Ref\_Classing\_Method
- Ref\_Make
- Ref\_Model
- Ref\_Trim\_Level
- Ref\_Lookup\_Value
- Ref\_Lookup\_Name
- Rate\_System
- Rate\_System\_Application
- Sku\_Vsc\_Eligibility
- VCI\_Vehicle\_Retail\_Sales
- Vsc\_Surcharge

### 1.1.3 Exceptions handled in geteverything()

Some of the major changes handled in the geteverything() procedure are,

1. Penske retail rate change – Retail rate (MSRP) for Penske dealers would be updated based on adding the Dealer cost to a commission amount.
  - (Product\_Plan\_Sku\_Price\_Exception table.)

Also, please note that this change would work only if the dealer should be a part of the dealer group in the below tables,

- Dealer\_Group
- Dealer\_Group\_list

2. After Market Codes – AM codes would be replaced (instead of the EAS product codes) in the response that is being sent to certain dealers. This information would be available in the below table,
  - Product\_Plan\_Xref.

## 1.2 Get specific product request decode

Get specific product request will return the rates for the provided product code/plan/term/mileage and deductible values, the request would be in the below format,

```
exec [dbo].[getSpecificProduct]
-- Contract Number: 6000004074
@ai_program_id = 20265,
@as_dealer_number = 'PORS0430',
@as_vin = 'WP1AA2A27GKA13970',
@ai_odometer = 42,
@ad_sale_date = '06/10/16',
@ad_inservice_date = NULL,
@ab_is_after_sale = 'NO',
@ad_vehicle_purchase_date = '06/10/16',
@as_product_type = '*',
@ai_vehicle_year = NULL,
@as_vehicle_make = NULL,
@as_vehicle_model = NULL,
@as_vehicle_trim = NULL,
@as_vehicle_Condition = 'New',
@as_finance_type = 'LEASE',
@as_plan_sku = '157667',
@as_product_code = 'POLS',
@ai_term = 24,
@an_deductible = 0,
@as_product_plan_code = 'POLSLEAS',
@as_mandatory_surcharges = null,
@ai_term_mileage = 0,
@as_debug = 'PROD'
```

go

Please note that the input values are provided from CMS system.

**@as\_plan\_sku:** Product plan sku value is passed as an input to the procedure. Please see the above section for more details on product plan sku.

**@as\_product\_code:** Product code for which rates are required, is passed as input.

**@ai\_term:** Term in months is provided.



**@an\_deductible:** Deductible information is provided.

**@as\_product\_plan\_code:** Product plan code/CMS coverage code value is provided.

**@as\_mandatory\_surcharges:** Surcharge (if exists for the contract) keyword is provided.

**@ai\_term\_mileage:** Term in miles information is provided as input.

### 1.2.1 Get specific product procedure

Based on the above-mentioned input values, get specific product procedure returns the pricing for the product/plan/term/deductible values. The procedure works in the same way as get everything procedure.

Difference between get everything and get specific product procedures is that, get everything procedure returns the rates for all the product/plan/term/deductible values for a dealer. But get specific product procedure returns rates only for the provided product/plan/term/deductible input values.

### 1.2.2 Calling Get specific product procedure

Contracts in CMS system hits EAS to get the pricing. This is done in the below order,

1. sg\_eas\_getdisbursementsforproduct function is called to rate a contract in CMS system.
2. The function in turn calls sg\_eas\_getcontractdataforweb, it obtains all the required input values of the contract.
3. Obtained input values are structured into a query and passed as an input value to the internal web services.
4. Internal web services in turn calls getspecificproduct procedure in EAS.
5. Rates obtained from the procedure is returned to the web service, which in turn returns to the CMS function.

The rates are populated in sg\_con\_d4 and other tables for the contract.

### 1.2.3 Exceptions handled in getSpecificProduct()

Some of the major changes handled in the getSpecificProduct() procedure are,

1. CFC Price exception – This works in a similar way as mentioned above for Penske retail rate change.

CFC functionality would be explained in a separate document.



### 1.2.4 EAS Rating failure

EAS Rating failure error is returned when no rates are being returned by getspecificproduct procedure for a contract. Please find the list of error codes and messages that are being returned for a contract below,

CMS Error Code	CMS Error Message
414	Dealer# is not valid for the contract
402	Product Code is not valid
403	Coverage code is not valid
404	Product Code is not assigned to the dealer
405	Product code not valid for dealer based on sale/business Dates
406	Term In months is not valid for the contract
407	Term In miles is not valid for the contract
408	Deductible is not valid for the contract
409	Coverage code is not valid based on odometer or in-service date
413	Coverage code is not valid based on the finance type
415	Rating details not found for the provided input values