

Optimizing Cross-Validation Rules in General Ledger

an eprentise white paper



tel: 407.591.4950 | toll-free: 1.888.943.5363 | web: www.eprentise.com

Author: Helene Abrams
www.eprentise.com

© 2022 eprentise, LLC. All rights reserved.

eprentise, FlexField Express and FlexField are registered trademarks of eprentise, LLC.

Oracle is a registered trademark of Oracle Corporation.

All other company or product names are used for identification only and may be trademarks of their respective owners.

Cross-validation rules determine which segment values in your chart of accounts (or other key flexfields) can be used together. Upon data entry, the rule determines and controls the valid values that may be used in conjunction with other values. This paper shows how to set up Cross-validation rules, discusses the design considerations for your chart of accounts so that you can reduce the number and complexity of your Cross-validation rules, and concludes with a list of the top seven things to remember when designing Cross-validation rules.

Setting up Cross-validation Rules

A Cross-validation rule applies only to a single chart of accounts structure or accounting flexfield. If you have more than one accounting flexfield, you will need to define Cross-validation rules for each COA structure name (even if different charts of accounts use the same value sets). Cross-validation rules only work on data entry and for the first time a combination is used. If a user tries to create a new combination that violates a new rule, they will get an error message and the combination will not be accepted. Cross-validation rules have no impact on code combinations that are already in use, so it is important to design and set up your Cross-validation rules before you enter transactions.

You set up Cross-validation rules by describing valid combinations. In the **Define Cross-Validation Rules** form, you will define rule elements that specify low and high values for each segment. Ideally, those low and high values are a range of individual values for each segment. You define each rule element as either an **Include** or **Exclude** type. Include contains all the values in the specified range, and Exclude eliminates values that are not valid combinations. With Include ranges you list valid combinations, while Exclude ranges list invalid combinations.

The screenshot shows the 'Cross-Validation Rules' form. It includes input fields for 'Application', 'Structure', 'Flexfield Title', and 'Description'. Below these is a table with columns 'Name', 'Description', and 'Enabled'. The 'Enabled' column contains a checked checkbox and a dropdown arrow. Below the table are fields for 'Error Message', 'Error Segment', 'From', and 'To'. At the bottom is a section for 'Cross-Validation Rule Elements' with a table with columns 'Type', 'From', and 'To'. The 'Type' column has a dropdown arrow.

Example Cross-validation Rules

Let's assume that your chart of accounts contains the following 5 segments:

	Company		Department		Account		Product Line		Intercompany	
	3 digits Numeric		4 digits Numeric		5 digits Numeric		3 digits Alphanumeric		3 digits Numeric	
Example Values	001	US Holding Co	1010	Administrative	40000	Revenue	C99	CPUs	001	US Holding Co
	010	ABC Hard Drives, Ltd.	5042	Consulting	20000	Liabilities	D47	Internal Hard drives	010	ABC Solutions, Ltd.
	320	DEF Systems, LLC	7640	Repair and Maintenance- Internal Hard Drives	50013	Expenses Replacement parts for Internal Hard Drives	M50	Memory	320	DEF Systems, LLC
	731	GHI Enterprises, Inc.	2020	Finance	35000	Sales - Hardware	S10	Installation Services	731	GHI Enterprises, Inc.

Example 1

Let's assume that you want to prevent revenue account values (between 30000 and 40000) from being used with the balance sheet department values (between 1000 and 1999). To do this, let's create a rule called Revenue Accounts –Balance Sheet Departments.

The Error Message is what your users will see when they try to enter an incorrect combination, and the Error Segment is where the cursor returns in the event of an error. In this case, let's make the error message, "CV – Revenue accounts may not be used with a balance sheet department." When determining where the cursor will return if the rule fails, identify the segment most likely to be in error. In this case, let's assume that the department value is where you expect that most of the errors will occur. Department will be the error segment.

Next, you need to define the Cross-validation Rule Elements.

The first rule should always be an Include statement that includes the entire range of values for each segment.

INCLUDE	Low Value	High Value
Company	000	999
Department	0000	9999
Account	00000	99999
Product Line	000	ZZZ
Intercompany	000	999

Then you will create Exclude rule statements or elements that limit the values that can be used. In this case, only one exclude statement is necessary. The following rule element prevents accounts 30000 to 40000 from being used with departments 1000-1999.

EXCLUDE	Low Value	High Value
Company	000	999
Department	1000	1999
Account	30000	40000
Product Line	000	ZZZ
Intercompany	000	999

Example 2

Next, let's create a rule that says only department 7640 can be used with any product line that starts with S. Again, the rule starts with an Include statement that includes the entire range of values for each segment.

INCLUDE	Low Value	High Value
Company	000	999
Department	0000	9999
Account	00000	99999
Product Line	000	ZZZ
Intercompany	000	999

You will need to exclude the ranges above and below the department value, and the ranges above and below the product line value. It will be 4 rule elements in total. The following exclude statement prevents the departments **below 7640** from using any product line that starts with anything **below S**.

EXCLUDE	Low Value	High Value
Company	000	999
Department	0000	7639
Account	00000	99999
Product Line	000	RZZ
Intercompany	000	999

The following exclude statement prevents the departments **above 7640** from using any product line that starts with anything **below S**.

Optimizing Cross-validation Rules in General Ledger

EXCLUDE	Low Value	High Value
Company	000	999
Department	7641	9999
Account	00000	99999
Product Line	000	RZZ
Intercompany	000	999

The following exclude statement prevents the departments **below 7640** from using any product line that starts with anything **above S**.

EXCLUDE	Low Value	High Value
Company	000	999
Department	0000	7639
Account	00000	99999
Product Line	T00	ZZZ
Intercompany	000	999

The following exclude statement prevents the departments **above 7640** from using any product line that starts with anything **above S**.

EXCLUDE	Low Value	High Value
Company	000	999
Department	7641	9999
Account	00000	99999
Product Line	T00	ZZZ
Intercompany	000	999

This cross-validation rule has 5 lines. The first line includes all possible values, and the next 4 exclude lines prevent the departments above or below 7640 from being used with any product line above or below those values starting with S.

A Good Chart of Accounts Design Helps in Optimizing Cross Validation Rules

Let’s now look at that same chart of accounts to see what would happen if the COA design did not follow the practices of good design. [Click here](#) to read more on best practices of chart of accounts design.

Out of Range Example 3

First, let’s see what would happen in Example 1 above if the values for the departments were not in a range. Let’s say instead that you want to prevent revenue account values (between 30000 and 40000) from being used with any department values other than these 5 values: 3001, 5057, 6124, 8537, and 9905. Now, my rule elements are much more complex because I cannot use a range for the department values.

I still start with my include statement.

Optimizing Cross-validation Rules in General Ledger

INCLUDE	Low Value	High Value
Company	000	999
Department	0000	9999
Account	00000	99999
Product Line	000	ZZZ
Intercompany	000	999

The next step is to create Exclude rules for all the values above and below each of the department values. This statement prevents all departments below 3001 from being used.

EXCLUDE	Low Value	High Value
Company	000	999
Department	0000	3000
Account	30000	40000
Product Line	000	ZZZ
Intercompany	000	999

This statement prevents all departments greater than 3001 and less than 5057 from being used.

Optimizing Cross-validation Rules in General Ledger

EXCLUDE	Low Value	High Value
Company	000	999
Department	3002	5056
Account	30000	40000
Product Line	000	ZZZ
Intercompany	000	999

This statement prevents all departments greater than 5057 and less than 6124 from being used.

EXCLUDE	Low Value	High Value
Company	000	999
Department	5058	6123
Account	30000	40000
Product Line	000	ZZZ
Intercompany	000	999

This statement prevents all departments greater than 6125 and less than 8537 from being used.

Optimizing Cross-validation Rules in General Ledger

EXCLUDE	Low Value	High Value
Company	000	999
Department	6125	8536
Account	30000	40000
Product Line	000	ZZZ
Intercompany	000	999

This statement prevents all departments greater than 8537 and less than 9905 from being used.

EXCLUDE	Low Value	High Value
Company	000	999
Department	8538	9904
Account	30000	40000
Product Line	000	ZZZ
Intercompany	000	999

This statement prevents all departments greater than 9906 from being used.

EXCLUDE	Low Value	High Value
Company	000	999
Department	9906	9999
Account	30000	40000
Product Line	000	ZZZ
Intercompany	000	999

I have very quickly gone to 6 lines and a lot more complexity. Imagine if the rule were to restrict more values, or if neither of the two segments that I am trying to control were in ranges. (In the above example, my accounts are still in a range.)

Similar Data in Multiple Segments – Example 4

In the above chart of accounts, I need to restrict Department 7640 and Account 50013, and Product Line S10 only to Company 010, Account 50013 only to Department 7640, and Product Line S10 only to Department 7640.

It is much too complex to try to exclude more than two segments in a cross validation rule, so I end up with 11 rule elements.

Optimizing Cross-validation Rules in General Ledger

INCLUDE	Low Value	High Value
Company	000	999
Department	0000	9999
Account	00000	99999
Product Line	000	ZZZ
Intercompany	000	999

Exclude Rules: CV – Department 7640 may only be used with Company 010.

EXCLUDE	Low Value	High Value
Company	000	009
Department	7640	7640
Account	00000	99999
Product Line	000	ZZZ
Intercompany	000	999

Optimizing Cross-validation Rules in General Ledger

EXCLUDE	Low Value	High Value
Company	011	999
Department	7640	7640
Account	00000	99999
Product Line	000	ZZZ
Intercompany	000	999

Exclude Rules: CV -Account 50013 may only be used with company 010.

EXCLUDE	Low Value	High Value
Company	000	009
Department	0000	9999
Account	50013	50013
Product Line	000	ZZZ
Intercompany	000	999

Optimizing Cross-validation Rules in General Ledger

EXCLUDE	Low Value	High Value
Company	011	999
Department	0000	9999
Account	50013	50013
Product Line	000	ZZZ
Intercompany	000	999

Exclude Rules: CV -Product Line S10 may only be used with Company 010.

EXCLUDE	Low Value	High Value
Company	000	009
Department	7641	9999
Account	00000	99999
Product Line	S10	S10
Intercompany	000	999

Optimizing Cross-validation Rules in General Ledger

EXCLUDE	Low Value	High Value
Company	011	999
Department	7641	9999
Account	00000	99999
Product Line	S10	S10
Intercompany	000	999

Exclude Rules: CV -Account 50013 may only be used with Department 7640.

EXCLUDE	Low Value	High Value
Company	000	999
Department	0000	7639
Account	50013	50013
Product Line	000	ZZZ
Intercompany	000	999

Optimizing Cross-validation Rules in General Ledger

EXCLUDE	Low Value	High Value
Company	000	999
Department	7641	9999
Account	50013	50013
Product Line	000	ZZZ
Intercompany	000	999

Exclude Rules: CV -Product Line S10 may only be used with Department 7640

EXCLUDE	Low Value	High Value
Company	000	999
Department	0000	7639
Account	00000	99999
Product Line	S10	S10
Intercompany	000	999

EXCLUDE	Low Value	High Value
Company	000	999
Department	7641	9999
Account	00000	99999
Product Line	S10	S10
Intercompany	000	999

As these very simple examples show, it is very easy to quickly get to thousands of cross validation rules to maintain. Notice also, that it was much more difficult to determine the ranges for alphanumeric data. If your chart of accounts follows [the basic principles of good design](#) of keeping values in ranges in each segment, keeping only one kind of data in a single segment, and putting similar data only in one segment, you will be able to simplify your cross validation rules and significantly reduce maintenance of the rule elements as your business changes.

7 Criteria for Designing Good Cross-validation Rules

1. Design your chart of accounts with cross-validation rules in mind (keep values in ranges, put similar data in only one segment, and keep each segment for only one type of data, use alphanumerics only when necessary).
2. Start each rule with an include statement that includes the entire range of values.
3. Limit each rule to only 2 segments.
4. Make your error messages very descriptive so that when a user gets an error, he or she understands what changes need to be made to the entered code combination.
5. Use several simple rules rather than one complex rule.
6. Start your Cross-validation Error Messages with CV so that the user knows that the combination is failing because of a Cross-validation rule (as opposed to a security rule, or an end-dated value) when they get the error message.
7. Use a single chart of accounts for all your operations. This greatly reduces the number of cross-validation rules necessary.

Curious?

For more information, please call **eprentise** at **1.888.943.5363** or visit **www.eprentise.com**.



About eprentise

eprentise provides transformation software products that allow growing companies to make their Oracle® E-Business Suite (EBS) systems agile enough to support changing business requirements, avoid a reimplementation and lower the total cost of ownership of enterprise resource planning (ERP). While enabling real-time access to complete, consistent and correct data across the enterprise, **eprentise** software is able to consolidate multiple production instances, change existing configurations such as charts of accounts and calendars, and merge, split or move sets of books, operating units, legal entities, business groups and inventory organizations.