

Productized Rules = Death of Custom Functions

Data-driven Edit Checks, Rules, and Dynamics in Veeva Vault EDC

Writing rules, edit checks, and their correlated test scripts is the most time-consuming part of building the study database in an Electronic Data Capture (EDC) system. Veeva is transforming the build process with an advanced study design environment where users configure most edit checks and use a powerful rules engine when creating rules and advanced checks right within the system, rather than outside of it. Perhaps the biggest advance is Veeva's progress toward eliminating the need for custom functions.

Terminology can vary by organization, therefore for clarity, it is important to level-set on the nomenclature used in this paper. **Edit checks** are logical tests applied to fields in the electronic case report form (eCRF) for the purpose of catching errors and improving data accuracy. **Derivations** use existing data to calculate a related value, such as using height and weight to calculate BMI. Derivations save time by automating calculations and reducing manual data entry. **Rules and dynamics** provide automation by performing actions such as sending emails, adding forms to visits, or displaying invisible fields. **Custom functions** are pieces of code written outside of the EDC and inserted into the system to perform an intended action.

Traditional EDC systems were developed years ago when clinical trials were more straightforward but with increased complexity in trial designs, now tend to require many custom functions to augment their functionality. Many standard rules, such as sending configurable emails, as well as cross-form edit checks and derivations require custom functions. Because most custom functions are written in C# or SQL, firms need to hire expensive software engineers with traditional programming skills and training.



PROBLEMS WITH CUSTOM FUNCTIONS:

- Time consuming to build
- Time consuming to test
- Add risk to database changes
- Increase staffing costs
- Impact system performance

Creating Rules and Edit Checks in Vault EDC

Veeva has a governing design principle to automate simple tasks and make the hard tasks easy. Applying that principle to rules and edit checks means automating the creation of simple univariate checks and providing an intuitive wizard to simplify the creation of complex rules.

/// *The real difficulty in building a study lies in the edit checks – that’s where you require a programming background. But that’s not needed with Vault EDC. If you know how to use Boolean logic and if/then statements, you can write edit checks in Studio. It is really very intuitive and easy to use. ///*

– **Harbal Sidhu**, director, EDC Programming, ICON

Automating simple checks

Simple data validation rules enforce constraints on field values, such as whether a value is required or whether the value is within a specified range. These simple checks are created and managed by the system based on the properties entered when defining the item. For example, in the properties panel for a temperature field, you can specify minimum and maximum temperature values. You do not need to program data validation rules for required fields, range checks, and future dates.



REDUCING EDIT CHECKS AT A GLOBAL TOP 20

One global Top 20 pharma eliminated 250 edit checks across 49 CRFs. The savings are even greater if considering the number of instances of these CRFs in a study, because edit checks are reused when copies of CRFs are made for multiple visits.

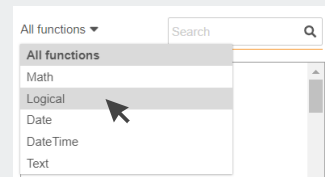
Simplifying Complex Checks

Vault EDC provides a powerful yet easy-to-use rules engine for creating edit checks, dynamics, and derivations. The script editor interface makes it easy to create sophisticated edit checks or business rules. First, you specify your data variables, then you select functions from a library of pre-defined options. And lastly, specify the desired action, such as running a query or sending an email.

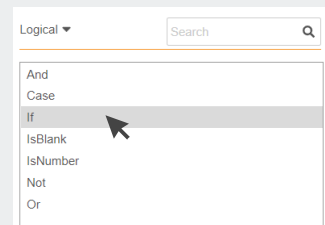
ADDING FUNCTIONS

You can add functions by typing them directly in your rule expression or by using the function selector tool.

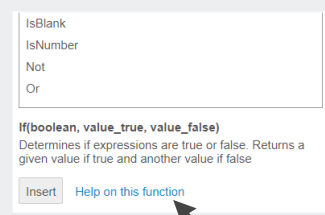
1. Filter the options by function type.



2. Double-click on the function to add it to your expression.



3. Single-click on the function to see a brief description of the function. Also click **Help on this function** to get more explanation and advice on how to use it.



Much like Google’s auto-complete for search terms, Vault provides auto-complete suggestions to help you write rules. Using the context of existing identifiers, variables, operators, and functions, Vault intelligently suggests appropriate options for your next step.

Vault can perform a large variety of tasks based on the rules and entered values, such as:

- Adding events, visits, or forms. For example, adding forms based on the route of administration.
- Sending an email if an adverse event is marked “Yes” for being serious.
- Deriving the value of a related item. For example, calculating a patient’s BMI or returning a value of true if the tumor size in the second visit is less than the first.

Advanced Study Designs with Rules and Dynamics

You can use rules in your study to dynamically add treatment cycles, visits, forms, and fields based on user-entered data. For example, you can configure rules to control which treatment cycle (event group) appears in a subject’s casebook based on their cohort or randomization number.

You can also connect multiple rules in your formula. For example, after deriving a change in tumor size that reflects a reduction in size, automatically add a new treatment cycle (event group) with related visits and forms. Complex study designs, such as adaptive platform oncology trials, can be configured with dynamic cycles, visits, and forms and a relatively small number of governing rules.

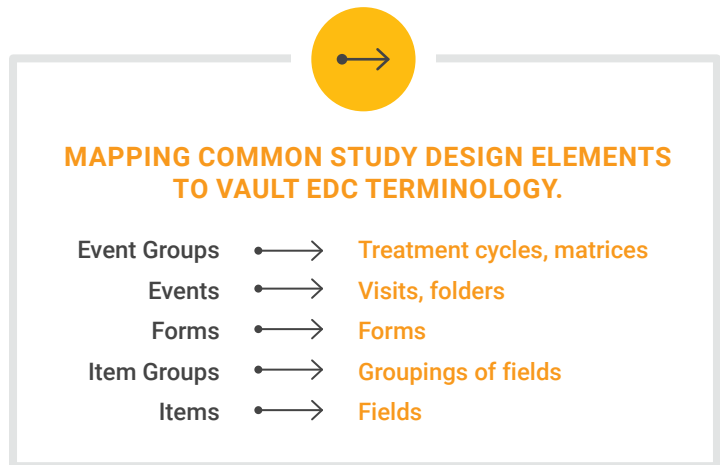
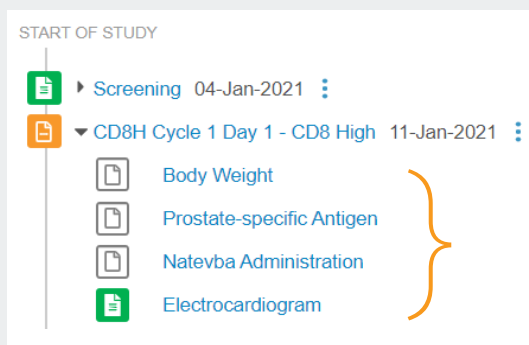


FIGURE 1. DYNAMICALLY ADDING FORMS TO A VISIT



A simple rule, with two lines of script, adds these three dynamic forms to the Cycle 1, Day 1 visit in the CD8 High arm of this study, when the site selects a visit date.

Configured Form Linking

In traditional systems, linking forms requires either writing custom functions to create dynamic search lists, or relying on sites to enter the right ID for cross referencing. If using dynamic search lists, you have all the challenges of custom functions – programming effort, risk, longer testing timelines, and an impact on performance. If sites enter IDs manually, the form must be lengthened to allow sites to enter multiple IDs and there is increased burden on the site, risk of human error, and resulting edit checks to ensure IDs match.

Veeva eliminates that trade-off and the need for custom functions with a form-linking feature for data entry. For example, site users can click to link one or more concomitant medication forms to an adverse event or medical history form to group related data or to indicate cause and effect. Form linking also enables field values to be automatically copied from one form to the other, such as copying the indication value between the adverse event and ConMed. In this way, duplicate data entry is not required.

Form linking is another capability within Vault EDC that reduces the need for edit checks and custom functions, improving the speed and quality of study builds.

Custom Functions within Vault EDC

While Veeva is seeking to eliminate the need for custom functions, there are a few scenarios where custom functions, called triggers within Vault EDC, are still needed. Veeva's services team writes triggers on behalf of our customers to ensure their quality and performance, and as part of our commitment to eradicate custom functions. By writing all the triggers, Veeva's team learns which capabilities are needed and adds the necessary functionality to the roadmap.

■ ■ *Needing a custom function for every aspect that doesn't quite fit in the system causes a lot of delays, not just for the build but for validation and when making changes. Using the rules in Vault EDC as opposed to custom functions is a big game changer for us. It makes complex designs easy to program. ■ ■*

– **Tanya du Plessis**, vice president of data strategies and solutions, Bioforum – The Data Masters

Eliminating the Need to Write Rules

Eliminating unnecessary checks

Veeva also uses the rules engine to eliminate edit checks where possible. Instead of requiring sites to enter data in inches and centimeters, or Celsius and Fahrenheit, and creating edit checks for both, Vault EDC will use unit conversion formulas to convert and store entered values with the correct unit, regardless of a site's localized entry format.

Vault EDC reduces the number of edit checks that need to be created, tested, and maintained. By building quality controls into the forms themselves, you reduce the number of edit checks needed.

- Progressive display logic controls how CRFs perform and what fields are displayed for data entry. You no longer need to write rules to check for conflicting data, such as having an entry of “Ongoing = Yes” and “End date = Complete.”
- Pre-defining ranges for expected values.

“One of the things I like about Vault EDC is the reduction in edit checks. For example, for typical study you’ll need 20 checks just to check future dates. As a CRO building hundred studies a year, that’s 2000 edit checks. And we don’t just program, we also validate by writing three scripts, one for future date, current date and a past date, just to test that it’s working. With Veeva, future date checks are just a setting, you don’t have to write any scripts.”

– **Harbal Sidhu**, director, EDC Programming, ICON

Configurable date comparison checks

In most studies, a large portion of the edit checks are date comparison rules. While these are relatively simple rules, they are numerous, manual, and tedious to write. Not only must companies write the edit checks, they must also write the test scripts to verify that the checks are functioning as intended. Veeva has created a configuration tool that automatically generates date field comparison rules based on the criteria provided, such as “subject enrollment date must come after informed consent date.” Study designers specify date comparisons within a dedicated interface that eliminates the need for a standalone specification document. Because this specification tool is purpose-built for capturing requirements, it is easier to use than Excel, and Vault automatically generates the rule expression for you.

FIGURE 2. CREATE RULES WITH AN INTUITIVE POINT-AND-CLICK INTERFACE DESIGNED FOR CAPTURING DATE COMPARISON REQUIREMENTS.

Search <input type="text"/>		Data Type: Date	Show: Specified			
Name	Label	Data Type	Operator	Compare To	Rule Name	
Screening	Screening					
Screening-Visit	Screening Visit					
DM	Demographics					
IG-DM	Demographics					
BRTHDAT	Birth Date	Date	cannot be after	Informed.Consent.Date	R_DCOMP_BRTHDAT	
MH	General Medical History					
IG-MH	General Medical History					
MHSTDAT	Date of Onset	Date	cannot be after	Informed.Consent.Date	R_DCOMP_MHSTDAT	
MHENDAT	End Date	Date	cannot be before	MHSTDAT(Date.of.Onset)	R_DCOMP_MHENDAT	
PELL	Physical Exam					
IG-PELL	Physical Exam					
PEDAT	Exam Date	Date	cannot be before	Informed.Consent.Date	R_DCOMP_PEDAT_2	
Cohort_A	Cohort A					
E_Visit1	Visit 1					
PELL	Physical Exam					
IG-PELL	Physical Exam					

The comparison configurator not only makes it easy to specify individual date comparisons, it also functions as a to-do list, showing the data manager the complete list of date fields across the entire study. The list becomes a checklist of all the dates the study designer needs to consider. And in each row, the designer either specifies the comparison required or indicates that an edit check isn't needed.

In addition to saving build time, configurable date checks eliminate the need to test these checks during UAT. Since the system is validated to function as designed, individual testing isn't needed. Instead, date comparisons should be visually checked to confirm they were configured properly.

The comparison tool is a critical part of Vault EDC supporting the notion of fusing specifications and design, transition from building a database to configuring a casebook. Over time, the configurator will support numeric comparisons, data points over time, and more.

In Conclusion

The trend toward increasingly complex trial designs has placed a burden on clinical programmers to supplement the functionality of traditional EDC systems with extensive custom functions.

The rules engine in Vault CDMS equips study builders with a deep reservoir of capabilities, enabling them to produce advanced dynamics and edit checks directly within Vault EDC. The rules engine was also designed to enable data managers to write edit checks and dynamics independently, in lieu of writing technical specs for the rules. Numerous capabilities such as drag-and-drop rule-building, casebook variables, and auto-suggest, lower the technical barrier to writing rules, enabling data managers to build all but the most complicated rules themselves. A clinical trial data manager at Alcon noted, *"In the time it takes to write and review a spec we can build the database by ourselves."* She went on to say, *"I highly recommend that data managers be empowered to do more than cleaning data. They should be enabled to build databases themselves in Veeva."*

Companies looking for ways to speed clinical trials are adopting Vault CDMS to benefit from the faster build times and greater agility during study conduct. The powerful rules engine in Vault CDMS delivers the automation and flexibility that organizations need to support today's complex trials.

FIGURE 3. SIX CAPABILITIES THAT MAKE RULES EASIER AND SMARTER



Casebook Variables

Reference commonly used dates such as birth date or informed consent date by using the correlated casebook variables, which are shorter and easier to access when writing rule expressions or specifying date comparisons.



Cross-Form Queries

You can create data validation rules across multiple items and across forms. The point-and-click interface in the rules editor makes it easy to locate and select fields from any other form in the casebook.

243
896

Sequence Number References

You can reference the sequence number of a repeating event group or form to perform the intended action only on specific instances of that group or form. For example, certain forms may only be added to the first visit in a treatment cycle, or certain visits may be only added to the first treatment cycle.



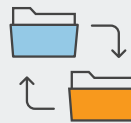
Wildcard Identifiers (@Form)

Vault will automatically evaluate a rule each time a form with a wildcard identifier is submitted. Vault will also re-evaluate the rule each time data changes on any form referenced in the rule. For example, if @Form points to Vital Signs form that is used in multiple visits, Vault evaluates the rule on each occurrence of the Vitals form.



Copying Rules

Copy rule definitions and rule actions from the current study and from other studies in your Vault. If the items referenced in a rule don't exist in your new study, Vault copies the rule and prompts you to update the identifiers before activating the rule in the new study. Copying forms also allows you to copy rules associated with those forms.



Create Queries for Specified Casebook Versions

You can configure a data validation rule to only evaluate subject casebooks on a certain version. This is useful when you've updated your casebook such that a rule is no longer valid based on the new design.

FIGURE 4. ACTIONS WITHIN THE SCRIPT EDITOR TO PERFORM EDIT CHECKS, DYNAMICS, CALCULATIONS, AND MORE.

Action Type	Result
Query	<p>Creates a query on the selected item or event date with the entered query message.</p> <p>EXAMPLE: Generate a query if the diastolic blood pressure is greater than the systolic blood pressure.</p>
Add Event Group	<p>Adds the selected event group to the casebook schedule.</p> <p>EXAMPLE: Dynamically add a new event group comprised of the appropriate visits and forms for the next treatment cycle.</p>
Add Event and/or Form	<p>Adds the selected event and/or form to the casebook schedule.</p> <p>EXAMPLE: Dynamically add the follow up visit and related forms when a user selects the <i>Follow Up Visit Required</i> checkbox on a form.</p>
Disable Item	<p>Disables the designated item(s) based on the value of controlling items.</p> <p>EXAMPLE: Disable the pregnancy test result field based on the value for the subject's gender.</p>
Set Item Value	<p>Populates the item with the calculated result of the expression.</p> <p>EXAMPLE: Calculate the Body Mass Index (BMI) for a subject based on the entered height and weight values and populate the BMI field with the result.</p>
Set Subject Status	<p>Updates the subject's status to the selected status and records the status change date.</p> <p>EXAMPLE: Automatically set a subject's status as <i>In Screening</i> once a date has been entered in the screening event.</p>
Send Email	<p>Sends an email, using the configured subject and message, to a pre-configured recipient group.</p> <p>EXAMPLE: Automatically send a notification to medical monitors in your study when a new adverse event is added to a casebook.</p>
Progressive Display	<p>Shows or enables items based on the value entered earlier on the form.</p> <p>EXAMPLE: An adverse event form can dynamically display the End Date field only after a user enters "no" to the question is the adverse event ongoing.</p>
Override Review Plan	<p>Overrides the default review plan for the form with the chosen review plan.</p> <p>EXAMPLE: Override the default SDV plan for an adverse event if the event is marked as "serious."</p>