

New Features Available in REDCap 11.1.15

Table of Contents

New feature: SendGrid Email API Integration	2
New feature: Select and modify multiple fields together on the Online Designer	2
New feature: Integration of Paul Litwin's Stealth Queue external module.....	2
New feature: Added "Language of text to be spoken" for the "Text-To-Speech" survey functionality.....	2
New feature: Users may re-evaluate some or all Automated Survey Invitations for all records in a project.....	2
New feature: Users may re-evaluate some or all Alerts & Notifications for all records in a project.	2
New action tag: @CALCDATE.....	3
New action tag: @CALCTEXT	3
New feature: Data Access Group import/export and DAG-User assignment import/export	3
New feature: Data Quality Rule import/export.....	3
New action tag: @PREFILL	3
New special functions: left(), right(), mid(), length(), find(), trim(), upper(), lower(), and concat().....	4
New feature: New logic editor for conditional logic, branching logic, calculations, report filters, etc.	4
New feature: Auto-numbering of repeating instances for data imports.....	4
New feature: New survey option "Save a PDF of completed survey response to a File Upload field"	5
New feature: Field Bank.....	5
New feature: @INLINE action tag.....	5
New feature: New "inline" piping option for File Upload fields.....	5
New feature: Instant Adjudication for Clinical Data Pull (CDP) with improved CDP Field Mapping page	5
New feature: New API "Export Logging" method.....	6
New feature: Ability to to import/export user rights via a CSV file on the User Rights page	6
New feature: Export Data Quality rule results.....	6
New feature: Field that maps to a participant's Twilio delivery preference	6
New feature: Custom offline message for surveys in offline status.....	6
New feature: Survey-level Stop Action controls (new section on Survey Settings page).....	6
New feature: Project Dashboards.....	7
New feature: Smart Functions	8
New feature: Smart Tables	8
New feature: Smart Charts	9
Optional parameters for Smart Functions, Smart Tables, and Smart Charts	9
New feature: CSV Delimiter as a user-level preference	11
New feature: More clinical data available via FHIR R4 endpoints for CDIS.....	11
New feature: Fields that are "sql" field type (Dynamic Query - SQL field) now work in the REDCap Mobile App.	11
New feature: Import/export alerts via CSV file on Alerts & Notifications page	11

New feature: SendGrid Email API Integration

- As an alternative for sending outgoing emails from REDCap (rather than using the standard settings in PHP.INI to send them natively from the web server), you may use SendGrid, which is a third-party paid service (owned by Twilio) that can send emails on behalf of REDCap.
- The option can be configured on the General Configuration page in the Control Center. You merely have to provide the API key for your SendGrid account, and it will begin using the SendGrid Web API to send *all* emails going out of REDCap.

New feature: Select and modify multiple fields together on the Online Designer

Users may select multiple fields on the Online Designer by holding the Ctrl, Shift, or Cmd key on their keyboard while clicking on the field in the table, which will reveal the options to Move, Copy, or Delete all the selected fields. To make users aware of this feature, a floating note now appears near the right side of the page in the Online Designer with instructions on how to use this.

New feature: Integration of Paul Litwin's Stealth Queue external module

- New "Keep the Survey Queue hidden from participants?" setting in the "Set up Survey Queue" dialog on the Online Designer
- This setting will keep the Survey Queue table hidden from participants, and will force Auto Start to be enabled for all queue-activated surveys. This is useful if users wish to use the Survey Queue to automatically guide survey participants to the next survey without displaying the queue of surveys.

New feature: Added "Language of text to be spoken" for the "Text-To-Speech" survey functionality

Available on the Survey Settings page. For several years, REDCap has had a Text-to-Speech feature for surveys that, when enabled, allows questions and other text on survey pages to be converted into natural-sounding audio for the participant to hear. Up until now, it supported English only, but now REDCap users may utilize the Text-to-Speech feature in a variety of non-English languages and voices, assuming that the survey text is in a non-English language. This includes Arabic, Brazilian Portuguese, English (UK and US), French, German, Italian, Japanese, and Spanish (Castilian, Latin American, and North American).

New feature: Users may re-evaluate some or all Automated Survey Invitations for all records in a project.

If an ASI has been modified after data has already been entered in the project, users may click the "Re-evaluate Auto Invitations" button in the Online Designer, which will re-evaluate selected ASIs for all records to ensure that invitations get properly sent or scheduled based on the new conditions of the ASI (otherwise they could only be triggered if each individual record had data modified). If a user modifies the conditional logic of an ASI, it will recommend that they utilize the "Re-evaluate Auto Invitations" functionality. If an ASI has the "Ensure logic is still true..." option checked, then it is possible during this process that some already-scheduled invitations might get removed (and thus would no longer be scheduled) based on the new conditions.

New feature: Users may re-evaluate some or all Alerts & Notifications for all records in a project.

If an alert has been modified after data has already been entered in the project, users may click the "Re-evaluate Alerts" button on the Alerts & Notifications page, which will re-evaluate selected alerts for all records to ensure that notifications get properly sent or scheduled based on the new conditions of the alert (otherwise they could only be triggered if each individual record had data modified). If a user modifies the conditional logic of an alert, it will recommend that they utilize the "Re-evaluate Alerts" functionality. If an alert has the "Ensure logic is still true..." option

checked, then it is possible during this process that some already-scheduled notifications might get removed (and thus would no longer be scheduled) based on the new conditions.

New action tag: @CALCDATE

Performs a date calculation by adding or subtracting a specified amount of time from a specified date or datetime field and then provides the result as a date or datetime value - e.g., @CALCDATE([visit_date], 7, 'd'). The first parameter inside the @CALCDATE() function should be a text field with date, datetime, or datetime_seconds validation, in which you may specify (if needed) the event and repeating instance - e.g., @CALCDATE([baseline_event][visit_date], 7, 'd'). The second parameter represents the offset number amount that should be added or subtracted. It can be a decimal number or integer. Tip: To subtract (i.e., go backwards in time), use a negative number. The third parameter represents the units of the offset amount, which will be represented by the following options: 'y' (years, 1 year = 365.2425 days), 'M' (months, 1 month = 30.44 days), 'd' (days), 'h' (hours), 'm' (minutes), 's' (seconds). The unit option must be wrapped in quotes or apostrophes. NOTE: Both the source field and the result field must be a text field with date, datetime, or datetime_seconds validation. It is important to realize that a field with @CALCDATE will not be editable on the survey page or data entry form, and the field will function almost exactly like a normal calculated field, in which its value may get updated via a data import, when running Data Quality rule H, or in real-time during normal data entry on a form or survey.

New action tag: @CALCTEXT

Evaluates logic that is provided inside a @CALCTEXT() function and outputs the result as text, typically performed with an if(x,y,z) function - e.g., @CALCTEXT(if([gender]='1', 'male', 'female')). NOTE: It is important to realize that a field with @CALCTEXT will not be editable on the survey page or data entry form, and the field will function almost exactly like a normal calculated field, in which its value may get updated via a data import, when running Data Quality rule H, or in real-time during normal data entry on a form or survey. If desired, it is possible to return the value as a number - e.g., @CALCTEXT(if([age] >= 18, 'adult', 5*[other_field])).

New feature: Data Access Group import/export and DAG-User assignment import/export

The Data Access Groups page in a project now displays a drop-down list of options for users to import/export Data Access Groups, which allows users to bulk create or rename DAGs via a CSV file. It also allows for the import/export of DAG-user assignments via CSV file to bulk assign/reassign/unassign users from DAGs in a project. Note: The DAG-user assignment import affects only a user's *current* DAG assignment; thus, it has no effect on the DAG Switcher assignments for the user.

New feature: Data Quality Rule import/export

The Data Quality page in a project now displays a drop-down list of options for users to import/export custom Data Quality rules via a CSV file. Note: This does not apply to the pre-defined DQ rules (rules A-I). Also, when DQ rules are imported, the process is additive only, meaning that the CSV upload cannot replace or edit existing DQ rules but will only add new ones to the project.

New action tag: @PREFILL

Sets a field's value to static text or dynamic/piped text whenever a data entry form or survey page is loaded, in which it will always overwrite an existing value of the field. The format must follow the pattern @PREFILL="????", in which the desired value should be inside single or double quotes. A field with @PREFILL will always be read-only, thus its value cannot be modified manually on the data entry form or survey page. For text fields, you may pipe and concatenate values from other fields in the project - e.g., @PREFILL='Name: [first_name] [last_name], DOB: [dob]'. For checkbox fields, simply separate multiple checkbox values with commas - e.g., @PREFILL='1,3,[other_field:value]'. NOTE: The piped value does *not* get applied during any data imports (via API or Data Import Tool) but only operates when viewing survey pages and data entry forms. NOTE: A field with @PREFILL will have its value updated ONLY when the page loads, which means that its value will not be updated in real-time if you modify other fields on the same page that are piped into the @PREFILL tag. NOTE: If being used on a date or datetime field, the date value inside the quotes must be in Y-M-D format - e.g., @PREFILL='2007-12-25' - regardless of the field's set date format. NOTE: The only difference

between @PREFILL and @DEFAULT is that @DEFAULT is only applied when an instrument has no data yet, whereas @PREFILL will always be applied on an instrument, meaning that @PREFILL will ALWAYS overwrite the value if a field value already exists. TIP: To pipe the value of one multiple choice field into another multiple choice field, make sure you append ':value' to the variable being piped - e.g., @PREFILL='[my_dropdown:value]'.

[New special functions: left\(\), right\(\), mid\(\), length\(\), find\(\), trim\(\), upper\(\), lower\(\), and concat\(\).](#)

These nine new functions can be specifically used when dealing with text values and may be especially useful when using them in conjunction with the @CALCTEXT action tag. To learn more and to see some practical examples of their usage, click the blue 'Special Functions' button in the Online Designer in any project.

- **left** (*text, number of characters*) - Returns the leftmost characters from a text value. For example, left([last_name], 3) would return 'Tay' if the value of [last_name] is 'Taylor'.
- **right** (*text, number of characters*) - Returns the rightmost characters from a text value. For example, right([last_name], 4) would return 'ylor' if the value of [last_name] is 'Taylor'.
- **length** (*text*) - Returns the number of characters in a text string. For example, length([last_name]) would return '6' if the value of [last_name] is 'Taylor'.
- **find** (*needle, haystack*) - Finds one text value within another. Is case insensitive. The "needle" may be one or more characters long. For example, find('y', [last_name]) would return '3' if the value of [last_name] is 'Taylor'. The value '0' will be returned if "needle" is not found within "haystack".
- **mid** (*text, start position, number of characters*) - Returns a specific number of characters from a text string starting at the position you specify. The second parameter denotes the starting position, in which the beginning of the text value would be '1'. The third parameter represents how many characters to return. For example, mid([last_name], 2, 3) would return 'AYL' if the value of [last_name] is 'TAYLOR'.
- **concat** (*text, text, ...*) - Combines/concatenates the text from multiple text strings into a single text value. For example, concat([first_name], ' ', [last_name]) would return something like 'Rob Taylor'. Each item inside the function must be separated by commas. Each item might be static text (wrapped in single quotes or double quotes), a field variable, or a Smart Variable.
- **upper** (*text*) - Converts text to uppercase. For example, upper('John Doe') will return 'JOHN DOE'.
- **lower** (*text*) - Converts text to lowercase. For example, lower('John Doe') will return 'john doe'.
- **trim** (*text*) - Removes any spaces from both the beginning and end of a text value. For example, trim(' Sentence with spaces on end. ') will return 'Sentence with spaces on end.'.

[New feature: New logic editor for conditional logic, branching logic, calculations, report filters, etc.](#)

In every place where users might add/edit logic or calculations, the new logic editor will be displayed in a modal dialog to provide a better user experience for entering their logic. The logic editor provides much more space for entering large amounts of logic, including a fullscreen mode to take maximum advantage of their screen's real estate. It also provides bracket-matching and parentheses-matching where it will highlight a pair of matching brackets/parentheses to make it easier for users to gauge which brackets/parentheses belong together in the logic, thus reducing possible errors in the logic when typing.

[New feature: Auto-numbering of repeating instances for data imports](#)

When using repeating events or repeating instruments, it may be difficult when performing dynamic imports of data for these because it is not easily known how many repeating instances already exist in a project for a given repeating event/instrument, thus often forcing users to invent clever ways to determine this, such as performing data exports beforehand and then dynamically determining what the next repeating instance number should be. However, that is no longer necessary. When performing a data import now for a repeating event/instrument, users may use the literal value "new" as the value for the "redcap_repeat_instance" field in their data import. By doing so, REDCap will perform the instance auto-numbering on its own to increment the repeating instances properly based on the highest numbered instance that already exists in the saved data in the project.

New feature: [New survey option “Save a PDF of completed survey response to a File Upload field”](#)

On the Survey Settings page in the Online Designer, users may select a File Upload field in the project where a static PDF file of a participant’s survey response will be stored immediately after they complete the survey. For longitudinal projects, if the target field exists on multiple events, users may set this feature so that it stores the PDF in the selected field in the current event (default) or else in a specific event in the project. Thanks to Philip Chase and his team at University of Florida for their inspiration for this feature, in which it was based on their “Save Survey PDF to Field” external module. NOTE: Upgrading to REDCap 10.6.0 will *not* automatically disable the “Save Survey PDF to Field” module if it is installed and enabled on any projects, nor will it transfer the saved settings of that module into this new feature in REDCap.

New feature: [Field Bank](#)

When adding new fields via the Online Designer, users will see an "Import from Field Bank" button, which will allow them to search different standardized catalogs of commonly used fields, such as in the U.S. National Library of Medicine catalog. The Field Bank helps users add new fields quickly and easily to their data collection instruments. Over time, more standardized catalogs of fields will be added to the Field Bank.

New feature: [@INLINE action tag](#)

Allows a PDF file or image file (JPG, JPEG, GIF, PNG, TIF, BMP) that is uploaded to a File Upload field to be displayed in an inline manner on the survey page or data entry form so that the PDF/image can be viewed by the user or survey participant without having to download it.

- The PDF/image will be displayed inline on the page immediately above the download link for the field and will be displayed with 100% width by default (i.e., 100% width of the area in which it is contained).
- Images will be displayed with their native width:height ratio, although PDFs will be displayed with a 300 pixel height by default. If you wish to manually set the width and/or height of the image/PDF, you may put the width/height values inside parentheses after the action tag in the following manner: @INLINE(width) or @INLINE(width,height). The width/height can be a percentage value (e.g., 50%) or a number representing size in pixels (e.g., 400). Thus @INLINE(50%) will display an image at 50% size for the area in which it is contained on the page, and @INLINE(400,100) would display the image always at 400px tall and 100px wide. To make an inline PDF appear taller on the page, you might use @INLINE(100%,600) since 300px is the default height for inline PDFs.
- The @INLINE action tag also works if the File Upload field is embedded inside another field on the page.
- Thanks to Andy Martin for his inspiration for this feature, in which it is based on his “Image Viewer” external module. NOTE: Upgrading to REDCap 10.7.0 will *not* automatically disable the “Image Viewer” module if it is installed and enabled on any projects, nor will it conflict with the “Image Viewer” external module.

New feature: [New “:inline” piping option for File Upload fields](#)

- If piping using the ':inline' option for a File Upload field, such as [my_field:inline], in which the uploaded file is a PDF file or image file (JPG, JPEG, GIF, PNG, TIF, BMP), the file will be displayed in an inline manner so that it is viewable on the page.
- The ':inline' option DOES work inside emails, so you can pipe a field with ':inline' inside the email body, thus allowing you to display inline images inside survey invitations or Alerts & Notifications.
- The @INLINE action tag does not need to be used on a field in order to utilize the “:inline” piping option.
- Note: Inline images are not able to be displayed inside a downloaded PDF of a survey/instrument that contains data.

New feature: [Instant Adjudication for Clinical Data Pull \(CDP\) with improved CDP Field Mapping page](#)

- Once enabled here for the whole system via the CDIS page, the Clinical Data Pull "Instant Adjudication" setting can be enabled on a CDP project's field mapping page, after which it will allow users to bypass the normal data adjudication process and will let them import and save all data into the project that has already been cached

from the EHR system. This can save a great deal of time when importing lots of patient records. After Instant Adjudication is enabled in a CDP project, users with CDP-adjudication privileges will see the button to initiate this process on the Record Status Dashboard. After the button is clicked, it will begin adjudicating the EHR data for all records in real time, thus saving the data into records in the projects. On the CDIS page in the Control Center, administrators can enable or disable the Instant Adjudication feature for all CDP-enabled projects in the system. By default, the system-level Instant Adjudication option is enabled.

- Also, the user interface for the CDP Field Mapping page in all CDP-enabled projects has been updated and improved to allow users to more quickly and easily map their REDCap fields to EHR fields for their CDP project.

[New feature: New API “Export Logging” method](#)

This new API method allows users to export a project’s logging via the API using very similar methods and filters as in the project’s user interface. See the documentation for all filter parameters that are available.

[New feature: Ability to to import/export user rights via a CSV file on the User Rights page](#)

Users can download a CSV file to view all the user privileges of the existing users in a project, including their instrument-level user rights. Users can upload a CSV file to grant new users access to the project and/or to modify the user privileges of existing users, including their instrument-level user rights.

[New feature: Export Data Quality rule results](#)

After running a data quality rule, users may export the results/discrepancies of the rule as a CSV file. The CSV file will be structured exactly like a date export/import file, which should allow for faster and easier cleaning of data so that values can be fixed and then re-uploaded as a data import.

[New feature: Field that maps to a participant’s Twilio delivery preference](#)

When using Twilio for surveys, users can control each participant's invitation preference automatically using a multiple choice field. If survey participants require using different methods (e.g., email, SMS w/ link, voice call survey) for receiving survey invitations and/or taking surveys, users can select a multiple choice field whose choices represent each survey invitation delivery method. After mapping the invitation preferences to a field, whenever the value of the field is added or modified, the participant's invitation preference will automatically be changed accordingly. **IMPORTANT:** The multiple choice codings for the selected field must be defined exactly as delineated below, although their corresponding choice labels can be modified to be whatever text the user desires. Also be aware that if the value of the field that is mapped is set to blank/null, then the invitation preference for the participant will revert to the project's default invitation preference (as defined in the Twilio configuration on Project Setup). Additionally, if a participant's invitation preference is modified via the Participant List, that change will also change the value of the mapping field selected above. Mapped field choice options:

- EMAIL, Email invitation
- SMS_INVITE_WEB, SMS invitation (contains survey link)
- SMS_INITIATE, SMS invitation (take survey via SMS)
- VOICE_INITIATE, Voice call (participant receives voice call)
- SMS_INVITE_MAKE_CALL, SMS invitation (contains phone number to call)
- SMS_INVITE_RECEIVE_CALL, SMS invitation (reply via SMS to receive voice call)

[New feature: Custom offline message for surveys in offline status](#)

Users can provide custom text that is displayed to participants only when the survey is offline. This custom text will be displayed in place of the default offline text on the survey while the survey is in offline mode. This text can be set at the top of the Survey Settings page.

[New feature: Survey-level Stop Action controls \(new section on Survey Settings page\)](#)

- **Alternative survey completion text** - Users can optionally set alternative survey completion text that is displayed in place of their standard survey completion text whenever a survey is ended via a Stop Action on any

field. This is useful when it doesn't make sense for non-eligible participants to see the same survey completion text as those who completed the survey fully.

- **Prevent survey responses from being saved if the survey ends via Stop Action** - Users can optionally choose to prevent submitted responses from being saved as data in the project if the survey ends via Stop Action. This is useful if survey administrators do not wish to keep the data for ineligible participants, for example. This means that if a one-page public survey is started but ends via Stop Action, no data from that response will be saved into the project (i.e., no new record will be created), but it will log this event on the project Logging page (so that users are at least aware of this happening despite no data being saved).
 1. **NOTE:** If any data has been saved on the survey instrument for a given record prior to the Stop Action being triggered, that data will be deleted from that instrument. For example, if the survey is a multi-page survey in which data has been entered on previous pages prior to triggering the Stop Action, all data collected thus far in that survey will be deleted as if the survey was never taken. Additionally, if the record does not contain data in any other instruments, the entire record itself will be deleted during this process. If data does exist in other instruments, the record will not be deleted.
 2. **PRIVACY NOTE:** If the option for Data Privacy/GDPR has been enabled in the project, in which it removes the contents of the log for a record that is deleted from the project, then if an entire record is deleted via this particular survey setting via a Stop Action, then all logged data values for the record will be removed from the log as per this project's data privacy setting.

New feature: Project Dashboards

- **INTRO:** Project Dashboards are pages with dynamic content that can be added to a project. They can utilize special Smart Variables called Smart Functions, Smart Tables, and Smart Charts (described below) that can perform aggregate mathematical functions, display tables of descriptive statistics, and render various types of charts, respectively. User access privileges are customizable for each dashboard, and anyone with Project Design privileges can create and edit them. A Wizard is provided on the Project Dashboard creation page to help users easily construct the syntax for Smart Functions, Smart Tables, or Smart Charts, and a basic list of helpful examples is also included. Example dashboard: <https://redcap.link/dash1>
- **Setting project dashboards as “public”**
 1. If enabled at the system-level (described in detail below), any project dashboard can be enabled as “public”, which means it can be accessed at a unique URL that does not require any authentication. Making a dashboard public is useful if you wish for people to view it without having to be REDCap users or log into REDCap. Public dashboards are simply standalone pages that can be viewed by anyone with a link to them.
 2. Users can opt to create a custom/short url (via the <https://redcap.link> service) for any project dashboard that is enabled as “public”.
 3. **System-level setting to allow/disallow public dashboards** (on the User Settings page in the Control Center) - By default, normal users will be able to set any project dashboard as public. If you do not want users to do this or even know about this feature, you can completely disable it on the User Settings page. Alternatively, it can be set to “Allow public dashboards with admin approval only”. If set to allow public dashboards after approval by an admin, the admin will receive the request from the user via the To-Do List page (and via email, if the email notification setting is enabled on the To-Do List page), and after the admin approves the request, the user will receive an email regarding the response to their request.
- **Setting to control data privacy on public dashboards and other public pages**
 1. The User Settings page in the Control Center has a setting to define the “Minimum number of data points required to display data for any Smart Charts, Smart Tables, and Smart Functions on a *public* project dashboard, survey queue, or survey page”. **By default, it is set to a value of “11”.** While only aggregate data is displayed in Smart Charts, Smart Tables, and Smart Functions, if any of these utilize

very few data values, it might pose a threat to an individual's data privacy if these are being displayed on *public* dashboards and other public pages (i.e., where authentication is not used).

2. If someone is viewing a public page that has Smart Charts, Smart Tables, and Smart Functions that utilize data that does not meet the minimum data point requirement, instead of displaying the chart/table/number on the page, it will instead display a notice saying “[INSUFFICIENT AMOUNT OF DATA FOR DISPLAY]” with a pop-up note with details about the minimum data requirements.
 3. **Project-level override:** While this behavior is controlled by a system-level setting, the system-level setting can be modified by an administrator via a project-level override for any given project on the “Edit A Project's Settings” page.
 4. Note: This setting does not get used when viewing project dashboards inside a project (i.e., at a non-public URL).
- **PDF export:** Each project dashboard can be exported as a one-page PDF file.
 - **Dashboard cache:** To prevent server performance degradation, each project dashboard will have its content cached (stored temporarily) automatically for up to 10 minutes at a time rather than generating its content in real time every time the dashboard is loaded. It will note at the top right corner of the dashboard page when the dashboard content was last cached. If a user is viewing the dashboard inside a project (i.e., not via a public dashboard link), they have the option at the top right to “Refresh” the dashboard at will, which will refresh/generate its content in real time. Note: The refresh option will only be displayed on the page when the dashboard content is at least 30-seconds old.

New feature: Smart Functions

- Smart Functions are aggregate mathematical functions that are utilized as Smart Variables. The following Smart Functions exist: [aggregate-min], [aggregate-max], [aggregate-mean], [aggregate-median], [aggregate-sum], [aggregate-count], [aggregate-stdev], and [aggregate-unique]. Each represents the mathematical functions minimum, maximum, mean/average, media, sum, count, standard deviation, and unique count, respectively. Each must have at least one field attached to it that follows a colon - e.g., [aggregate-mean:age]. Multiple fields may be used in each one, which will perform the function over all the data values of all the fields. By default, the functions will utilize all data values for all records in the project. To limit the data values being utilized to a subset of the total project data, see the Smart Variable documentation on how to apply filters, such as attached unique report names, DAGs, and other parameters
- Note: When using [aggregate-count:record_id], in which “record_id” in this example represents whatever the variable of the Record ID field is, it performs a special count that does not literally count the number of data values but instead returns a count of the total number of records in the project. This is a quick way to display the total record count of the project.
- Smart Functions can be used anywhere in a project where piping is allowed, and can even be used inside calculations, branching logic, and other conditional logic (report filters, alert conditions, etc.).

New feature: Smart Tables

- Smart Tables are tables displaying aggregate descriptive statistics in which the results of any or all of the following stats functions can be displayed for one or more fields: minimum, maximum, mean/average, media, sum, count, standard deviation, count of missing values, and count of unique values.
- Smart Tables are represented with the Smart Variable [stats-table], which accepts as a parameter the variable names (comma delimited) of all the fields to be displayed as separate rows in the table. There is no limit to the number of fields that can be used. For example, [stats-table:field1,field2,field3].
- By default, all available columns will be displayed in the table and are as follows: Count, Missing, Unique, Min, Max, Mean, Median, StDev, Sum. To display only a subset of the columns, you may provide any of the following designations (comma-separated) that represent a specific column in the table: count, missing, unique, min, max, mean, median, stdev, sum. For example, [stats-table:field1,field2,field3:mean,max].

- By default, each stats table will have an "Export table (CSV)" link displayed immediately below it to allow users to download the table as a CSV file. But if users wish to hide the export link, they can simply attach “:no-export-link” to the Smart Variable, which will cause the link not to be displayed. For example, [stats-table:field1,field2,field3:no-export-link].
- Smart Tables can be used anywhere in a project where piping is allowed.

New feature: Smart Charts

- Smart Charts are various aggregate plots and charts utilized as different Smart Variables. The following plots are available for use: bar charts, pie charts, donut charts, scatter plots, and line charts. These are all represented by the following Smart Variables, respectively: [bar-chart], [pie-chart], [donut-chart], [scatter-plot], and [line-chart]. These Smart Variables accept one or more field names and also other optional parameters, as described below for each.
- **Bar charts** - Displays a bar chart for a single multiple choice field. It can optionally perform color grouping if a second field (multiple choice only) is provided. The fields must be comma-separated. For example, [bar-chart:field,grouping-field:parameters]. Bar charts have optional parameters that can be applied to alter their appearance. By appending the parameter “:bar-stacked” when two fields are used, the bars in the chart will appear stacked on top of each other rather than side by side. By default, bar charts are displayed with their bars going horizontally, but by appending the parameter “:bar-vertical”, the orientation will be changed to display vertically instead.
- **Pie charts** - Displays a pie chart for a single multiple choice field. For example, [pie-chart:field:parameters].
- **Donut charts** - Displays a donut chart for a single multiple choice field. Note: A donut chart is essentially the same as a pie chart but with the center removed. For example, [donut-chart:field:parameters].
- **Scatter plots** - Displays a scatter plot of one number/date/datetime field for the x-axis and a second field (number field only) for the y-axis. (If a second field is not provided, a random value will be assigned for the y-axis.) It can optionally perform color grouping if a third field (multiple choice only) is provided. All fields must be comma-separated. For example, [scatter-plot:x-axis-field,y-axis-field,grouping-field:parameters].
- **Line charts** - Displays a line chart of one number/date/datetime field for the x-axis and a second field (number field only) for the y-axis. It can optionally perform color grouping if a third field (multiple choice only) is provided. All fields must be comma-separated. Note: A line chart is essentially the same as a scatter plot except with dots connected with a line. For example, [line-chart:x-axis-field,y-axis-field,grouping-field:parameters].
- **Color blindness accessibility:** Pie charts and donut charts have the ability for the user to enable color blindness accessibility, via a gray link displayed immediately below each chart, in which it overlays different patterns onto the colored pieces of the chart to make each color more distinct for many types of color blindness. This option to enable color blindness accessibility is stored in a secure cookie on the user’s device and will be used to remember this choice anytime a pie/donut chart is displayed on any page for any REDCap project for that REDCap server.
- The colors displayed in each chart/plot are preset and are not modifiable.
- Smart Charts can be used anywhere in a project where piping is allowed *except* for inside the body of outgoing emails.

Optional parameters for Smart Functions, Smart Tables, and Smart Charts

- There exist various optional parameters that can be used with Smart Functions, Smart Tables, and Smart Charts to either filter the data used in them (e.g., via a unique report name) or to change their appearance (e.g., bar-vertical). See the descriptions for each below, which are all documented in the Smart Variables documentation.
- **:R-XXXXXXXXXX Unique Report Name** - For Aggregate Functions, Charts, and Tables, filter the data being used by appending a Unique Report Name. Next to each report on the 'My Reports & Exports' page is its unique report name, which has 'R-' following by alphanumeric characters. By default, all Aggregate Functions, Charts, and Tables will use the values of all records in the project, but if a unique report name is appended to any of

them, only data from that specific report will be used. Using a report as a surrogate to filter data is a very useful technique of performing complex filtering logic for Aggregate Functions, Charts, and Tables.

- **:record-name "record-name"** - For Aggregate Functions, Charts, and Tables, filter the data being used to the *current record* by using the literal value 'record-name'. Note: This parameter will only work in a context where a single record is being viewed/accessed, such as on a survey page, data entry form, etc. This parameter can be used with any of the other parameters except unique report names.
- **:event-name "event-name"** - For Aggregate Functions, Charts, and Tables, filter the data being used to the *current event* (longitudinal projects only) by using the literal value 'event-name'. Note: This parameter will only work in a context where a single record/event is being viewed/accessed, such as on a survey page, data entry form, etc. This parameter can be used with any of the other parameters except unique report names.
- **:unique-event-names Unique Event Names** - For Aggregate Functions, Charts, and Tables, filter the data being used to specific events (longitudinal projects only) by providing an event's unique event name (found on the Define My Events page). You may use one or more unique event names (comma-separated). Note: This parameter can be used with any of the other parameters except unique report names.
- **:user-dag-name "user-dag-name"** - For Aggregate Functions, Charts, and Tables, filter the data being used to the records assigned to the *current user's Data Access Group* by using the literal value 'user-dag-name'. Note: This parameter will only work in a context where an authenticated user belongs to a project and has been assigned to a DAG in the project (this excludes survey pages and public project dashboards). This parameter can be used with any of the other parameters except unique report names.
- **:unique-dag-names Unique DAG Names** - For Aggregate Functions, Charts, and Tables, filter the data being used to the records assigned to specific Data Access Groups by providing a DAG's unique group name (found on the Data Access Groups page). You may use one or more unique DAG names (comma-separated). Note: This parameter can be used with any of the other parameters except unique report names.
- **:bar-vertical "bar-vertical"** - Display a bar chart with the bars going vertically instead of horizontally (the default) by using the literal value 'bar-vertical'. Note: This parameter can be used with any of the other parameters.
- **:bar-stacked "bar-stacked"** - Only for bar charts using two fields, display the bar chart with the bars stacked on top of one another for each choice. Whereas the default view is that the bars of each field are displayed side by side to show the color grouping. To enable this, use the literal value 'bar-stacked'. Note: This parameter can be used with any of the other parameters.
- **:no-export-link "bar-stacked"** - Only for bar charts using two fields, display the bar chart with the bars stacked on top of one another for each choice. Whereas the default view is that the bars of each field are displayed side by side to show the color grouping. To enable this, use the literal value 'bar-stacked'. Note: This parameter can be used with any of the other parameters.

NOTE: Using Smart Functions/Tables/Charts elsewhere in a project - While project dashboards are an excellent place to use Smart Functions, Smart Tables, and Smart Charts, it is important to know that Smart Functions/Tables/Charts can actually be used *almost anywhere* in a project, such as on data entry forms, on survey pages, and in report instructions (to name a few). You can use Smart Functions/Tables/Charts anywhere that piping can be used. Click the green "Smart Variables" button on the Project Setup page to learn more about them. Note: The only place that Smart Charts cannot be used is inside the body of outgoing emails.

NOTE: Smart Functions/Tables/Charts do not yet work in the REDCap Mobile App; however, it is planned that they eventually will (to a certain degree).

NOTE regarding permissions for Smart Functions/Tables/Charts:

- DAG permissions (i.e., filtering out records not assigned to the current user's DAG) are NOT applied by default to Smart Charts/Tables/Functions but are only applied when the Smart Chart/Table/Function utilizes a unique report name as a parameter (thus mimicking the natural DAG-filtering behavior of reports themselves) OR when the Smart Chart/Table/Function utilizes the "user-dag-name" parameter. This means that if a user is assigned to a DAG and views a project dashboard with the Smart Chart [scatter-plot:weight], for example, the plot will

display data for ALL records in the project and not just the user's DAG. To limit the plot to just data in the user's DAG, it could be changed to [scatter-plot:weight:user-dag-name] in this case.

- Smart Charts/Tables/Functions that utilize a unique report name as a parameter for data filtering purposes will still function and display normally even if the user does not have explicit access to view that specific report referenced as a parameter.

[New feature: CSV Delimiter as a user-level preference](#)

The My Profile page now has a new user preference to allow a user to set their own preferred CSV delimiter (e.g., comma, semi-colon) that will be used as the delimiter character in all CSV file downloads throughout REDCap, such as data dictionary import/export, event import/export, user rights import/export, etc. This setting is not used by data imports and exports because those already have a way to specify the CSV delimiter manually. The system-level default value for this user preference can be set on the User Settings page in the Control Center, in which all new users created afterward will have their user-level preference set with this system-level default value. To modify all existing users' preference after upgrading (if your users would not want a comma delimiter), it will require running an "update" query in the database, such as this: `UPDATE redcap_user_information SET `csv_delimiter` = ',';`

[New feature: More clinical data available via FHIR R4 endpoints for CDIS](#)

The CDIS services Clinical Data Pull and Clinical Data Mart can now utilize version 4 (called "R4") of the FHIR web services from their local EHR system. The new R4 endpoints include the existing data that could be pulled in earlier versions as well as the following: Adverse Events, Core Characteristics (Observation), Encounters, and Immunizations. Note that "Adverse events" are only available for "research" projects where an IRB number is specified, in which the project's IRB number corresponds to the "Study ID" value from the EHR interface for a particular study (which is often the same as the study's IRB number).

[New feature: Fields that are "sql" field type \(Dynamic Query - SQL field\) now work in the REDCap Mobile App.](#)

In previous versions, they were not functional at all in the mobile app. Now when a project is loaded into the mobile app, any "sql" fields will be converted into static drop-down fields in the app. If new choices get dynamically added to the sql field on the server afterward, the project will need to be loaded again in the mobile app to obtain those choices for the sql field. (Ticket #107409)

[New feature: Import/export alerts via CSV file on Alerts & Notifications page](#)

Users may export and import alerts to the same project or another project using a CSV file. If updating an existing alert, the unique alert ID must be included in the CSV file to identify the alert that the user wishes to modify. If the unique alert ID is left blank in the CSV file being uploaded, it is assumed that the user wishes to create a new alert.

[New feature: Reorder alerts on Alerts & Notifications page](#)

In the options menu for any given alert, a user can select an alert to be moved to another position on the Alerts & Notifications page. When this is done, it notifies the user that moving the alert will in most cases cause the alert numbers to be renumbered for many existing alerts (since they are numbered based on their order). However, their alert title and unique alert ID will not change during this process.

