A.D.

Consents in Raiser's Edge: Considerations and Shortcomings

Consents is a feature in The Raiser's Edge 7 and NXT that was designed for clients in the EU to help them be compliant with GDPR in 2018. It is intended to work together with Solicit Codes and acts as a place to store a history of when consent was obtained and what it was for. You do not have to be bound by the EU GDPR to use it. **I encourage all organizations to explore its use to enhance data protection practices and help prepare for future regulation.** There is extensive guidance from Blackbaud on what the Consents feature is, how you can set it up and use it day to day (link below). This document looks at specifically at the challenges of using this feature.

Considerations

Learn and plan before you build

It's easy to launch into setting up Consents without planning or understanding how it is intended to be used. A lot of consideration is needed as to how you can best reflect constituent preferences, how you can make processes user-friendly for staff and how you can fulfill your legal obligations. Think long term and as wide a picture as necessary. Read the Blackbaud guidance to grasp what Consents are for and how they were designed to be used.

With no one-size-fits-all solution you will need to determine the best setup to suit your organization and constituents. Some organizations used broad Solicit Codes with associated Consents and simple mapping, ie. "Do Not Mail", "Do Not Email". Others that offer constituents more specific preferences reflect this in Consent mapping. Their Solicit Codes are as granular as needed and they might make use of Attributes as well. ie. No Annual Report, One Appeal a Year, Thanks Only, Mail Consent–Appeals, Mail Consent–Thanks, Mail Consent–ALL, Email Consent–Appeals, Email Consent–Thanks, Email Consent–ALL, and so on.

Consents are *not* a filter

A common mistake is to assume that Consents can be used as a filter, or can be used instead of Solicit Codes. This is not what Consents are for–they fulfill a need under GDPR to store the *history of constituent preferences*. They are a repository of information, not a categorization tool. They are difficult to query on and cannot be exported (unless you're exporting from Query, which is not normally a good idea). Whilst frustrating from a database maintenance and cleanup perspective, this is deliberate: they are just a storage mechanism. You should use Solicit Codes and other filters when compiling mailing lists. **Do not** attempt to use Consents for this.

Shortcomings

Consent Source is not required

Under GDPR, the source of a consent is just as important as when and how it was made. The Consent Source field provides an editable lookup table to store this. However, Consent Source is not a required field and cannot made to be required (none of the Consent fields are exposed in Config). Instead, you have to rely on users to remember to complete it. If they enter a Consent

A.D.

without a Source, they cannot edit that Consent. Instead they have to create another one then have someone with Supervisor rights delete the erroneous Consent. It is logical that only super-users can delete Consents but until Source can be made a required field this clumsy problem will continue.

If you use any apps that push information to Raiser's Edge–such as a sync to an email platform–find out if that sync can include Consent Source. If it does not you will need to ensure the username the app uses to push information is intuitive so you can use this to determine where a Consent came from. For example, if you use NXT email and a constituent unsubscribes from email a Consent is automatically added to the record with no Consent Source but the username "NXT System User (Email)". This username is enough to infer where the consent came from.

Solicit Code dates are half baked

In NXT web view when you add a Solicit Code you can apply a date. This is a great idea and a step forward but as these dates are not in database view this can cause confusion. Having no time or user stamp on Solicit Codes can cause difficulty when trying to understand a record's history, especially when something goes wrong. For example, when a record shows a repeat opt-out request, has the record not been updated or is something wrong with mailing list criteria? If a Consent was added each time the opt-out was made that will provide some clue, but Solicit Code can be edited independently of Consents. You could look at the Solicit Code in web view to see if a date was added. Having to check in several places is a burden that database managers could do without.

Consent Channels cannot be inactivated

The table for Consent Channel is hard coded in RE so you cannot hide any values your organization does not use (those values are explained here). So long as your users know which ones to use this works fine however it is too easy for them to select the "ALL" option. This was designed to be a timesaver, automatically adding all applicable Solicit Codes rather than users having to add a Consent for each channel. However, an unintended consequence of this option is that it adds consents to the record for *all* combinations that have a Solicit Code mapped. So if you've diligently set up many mappings for different scenarios, all of these will be added to the record (ie. Email-Newsletters, Email-Appeals, Email-Invitations etc). What it should do is just add one consent for each Channel. The workaround is to map Solicit Codes for each Channel with the "ALL" category only and ensure other mappings have no Solicit Codes selected. Instead they are entered manually after saving the Consent. See separate resource *AD Consents in RE – Sample setup* for an example.

NXT email opt-outs only do half the job

If you use NXT email and a recipient unsubscribes, this comes through to the record as a global opt-out Consent for email. A Consent is automatically added and the email address is marked DNC. If you have set up a consent mapping for NXT email (see "Set consent rules for Email" here) the Solicit Codes will also be updated. However, Consent Source is not added and the "Requests no email" box is not checked. Depending on how you track emails you may also need to update Appeal response. Although NXT email feeds through nicely to database view (it is easy to create

A.D.

static queries of email responses, for example) it's a pain that there's still additional steps you have to complete to keep your records updated. As noted above, the lack of a Consent Source is a shortcoming that cannot be resolved with a workaround, but you can infer from the username "NXT System User (Email)" where the consent came from.

NXT email opt-outs are all or nothing

There is currently no granularity in NXT email so if you use it for various topics and the recipient wishes to opt out of *one* of these, there is no function to reflect that as a preference in Raiser's Edge. Once opted out, if by mistake or only for one type of email, the recipient has to opt themselves in again. You cannot do this for them. The workaround is to make sure your emails contain a link to a preferences webpage. That page on your website should host a secure form for gathering preferences (email topics, frequency) which you regularly retrieve and update in database view. It's another manual workaround that database managers could do without.

Compare this to NetCommunity which has much better options for recipient preferences and opt-outs. If you currently use BBNC or another email platform and are considering moving to NXT email, carefully examine first whether this all-or-nothing issue is going to be a hindrance. If you are using a third party email platform, weigh this up against the burden of syncing records between the two systems.

Blackbaud's current guidance (read "Opt-out link" → Tip box) hints that future releases of NXT will enable email recipients to manage their interests but it's anyone's guess when this will happen.

Consents don't get dupe merged in web view

It's fine to use duplicate merging in web view (in Tools → Data health), especially if you use NXT email, routinely attach items to records or use Solicit Code dates. However, Consents are only included in database view duplicate merging. As you know if you use web view duplicate merging, you have to go into database view to complete the merge anyway (doing so in web view only inactivates the duplicate, it doesn't delete it). The solution is to start a merge in web view (or use Data Health just to identify potential matches) then complete it in database view to ensure Consents are retained correctly.

Consent Statements are not a table

When adding a consent there are fields to paste in the opt-in statement that the donor was given when they made their consent, and a place for your privacy policy. If your statements are the same everywhere that a donor can submit one (your website, email marketing, all payment platforms etc) this is not an issue. However, if your donors see different statements depending on where they submitted it, and you want to be able to refer to that within RE, you have to copy-paste the statement text into the field when adding the consent. As of 2018, according to the Idea Bank, there were plans to make Statement a drop-down field in NXT web view. This did not happen.