**Validation Rules in the Upload Process**

LegCRM validates input data on multiple levels while giving considerable flexibility as to the quality and file format of data that you choose to import. LegCRM considers the following questions as it looks at data that is being pre-processed for upload.

The upload process can consider each question at two levels:

* At the column level (the columns in the input file)
* At the [individual data item level](http://wp-issues-crm.com/specific-fields-in-the-upload-process/) (the row items in each column).

**Does the input file record include enough identifying information to be saved as a new constituent?**

*Conceptual Rule*:

* You cannot upload a file without mapping at least some constituent fields. If data is missing for some rows for the mapped fields, however, that will not stop the creation of constituents. By contrast, when entering constituents from the form, at least one of name, email or a partial address needs to be supplied.

*Column Implementation*:

* You cannot move past the mapping stage without mapping some fields.
* You cannot run express if you map activity fields. When you do not run express, you have to go through the matching step and that step forces you to have some constituent identifiers mapped for matching.

*Item Implementation*:

* No enforcement at the item level. If you inadvertently create incomplete records through with a bad file, you can always back the upload out.

**Does the input file include enough information to save valid phone, email, address or activity records for the constituent?**

*Conceptual Rules*:

* For a phone record, phone number or extension must be present.
* For an email record, email address must be present.
* For an address record, some address term — street, city, state or zip must be present.
* For activity records, issue must be present.

*Column Implementation*:

* Enforced at the mapping stage for phone, email and address — not required to furnish phone, email or address, but cannot map phone, email or address type without mapping corresponding data. These rules are enforced in WIC\_Entity\_Upload\_Map.
* Enforced at the default setting stage for activity. If activity data is supplied, issue must be either mapped from the source file or defaulted. Default values are saved as soon as changed, but upload status is not updated to “defaulted” until all necessary choices are made. This rule is enforced by the function decideWhatToShow() in upload-set-defaults.js.

*Item Implementation*:

* The requirement that some data be present on phone, email or address records is enforced at the final upload stage in WIC\_Entity\_Upload\_Complete — phone, email or address subrecords that are missing all values are simply not stored.
* For activity records, WIC\_Entity\_Upload\_Validate enforces the limited required logic for activities (issue must be present) that is enforced in form submissions *but only if issue is mapped.* Issue and activity dates can be set later in the upload process at the default setting stage. If post\_title is mapped, post\_title cannot be blank, but non-blank post\_titles that do not already exist on posts can be created automatically at the default setting stage.

**Is every submitted data item valid?**

*Conceptual Rules*:

* Dates must be parsable as dates (common formats recognized and accepted).
* Emails must meet email standards.
* Etc. — see [Upload Tips — WP Issues CRM Fields](http://wp-issues-crm.com/?page_id=213)

*Column Implementation*:

* Not a column level issue — only applies to items for mapped columns.

*Item Implementation*:

* Strictly enforced at the **Validate Data** stage and, in turn, at subsqequent stages: WIC\_Entity\_Upload\_Validate tests each mapped value on each record using the same validation routines that are used in the form context. If any field fails validation, the record is marked as having failed validation.
* You do have the option to by pass all validation by running express as long as you are not mapping activities.

**Is the data from a better source than existing data?**

*Conceptual Rules*:

* The user can make a decision as to whether input data is at least as reliable as what is already in the database and so whether that data should be allowed to replace what is in the database.
* If not, then the user can choose to not do updates on the core constituent data (name, date of birth, gender, etc.) and address data. This would make sense, for example, if one were uploading a list of people to whom one wanted to add the same activity. The user might define soft matching ( for example, based on first five characters only) to maximize matching on low quality input data and get the activities loaded, but would want to avoid updating good database records with unreliable name and address data.

*Column Implementation*:

* At the default setting stage, the user can check “Protect primary constituent data and address”. If this value is checked, then no updates will be performed for the constituent or address records. However, setting protection on will not prevent the addition of new constituents. This logic is enforced in WIC\_Entity\_Upload\_Complete.

*Item Implementation*:

* Setting protection on will not prevent the addition of address information if the address type (set by default, or, if default not set, on the individual record) is not already present on the database. So, if one had reliable home addresses one wanted to protect, one could set protection on, but this would not prevent the upload of rougher work address data mixed in to the input file.

**Should blank values overly existing data?**

*Conceptual Rules*:

* Generally, the answer to this will be no, but it is a user call. One might want to allow blank values to overwrite existing values if one felt that the existing values were unreliable.

*Column Implementation*:

* N/A

*Item Implementation*:

* At the **Set Defaults** stage, the user can check “Protect all fields from being overwritten by blank input.” If this value is checked (which is the default), then no updates will be performed to any field that would write a blank into the field *for an existing record or subrecord*. This logic is enforced field by field in WIC\_Entity\_Upload\_Complete.