

4 User Guide: Critical Records Program

Concern regarding data loss from the data inventories inspired the *Critical Records Program* component of this product. The large number of expected datasets with different update dates and update frequencies will be very difficult to maintain manually. The *Critical Records Program* is designed to alleviate this concern by discovering data inventory records that are overdue, due, or due soon for updates or that have missing, invalid, or illogical data.

The *Critical Records Program* reads data inventory records from multiple data inventory files to discover and write *critical* records to a report. See section 5.4.2.6 *Field Descriptions* for a summary of each field in the critical records report. Critical records are records that have one or more of the following health status notifications:

- Records are ‘Public’ but ‘Not yet open’
- Restricted or Private records that do not have a ‘Public’ record in the data inventory
- Records are overdue for updates
- Records are due for updates
- Records should be updated soon
- Missing or invalid data in one or more of the following critical fields:
 - Date Created/Imported
 - Internal Priority
 - Update Frequency
 - Date Updated [†]
 - Data Classification
 - Sensitivity Rating
 - Public Status

Invalid values include any values that do not match one of the values accepted for a given field or that are not dates in a date field, and this includes values with minor typographical errors or inconsistent capitalization.

Any data inventory record could appear more than once as a record in the critical records report. For example, a dataset that is overdue for updates and that is missing data in the Data Classification column will have two entries in the critical records report. The *Critical Records Program* assigns a health status to each critical record to alert users about what needs to be done to maintain the data inventory record and resolve the health status notifications for it. See section 5.4.2.5 *Health Status Descriptions* for more information on the health status classifications assigned by the program.

[†]The Date Updated column in the data inventory is validated for valid date format only, not missingness; this field may be missing if the record has not been updated since it was created/imported.

Important note: Do not edit the critical records report. Resolve all errors directly in the data inventory files. To assist data coordinators in locating the correct data inventory record, the critical records report includes the filename and row number of the record in the Source Filename and Source Row Number columns, respectively.

4.1 Data Inventory

4.1.1 Data Inventory Features

To assist users in entering valid data into fields used by the *Critical Records Program*, the data inventory template consists of data validation for Internal Priority, Update Frequency, Data Classification, Sensitivity Rating, and Public Status. Also, the columns for Date Created/Imported and Date Updated are formatted as Date. Excel data validation functions better if columns are inside a table, therefore, the data inventory template is formatted as a table up to row 50.

Another feature for the data inventory is frozen rows and columns. The first row is frozen, which allows users to see the headers for every record they enter. The first two columns are frozen so users can see the dataset title as they enter data into the columns.

4.1.2 Protocol for Dataset Titles in the Data Inventory

The Dataset Title column in the data inventory is required. The *Critical Records Program* does not check if this field is missing. It only reads records up to the first row with a blank value in this field.

All dataset titles must be unique for a given department or data inventory file. The exception to this rule is for restricted and private datasets. When any restricted or private dataset is transformed to remove personally identifiable information, the new, public dataset must have the same title as its restricted or private counterpart. If a restricted or private dataset does not have a public counterpart, that dataset will appear in the critical records report.

4.1.3 Program Data Inventory Column Indices

The *Critical Records Program* uses Excel column indices when reading the data from the data inventory columns. In Excel, column A has index=1, column B has index=2, and so on. If any data inventory columns are deleted, added, or rearranged, the indices in the code must be verified and possibly modified for the program to work correctly. Renaming data inventory columns will **not** affect the program.

The indices for data inventory columns used by the program are defined as constants before the macro in the Visual Basic Editor as follows:

```
Const DATA_SRC_INDEX = 1
Const DATA_TITLE_INDEX = 2
Const DATE_CREATED_INDEX = 4
Const PRIORITY_INDEX = 5
Const DATA_STEWARDS_INDEX = 13
Const UPDATE_FREQ_INDEX = 19
Const DATE_UPDATED_INDEX = 21
Const CLASSIFICATION_INDEX = 23
Const SENSITIVITY_INDEX = 24
Const PUBLIC_STATUS_INDEX = 29
```

See section 4.5 *Modify the Program Code* for instructions on opening the Visual Basic Editor to modify the code.

4.2 Prepare to Run the Critical Records Program

1. Verify that all records in each data inventory have a value in the Dataset Title column. The program only reads records up to the first row with a blank value in this field.
2. All data inventory files must be in the same folder so filenames must be unique. It is recommended to include the department and/or division in the filenames. The *Critical Records Program* root folder contains a *Data Inventory Files* folder for placing copies of data inventory files. This is a logical location for the data inventory files, but it is not required to place them there.
3. If columns in the data inventory have been rearranged or removed or new columns added since the program was last run, program constants related to the data inventory indices may need to be updated to match the new data inventory column indices. See section 4.1.3 *Program Data Inventory Column Indices* for more information about Excel column indices and the related program constants. See section 4.5 *Modify the Program Code* for instructions on opening the Visual Basic Editor to modify the code.
4. If running on a particular computer for the first time, add correct paths to SOURCE_PATH and TARGET_PATH as follows:
 - a) Change the value of the SOURCE_PATH constant to the absolute path of the folder containing the data inventory files. Make sure to have the backslash at the end. It may be helpful to create copies of the data inventory files and place them in the *Data Inventory Files* folder that is within the program folder, but this is not necessary.
 - b) Change the value of the TARGET_PATH constant to the absolute path of the *Critical Records Reports* folder, which is where the program will save critical records reports. This folder is a subfolder of the *Critical Records Program* folder.

See section 4.5 *Modify the Program Code* for instructions on opening the Visual Basic Editor to modify the code.

4.3 Run the Critical Records Program

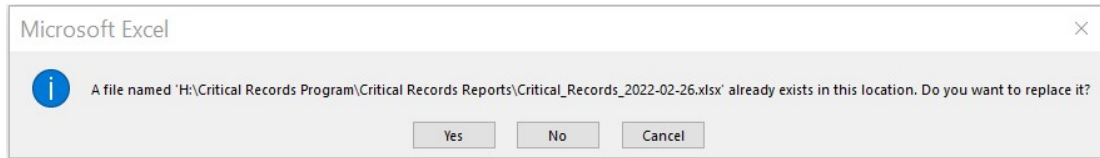
1. Open *Critical Records Reports Macro.xlsm*
2. Click **Create Critical Records Report**.

The Critical Records Program creates an Excel file and saves it as Critical_Records_<today's date>.xlsx to the Critical Records Report folder (the TARGET_PATH). The date is included to allow for archiving past reports, if desired.

If the program is run more than one time in a day without deleting the other report, Excel will display a dialog box asking if it is okay to overwrite the existing file. Selecting **Yes** only overwrites the critical records report and does not affect any of the data inventory files.

Important note: If a critical records report file from the same day is open when running the report, the program will close the report without saving but will prompt the user before overwriting it.

3. Click **Yes** to overwrite the existing file when prompted with the following message.



Copy columns A through F of each table and paste into the newsletter. Use judgement when copying more columns because the text size may be too small to read when more than six columns are fit onto one page. See section 5.4.2.6 *Field Descriptions* for a summary of each field in the critical records tables.

4.4 Due Dates, Update Windows, and Level of Criticality Assignment

The following factors determine the level of criticality assigned to data inventory records that are overdue, due, or due soon to be updated:

- Internal Priority data inventory column value
- Update Frequency data inventory column value
- Date Update Due, which is calculated using program constant WEEKS_TO_UPDATE_FREQ or WEEKS_TO_UPDATE_YR and the date in the Date Updated (or Date Created/Imported) data inventory column
- Update windows, which are defined by program constants DUE_LIMIT, DUE_SOON_LIMIT_FREQ, and DUE_SOON_LIMIT_YR

The following diagram illustrates the interaction of these factors in determining the level of criticality for each record.

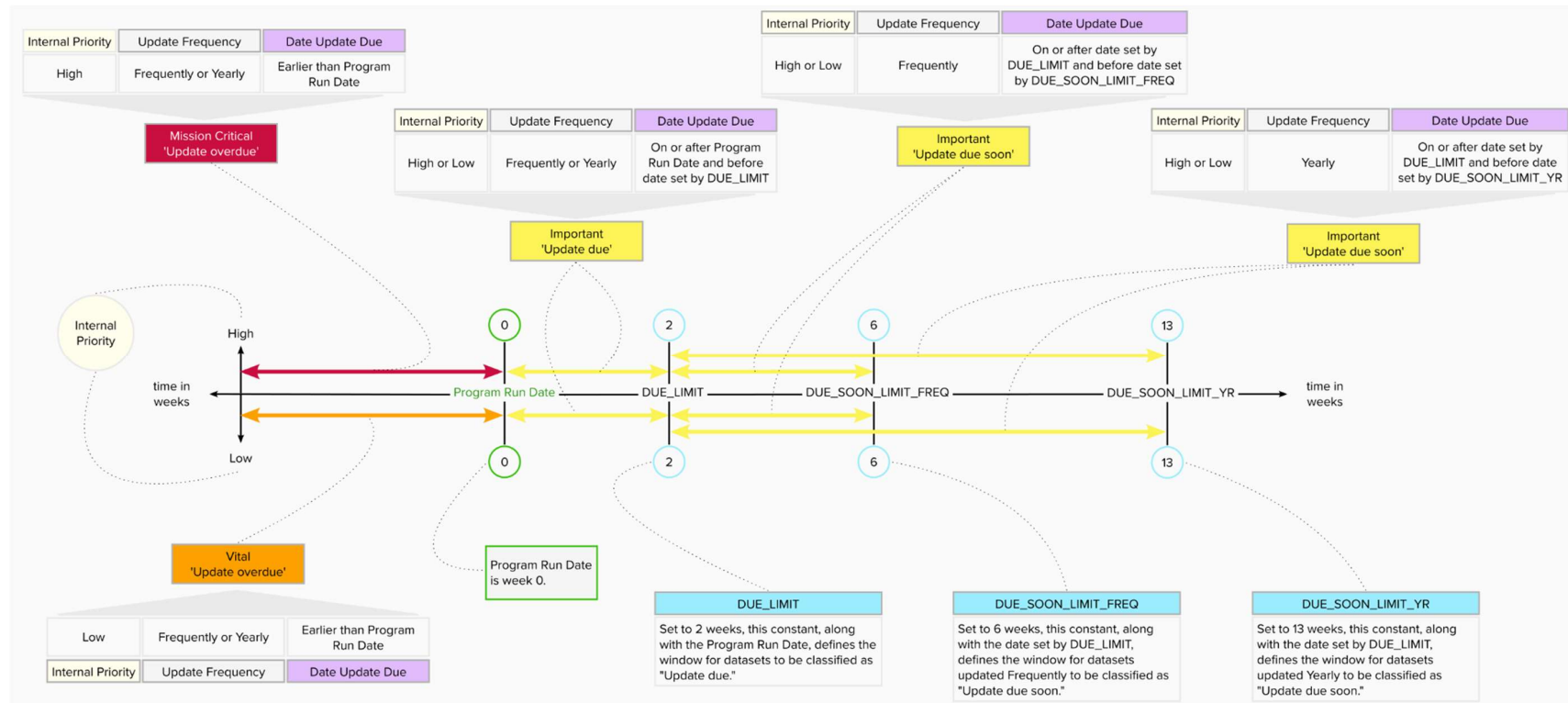


Figure 1: Outline of factors determining date update due, health status windows, and levels of criticality assignment

The update due date, update windows, and assigned level of criticality can be modified. See sections 4.4.4 *Modify the Date Update Due or Update Due Windows* and 4.4.5 *Modify the Level of Criticality Assignment* for information on modifying program functionality with respect to ‘Update overdue,’ ‘Update due,’ and ‘Update due soon’ health status windows and the level of criticality assignment.

4.4.1 Date Update Due

Due dates for updating a data inventory record depend on the Update Frequency and Date Updated (or Date Created/Imported) data inventory column values and are calculated as follows:

Date Update Due			
Update Frequency Value	Constant Name	Constant Value *	Update Due Date Calculation
Frequently	WEEKS_TO_UPDATE_FREQ	13	Date Updated ** + WEEKS_TO_UPDATE_FREQ
Yearly	WEEKS_TO_UPDATE_YR	52	Date Updated ** + WEEKS_TO_UPDATE_YR

Table 3: Date Update Due calculation

* The program interprets this as 13 and 52 *weeks* because of the constant INTERVAL_TYPE, which is set to *weeks*, “ww,” when adding values to dates.

** If the Date Updated is null, the program uses the Date Created/Imported to calculate the update due date.

This means a dataset that is updated frequently is due for an update 13 weeks after its last update (or 13 weeks after it was created/imported), and a dataset that is updated yearly is due for an update 52 weeks after its last update (or 52 weeks after it was created/imported).

4.4.2 Update Due Windows

Two levels of update status, ‘Update due soon’ and ‘Update due,’ signal that a record is due to be updated and give data coordinators time to plan for the updates. Three constants determine the update windows for these classifications: DUE_LIMIT, DUE_SOON_LIMIT_FREQ, and DUE_SOON_LIMIT_YR.

DUE_LIMIT			
Update Frequency Value	Constant Name	Constant Value	Health Status
Frequently or Yearly	DUE_LIMIT	2	Update due
Calculation An update is due when the due date is on or after the Program Run Date and before the Program Run Date + DUE_LIMIT and before the date determined by Program Run Date + DUE_LIMIT.			

Table 4: DUE_LIMIT constant calculation

DUE_SOON_LIMIT_FREQ			
Update Frequency Value	Constant Name	Constant Value	Health Status
Frequently	DUE_SOON_LIMIT_FREQ	6	Update due soon
Calculation An update is due soon when the due date is on or after the date that is two weeks after the report date and before 6 weeks after.			

Table 5: DUE_SOON_LIMIT_FREQ constant calculation

DUE_SOON_LIMIT_YR			
Update Frequency Value	Constant Name	Constant Value	Health Status
Yearly	DUE_SOON_LIMIT_YR	13	Update due soon
Calculation An update is due soon when the due date is on or after the date that is two weeks after the report date and before 13 weeks after.			

Table 6: DUE_SOON_LIMIT_YR constant calculation

4.4.3 Level of Criticality Assignments

This section discusses the structure of the program code with respect to level of criticality assignments. For a high level view of criticality levels, i.e., a table of all possible health status notifications for each level of criticality, see section 5.4.2.4 *Levels of Criticality*.

Three functions assign the update status of data inventory records: OverdueCheck, UpdateCheck, UpdateSoonCheck. Each function receives the following parameters:

- Current data inventory sheet as a Worksheet object
- Row number of the current record that is being processed as loop variable i
- Criteria date (Date Updated or Date Created/Imported)
- Due_dt

Each function returns 0 or one of the following constants:

- MISSION_CRITICAL = 1
- VITAL = 2
- IMPORTANT = 3

The following table outlines the possible outcomes from these functions:

Outcome	Function	Internal Priority	Return Value	Assignment
Mission Critical 'Update overdue'	OverdueCheck	High	MISSION_CRITICAL	Update overdue
Vital 'Update overdue'	OverdueCheck	Low	VITAL	Update overdue
Important 'Update due'	UpdateCheck	High	IMPORTANT	Update due
Important 'Update due'	UpdateCheck	Low	IMPORTANT	Update due
Important 'Update due soon'	UpdateSoonCheck	High	IMPORTANT	Update due soon
Important 'Update due soon'	UpdateSoonCheck	Low	IMPORTANT	Update due soon

Table 7: Level of Criticality assignment

4.4.4 Modify the Date Update Due or Update Due Windows

The values used to calculate the Date Update Due and the two health status update windows, 'Update due' and 'Update due soon' are defined as constants before the macro in the Visual Basic Editor.

The Date Update Due calculation uses one of the two constants, WEEKS_TO_UPDATE_FREQ and WEEKS_TO_UPDATE_YR, depending on the Update Frequency column value. Changing these constants will change how the program calculates Date Update Due.

The ‘Update due’ window is defined by the constant DUE_LIMIT, and the ‘Update due soon’ window is defined by one of the two constants DUE_SOON_LIMIT_FREQ and DUE_SOON_LIMIT_YR, depending on the Update Frequency column value. Changing these constants will change how the program calculates the health status update windows.

See section 4.5 *Modify the Program Code* for instructions on opening the Visual Basic Editor to modify the code.

4.4.5 Modify the Level of Criticality Assignment

The OverdueCheck, UpdateCheck, and UpdateSoonCheck functions each handle ‘High’ and ‘Low’ priority records (based on the Internal Priority value) separately, even though at the time of release, the latter two functions assign the same level of criticality to both ‘High’ and ‘Low’ priority records. This flexible structure allows users to change how the program assigns the update status for one type of record without changing the functionality of the other. The return value for any of the functions can be modified by changing the constant that is returned.

See section 4.5 *Modify the Program Code* for instructions on opening the Visual Basic Editor to modify the code.

4.5 Modify the Program Code

Reasons for modifying the Visual Basic code include

- To set up the SOURCE_PATH for reading the data inventory files and the TARGET_PATH for writing the critical records reports when running the program on a particular computer for the first time.
- To change the data inventory index constants if the columns in the data inventory change.
- To change the constants for calculating the date that a data inventory record is due for updates, WEEKS_TO_UPDATE_FREQ and WEEKS_TO_UPDATE_YR.
- To change the constants for calculating update due windows, DUE_LIMIT, DUE_SOON_LIMIT_FREQ, and DUE_SOON_LIMIT_YR.
- To change how records are assigned a level of criticality.

Modify the Program Code

1. Before making changes, make a copy of the *Critical Records Report Macro v1.0.xlsm* file and save it in the Archive folder located within the main *Critical Records Program* folder. Change the version number in the filename of the file that is not archived.
2. Open the macro for editing:
 - a) Open the Developer tab
 - b) Select Macros (left side)
 - c) Select **CreateReport**
 - d) Click **Edit**

3. If running on a particular computer for the first time, add correct paths to SOURCE_PATH and TARGET_PATH constants. See section *4.2 Prepare to Run the Critical Records Program* for information on these constants.
4. Edit constants related to the calculation of Date Update Due and update due windows as needed. All constants are located before the CreateReport macro in the Visual Basic Editor. See section *4.4 Due Dates, Update Windows, and Level of Criticality Assignment* for more information on these constants and how they are used by the program.
5. Edit the return values for the level of criticality assignment functions as needed. These functions are located after the CreateReport macro in the Visual Basic Editor. See section *4.4 Due Dates, Update Windows, and Level of Criticality Assignment* for more information on the level of criticality functions and how they are used by the program.
6. If any changes other than the SOURCE_PATH and TARGET_PATH constants were made, change the VERSION constant, which is located with all other constants before the CreateReport macro in the Visual Basic Editor.
7. Save the changes.
8. Click the x in the upper right corner of the window to close the Visual Basic Editor.