Aleph Product Update

IGeLU 2022

Carmit Marcus - VP Strategic Partners & Product Management

IGELU Conference
12th - 15th September 2022

ExLibris
Part of Clarivate

Agenda

- Aleph Roadmap
- Aleph 23.4 highlights
- Aleph 24.0 highlights
- Plans for 24.1 and beyond

Aleph Roadmap – Updated

- V23.3 – Minor Release
  - Available – June 3, 2019
- V23.4 – Minor Release
  - Available – January 2021
- V24.0 – Major Release
  - Available – July 2022
- V24.1 – Minor Release
  - Planned – December 2023
- V24.2 – Minor Release
  - Planned – July 2025

NERS enhancements

Aleph V23.4 Highlights (Released January 2021)

Enhancements V23.4

- Circ - barcode scanning for reading room loans - add missing functionality
- Delete Item Records (p-item-11) - add option to use STA field for BIB record management
- Expanding ORCID from authority records to source record
- Check routine needed to identify records that exceed size limits

Infrastructure Changes in V23.4

- End support for Solaris

Aleph 24.0 Highlights

Aleph V24.0 Highlights

- Validation checks across fields - create dependencies for subfields across two or more fields – V24.0
- Prompt for confirmation before deleting memos – V24.0

Prompt for confirmation before deleting triggers

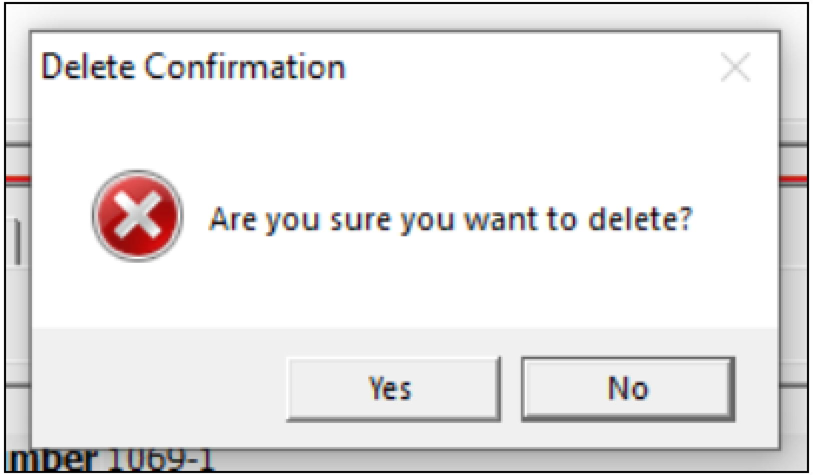
- Current Aleph GUI 'Trigger List' interface doesn't provide an alert when user activate the 'Delete' trigger action
- Once 'Delete' is pressed, the trigger is immediately deleted without any recovery option
- Having a confirmation message before deletion will attract the user's attention and avoid redundant deletion.

Prompt for confirmation before deleting triggers

- **Solution Description**
  - The user's flows for trigger deletion will be enhanced with an additional confirmation message before the delete is executed
  - This change will be implemented for all GUI users without any special customization nor setup
  - It will cover the three on-line triggers that offer the 'Delete'

Prompt for confirmation before deleting triggers

- A new pop-up message will appear when the 'Delete' action will be activated by the user via GUI-Triggers List:



Delete Confirmation

Are you sure you want to delete?

Prompt for confirmation before deleting triggers

- The default selected option will be 'No’
- The user will have to explicitly press the 'Yes' option in order to submit the deletion
- The delete action will be executed AFTER the user's confirmation

Prompt for confirmation before deleting triggers

- Triggers to be covered
  - *GUI Cataloging/Acquisitions-Serials/Circulation - Trigger List (for a specific record)*
  - *GUI Cataloging - Trigger List (for all records)*
  - *GUI Cataloging - Edit Actions menu - Record's Triggers - Trigger List (for a specific record)*

Validation checks across fields

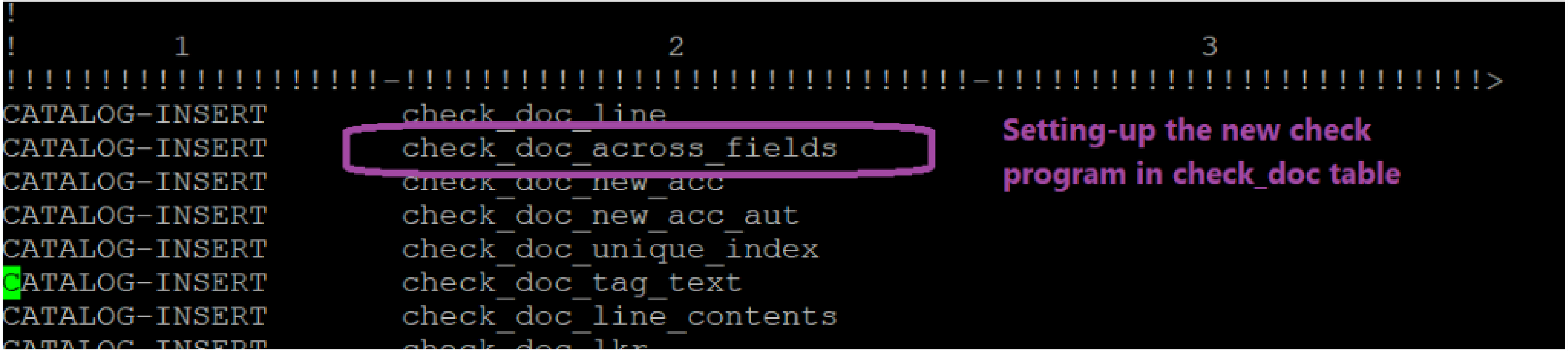
- Current Aleph check_doc validation mechanism, provides the functionality of checking dependencies for subfields within the same field.
- There is no way to define dependencies across fields check_doc_line and check_doc_doc)
- Improving the check_doc functionality to include validation checks across fields, will assist the catalogers when editing a record and improve the quality of the records.

Validation checks across fields

- A new check_doc program will be added to create dependencies across fields: check_doc_across _fields
- This will allow checking a BIB/AUTH record based on data cataloged in subfields of different fields (within the same record)
- Library will configure the relevant dependencies
- New messaged (errors) will be added to reflect the detected inconsistencies
- Will impact cataloging and related services that use 'check doc'

Validation checks across fields

- Setting-up the new check program in check_doc (Example)
  - Table location: ./usm01/tab/check_doc (already exists)
  - Col.1: CATALOG-INSERT -check will be activated when a cataloging record is saved, updated or when the 'Check Record' option is selected from the Cataloging module (already exists)
  - Col.2: The check program: check_doc_across_fields (new functionality)



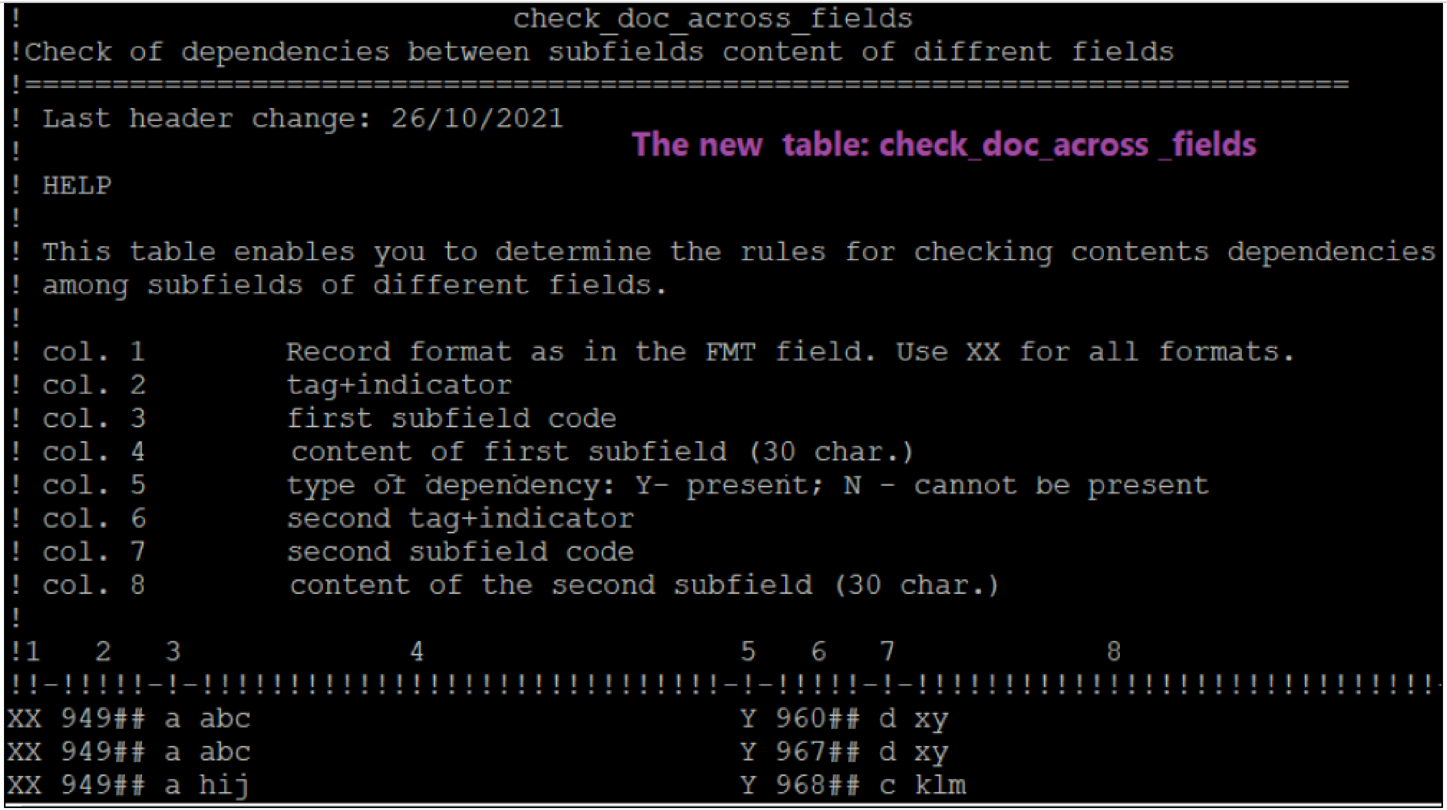
```
!
!        1                         2                            3
!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!>
CATALOG-INSERT       check_doc_line
CATALOG-INSERT       check_doc_across_fields
CATALOG-INSERT       check_doc_new_acc
CATALOG-INSERT       check_doc_new_acc_aut
CATALOG-INSERT       check_doc_unique_index
CATALOG-INSERT       check_doc_tag_text
CATALOG-INSERT       check_doc_line_contents
CATALOG-INSERT       check_doc_lkr
```

Setting-up the new check program in check_doc table

Validation checks across fields

- Setting-up the new table: check_doc_across_fields
  - A new $data_tab/check_doc_across_fields table will define the rules of checking content of subfields across fields.
  - The table will enable to define the rules for checking content dependencies among subfields of different fields
  - Following is a sample table in the demo BIB library: ./usm01/tab/check_doc_across

Validation checks across fields



The new table: check_doc_across _fields

```
!                           check_doc_across_fields
!Check of dependencies between subfields content of diffrent fields
!===========================================================================
! Last header change: 26/10/2021
!
! HELP
!
! This table enables you to determine the rules for checking contents dependencies
! among subfields of different fields.
!
! col. 1        Record format as in the FMT field. Use XX for all formats.
! col. 2        tag+indicator
! col. 3        first subfield code
! col. 4        content of first subfield (30 char.)
! col. 5        type of dependency: Y- present; N - cannot be present
! col. 6        second tag+indicator
! col. 7        second subfield code
! col. 8        content of the second subfield (30 char.)
!
!1   2   3             4                  5   6   7            8
!!-!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
XX 949## a abc                            Y 960## d xy
XX 949## a abc                            Y 967## d xy
XX 949## a hij                            Y 968## c klm
```

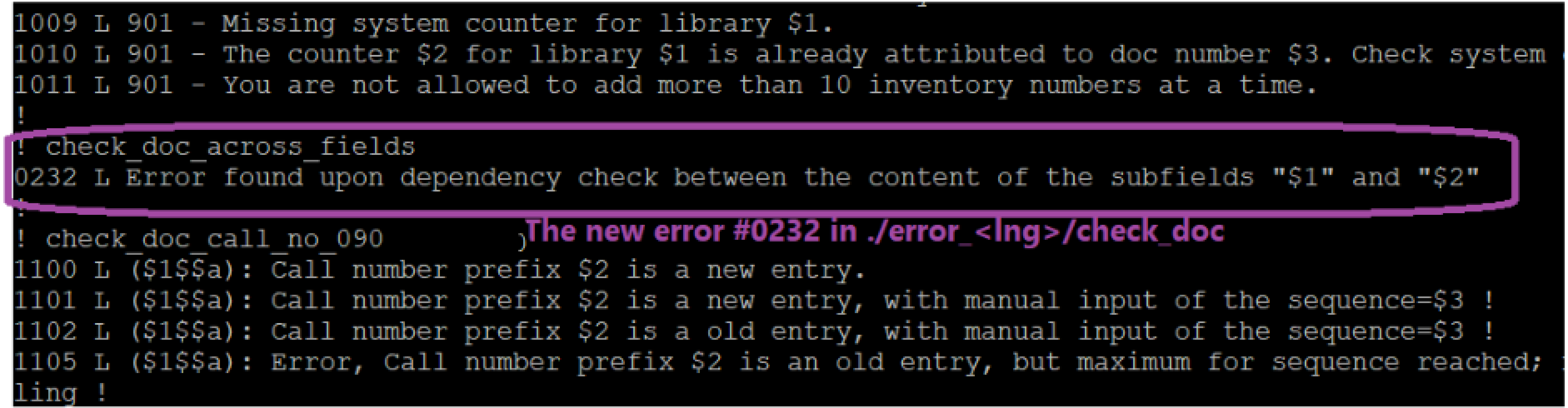
Validation checks across fields

- The rules may be defined per record format (col.1) . For example books (BK) might have different logic then serials (SE). The code XX can be set for all formats
- Col. 2 thru 4 set the 1st tag+indicator+subfield+content that should be checked against the 2nd set
- Col. 6 thru 8 set the 2nd tag+indicator+subfield+content that should be checked against the 1st set
- Col. 5 will set the type of dependency:
  - Y = the defined content should be presented in the other set
  - N = the defined content cannot be presented in the other set

Validation checks across fields

- The New Block/Warning Message
  - A new error text will be added to ./error_<lng>/check_doc file (file already exists)
  - **Error number:** 0232 (hard-code;' cannot be changed by the library)
  - **Error text**: Error found upon dependency check between the content of the subfields "$1" and "$2"
  - $1 - the place holder for the tag: (field+indicator+subfield) of the 1st field (col. 2 and 3 of check_doc_across_fields table)
  - $2 - the place holder for the tag: (field+indicator+subfield) of the 2nd field (col. 6 and 7 of check_doc_across_fields table)

Validation checks across fields



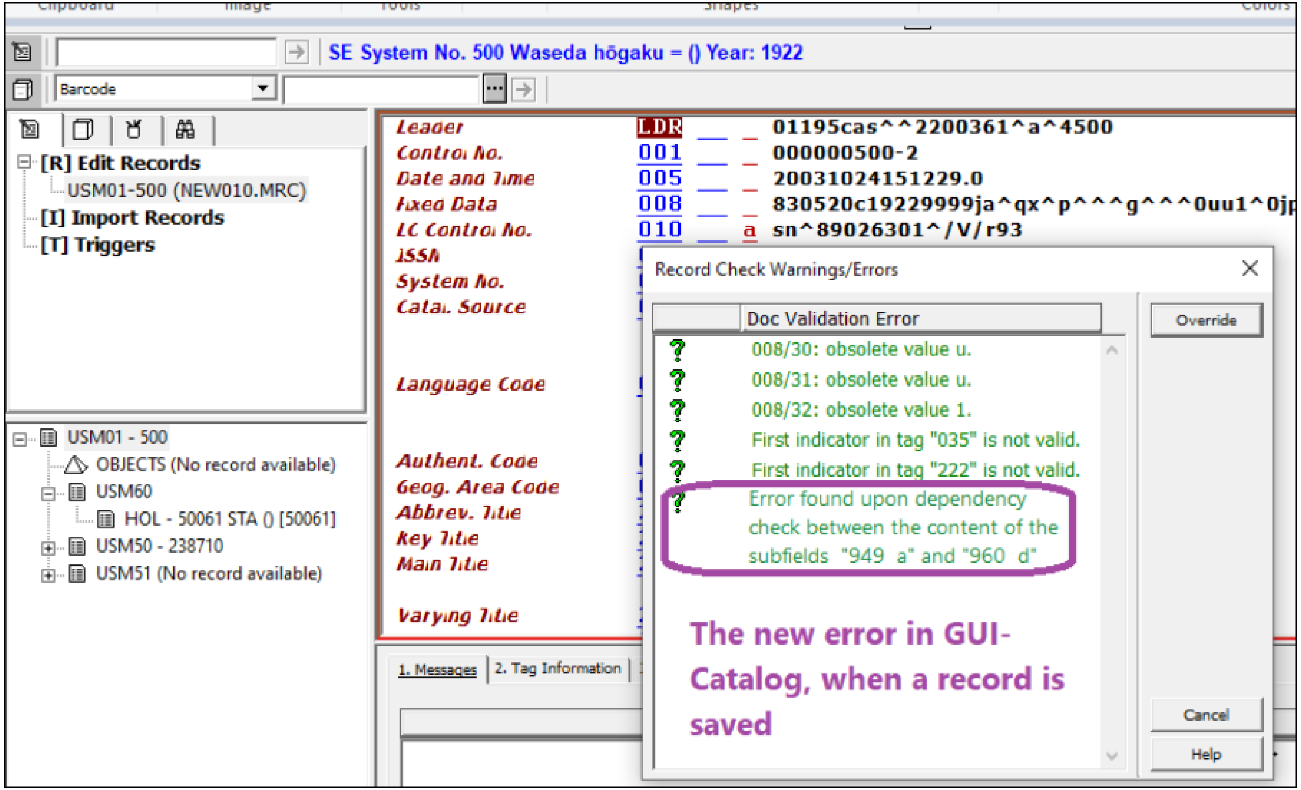
```
1009 L 901 - Missing system counter for library $1.
1010 L 901 - The counter $2 for library $1 is already attributed to doc number $3. Check system
1011 L 901 - You are not allowed to add more than 10 inventory numbers at a time.
!
! check_doc_across_fields
0232 L Error found upon dependency check between the content of the subfields "$1" and "$2"
!
! check_doc_call_no_090
1100 L ($1$$a): Call number prefix $2 is a new entry.
1101 L ($1$$a): Call number prefix $2 is a new entry, with manual input of the sequence=$3 !
1102 L ($1$$a): Call number prefix $2 is a old entry, with manual input of the sequence=$3 !
1105 L ($1$$a): Error, Call number prefix $2 is an old entry, but maximum for sequence reached;
ling !
```

The new error #0232 in ./error_<lng>/check_doc

Validation checks across fields

- Setting the New Check as a Block or Warning
  - As in any other record checks, the library should define the the new check as a Warning (T) or Mandatory (M).
  - T = error which writes a cataloging trigger record, but allows database update.
  - M = error which does not allow database update.
  - This can be done by adding the new entry to $data_tab/check_doc_mandatory (table already exists)

Validation checks across fields



SE System No. 500 Waseda hōgaku = () Year: 1922

```
LDR      01195cas^^2200361^a^4500
001      000000500-2
005      20031024151229.0
008      830520c19229999ja^qx^p^^^g^^^0uu1^0j
010    a sn^89026301^/V/r93
```

Record Check Warnings/Errors

Doc Validation Error

- ? 008/30: obsolete value u.
- ? 008/31: obsolete value u.
- ? 008/32: obsolete value 1.
- ? First indicator in tag "035" is not valid.
- ? First indicator in tag "222" is not valid.
- ? Error found upon dependency check between the content of the subfields "949 a" and "960 d"

The new error in GUI-Catalog, when a record is saved

Infrastructure Changes in V24.0

- RedHat8

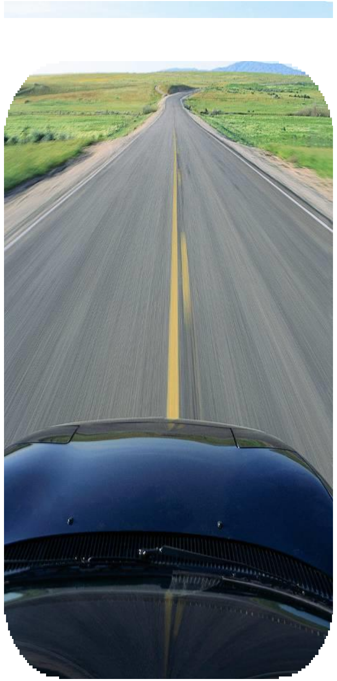
Aleph V24.1 Highlights

- Fiscal Year Purge - remove all vendors without any remaining orders or invoices

Plans for V25 and Beyond

Plans for V25 and Future Versions

- Items for consideration:
  - User group enhancements (NERS)
  - Extending support for protocols and standards such as NCIP, SIP2, RFID, EDI
  - Infrastructure SW/HW certifications



Thank you!

Carmit.marcus@clarivate.com

IGELU Conference
12th -15th September 2022

ExLibris
Part of Clarivate