CONFIDENTIAL AND PROPRIETARY

**Aries Systems Corporation**



**Product Functional Specification**

13.0-18 v9

Select Authors from External Author List File in Invited Workflows

**Status: Ready for QA**

Copyright © 2015, Aries Systems Corporation

This document is the confidential and proprietary property of Aries Systems Corporation. The contents hereof are for use only by recipients specifically identified by Aries, and may not be further disclosed to any other third party. Specifications, designs and drawings contained herein are subject to withdrawal or change by Aries, without notice. Aries shall have no liability resulting from such withdrawal or change. Editorial Manager is a registered trademark and the property of Aries Systems Corporation.

Aries Systems reserves the right to share this specification with other individuals or organizations that have entered into confidentiality agreements with Aries Systems, in which case the specification will be appropriately blinded so that no identifying or proprietary information will be exposed.

**Document Summary**

|  |  |
| --- | --- |
| **Enhancement Name** | **Select from File of Authors to Invite for Proposals** |
| **Specification Number** | 13.0-18 |
| **Enhancement Type** | Aries Funded |
| **Implemented In** | 13.0 |

**Document Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Comments** |
| 6/11/15 | Di Natale | Initial draft v0 |
| 7/13/15 | Di Natale | V1 |
| 8/07/15 | Di Natale | V2 – File upload restricted to one file per invite session, logic clarified and detailed for Author List File format, validation an ingest, interface streamlined/overhauled to use fewer pages and clicks, NOTE field mapped to existing Invitation Notes for Author, added support for Alternate Authors |
| 8/11/15 | Di Natale | V3 – Minor corrections and clarifications added, tweaks to warning messages made to improve visual display |
| 8/13/15 | Di Natale | V4 – Clarifications of search method change, blue spinner, clarification of intended functionality, updated figures, table added showing priority of match conditions  Confirmed Match logic updated to include:   * unauthenticated ORCID and LASTNAME match * consideration of alternate EMAIL value in ADDRESS table |
| 8/17/15 | Di Natale | V5 – edited based on feedback derived from design meeting with Lyndon, CS, and Eng.   * Alternate Author support cut * Noted support for Spec 13.0-33 (13.1) * Error messaging redone – all data displayed in overlay * Limit on size of file set initially to 200 rows – pending engineering experimentation |
| 8/26/15 | Di Natale | V6 Ready for Management Approval |
| 9/4/14 | Di Natale | Approved by management |
| 9/4/14 | Di Natale | Engineering handover |
| 9/8/15 | Di Natale | V7 Changes made per Engineering feedback during handover – Author Selection Summary page no longer only page used during upload, validation and ingest of Author List File; new page used for uploading, Candidates list displayed on modified search results page. Test cases updated. |
| 10/6/15 | Di Natale | V8 – Removed Excel File support per Lyndon – Security issues with allowing Excel files to process on webserver |
| 10/29/15 | Di Natale | V9 – Corrected cancel button behavior on Select Authors – Confirm Selections and Customize Letters page |
|  |  | Removed bad test case, clarified instructions to translation, future development |
| 11/12/15 | Di Natale | Tweaked after OTS, some changes per engineering – tool tip when selecting upload option, images, test cases |
| 11/20/15 | Di Natale | V10 Added Tech Spec |

Specification Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| **Department** | **Date** | **Version** | **Name** |
| Corporate | 8/27/15 | 6 | LH |
| Product Mgt. |  |  |  |
| Engineering | 9/1/15 | 6 | BM |
| Operations |  |  |  |
| Customer Service | 9/4/2015 | 6 | JF |
| Aries KG | 9/1/15 | 6 | RN |
| Sales | 9/1/15 | 6 | RW |

**Table of Contents**

[1 Overview 6](#_Toc429493973)

[1.1 Glossary 8](#_Toc429493974)

[1.2 Related Specifications 8](#_Toc429493975)

[2 Application Impact 9](#_Toc429493976)

[2.1 Author List File 9](#_Toc429493977)

[2.1.1 Format of Author List File 10](#_Toc429493978)

[2.1.2 Metadata Mapping at Proxy Registration 13](#_Toc429493979)

[2.1.3 Validation of Author List File 14](#_Toc429493980)

[2.1.4 Ingest of Author List File and Matching of Records to the EM Database 22](#_Toc429493981)

[2.2 Author Selection Summary page [authorSelectionSummary.aspx] 31](#_Toc429493982)

[2.3 Upload Author List page [new page] 36](#_Toc429493983)

[2.4 ‘Author Candidates’ Results page [new page] 44](#_Toc429493984)

[2.5 Select Authors – Confirm Selection and Customize Letters page [confirmAndCustomizeInvitedAuthors.aspx] 55](#_Toc429493985)

[3 Database Impact 58](#_Toc429493986)

[3.1 DB Content Changes 58](#_Toc429493987)

[3.2 UDB Impact 58](#_Toc429493988)

[4 Multi-Lingual Considerations 58](#_Toc429493989)

[5 Performance Processing Impact 58](#_Toc429493990)

[5.1 Performance Requirements 58](#_Toc429493991)

[5.1.1 Data Server 58](#_Toc429493992)

[5.1.2 Web Server 58](#_Toc429493993)

[5.1.3 Network 58](#_Toc429493994)

[5.1.4 Client Machine 58](#_Toc429493995)

[5.2 Engineering Assessment 58](#_Toc429493996)

[5.2.1 Engineering Test Results 59](#_Toc429493997)

[6 Merge Field Impact 59](#_Toc429493998)

[7 Upgrade Script Impact 59](#_Toc429493999)

[8 Site Cleanout 59](#_Toc429494000)

[9 Report Implications 59](#_Toc429494001)

[9.1 Canned Reports 59](#_Toc429494002)

[9.2 General Data Export 59](#_Toc429494003)

[9.3 Custom Reports 59](#_Toc429494004)

[9.4 Enterprise Analytics Reporting 59](#_Toc429494005)

[9.5 Cross-Publication EAR 59](#_Toc429494006)

[10 Hosting Documentation Impact 59](#_Toc429494007)

[11 Portal (IJRS, etc.) Impact 59](#_Toc429494008)

[12 Hosting Administration Impact 60](#_Toc429494009)

[12.1 Hosting Administration Audit Trail 60](#_Toc429494010)

[13 Sys Admin Audit Trail 60](#_Toc429494011)

[14 Import/Export Impact 60](#_Toc429494012)

[14.1 Transmittal Impact 60](#_Toc429494013)

[14.2 Aries Export Impact 60](#_Toc429494014)

[14.3 Aries Import Impact 60](#_Toc429494015)

[15 Transfer Impact 60](#_Toc429494016)

[16 Cross-Publication Impact 60](#_Toc429494017)

[17 Documentation and Training 60](#_Toc429494018)

[18 Operations’ Considerations 60](#_Toc429494019)

[18.1 Journal Rename 60](#_Toc429494020)

[18.2 Journal Redirects 60](#_Toc429494021)

[19 Utilities 60](#_Toc429494022)

[20 Audit/Monitor/Diagnostics 61](#_Toc429494023)

[21 Journal Configuration 61](#_Toc429494024)

[22 Style Sheet impact 61](#_Toc429494025)

[23 Deployment Considerations 61](#_Toc429494026)

[23.1 Associated Revenue Model 61](#_Toc429494027)

[23.2 Sales and Marketing Rollout Anticipation 61](#_Toc429494028)

[23.3 Customer Service Rollout Preparation 61](#_Toc429494029)

[23.4 Release Target 61](#_Toc429494030)

[24 Future Development 61](#_Toc429494031)

[25 Mobile Considerations 62](#_Toc429494032)

[26 Use Cases and Test Scenarios 62](#_Toc429494033)

[26.1 Use Cases 62](#_Toc429494034)

[26.2 Specification Test Cases 62](#_Toc429494035)

[27 Appendix A: Technical Specification 72](#_Toc429494036)

# Overview

In a books workflow, different authors are typically invited for each chapter. These authors are selected by publishers from a field of study ahead of time by a publication staff working on the book which then compiles them into a list. Today, when using the existing Proposal functionality for invited workflows in Editorial Manager to manage a books project, inviting the authors from these lists is a time consuming task which requires users to search for authors and then register those who are not already in the system.

In order to streamline this workflow, customers would like the ability to upload a list of Authors in a standardized format instead of using the existing search functionality when inviting Authors. An Editor would use the existing functionality for Proposals to invite the Authors, but instead of using the search options on the *Author Selection Summary* page to locate Authors already registered in EM, the Editor would have the option to select a previously prepared, local file ( \*.csv. \*.tab) that contains the intended Author’s names, emails, and optionally, ORCID iDs and a NOTE field. The file would be created outside of EM and then uploaded into the system.

The optional NOTE field in the file is a text field intended to hold a description the content an Author has been invited to provide. For example, the NOTE field could hold the title of the book chapter the Author will be asked to provide which would then be carried over into the invitation letter sent to the author. In the event that the provided text is over the limit it will be truncated rather than causing the upload to fail.

When the Editor selects and uploads a file, the system will read and validate the file and then present the valid records to the Editor to invite within the EM user interface. The system would first attempt to match up these people with existing EM registered users.

* If the Editor chooses to select an Author who is then determined to be a registered user, the system sends the user an invitation, as per the usual EM functionality for inviting Authors for a proposal
* If the Editor chooses to select an Author who is not a registered EM user (or for which EM could not determine a match), the system proxy-registers the person first, and then sends the newly-registered user an invitation, per the usual EM functionality

The option to upload an Author List File will be available at upgrade to users with permission to invite Authors to submit. The functionality will be only available for selecting Authors for invitation. Support will not be included for Alternate Authors. To accommodate the new functionality a new option will be added to the search method drop-down on the *Author Selection Summary* page, ‘Upload Author List’. When the option is selected Editors will be able to upload an Author List File (ALF).

Once the ALF is uploaded the system will perform validation on the file, matching Authors to existing records using the ORCID iD and/or Email address. Authors without existing people records will be noted as such and will have accounts proxy-registered prior to the point of invitation with efforts made to not introduce duplicate accounts when possible. If any required data is missing a warning message will alert the editor to the specific issue without halting the ingest process for the entire list. An uploaded ALF may fail validation or be accepted. In some cases detailed below, parts of the file will be validated and eventually ingested while records with missing or malformed data will be excluded. An overlay will display showing all the records in the ALF and highlighting all errors to the user if they exist after validation and prior to being ingested

Once the uploaded file is ingested, the Authors included from the ALF will appear as a list of ‘Author Candidates’ on a new *‘Author Candidates’ Results* page. Authors may be selected by default, but Editors may exclude them by deselecting them in the grid just as they would in the existing search options. This gives Editors a manual override if an error is found after the list is uploaded.

Authors who appear in the uploaded Author List File multiple times will appear once in the ‘Author Candidates’ grid. Duplicates in the list are not assumed to be erroneous; instead the number of appearances of an Author in the list is interpreted as the appropriate Number of Invitation Letters the author will receive. The number of appearances of a user in the file is tallied and entered automatically in the Number of Invitation Letters field allowing authors who are writing multiple book chapters to be accounted for in the Author List File. Editors can freely adjust this number in the interface once the ALF is ingested.

Once the ‘Author Candidates’ grid is populated to an Editor’s satisfaction they will be able to proceed with the invitation process per existing functionality. At this time Authors included in the ‘Author Candidates’ grid without EM accounts will be proxy-registered prior to invitations being sent using the information provided in the Author List File using a process similar to what is in place for Reviewers registered through the existing Reviewer Discovery functionality.

A simplified sample workflow for the process of importing an Author List File follows:

1. Editor chooses to invite Authors
2. Editors selects ‘Upload Author List’ option on the *Author Selection Summary* page and is brought to the new *Upload Author File* page when clicking ‘Go’
3. Editor selects and uploads an Author List File on the new *Upload Author File* page
4. Editorial Manager confirms that the file is in the correct format (file type and structurally)
5. Contents of the file are displayed in an overlay on the new *Upload Author File* page
   1. Instances of missing or invalid data are highlighted and marked for exclusion in this overlay
6. Editor chooses to import the file after view the data in the overlay and is sent to the new *‘Author Candidates’ Results page*
   1. Editorial Manager attempts to match uploaded Author records to existing EM accounts while the transition from the *Upload Author File* page to the *‘Author Candidates’* *Results* page takes place
      1. Displays a loading overlay while this process is running
7. Editorial Manager populates the approved contents in the ‘Author Candidates’ grid on the new *‘Author Candidates’* *Results* page
8. Editor makes any adjustments to selections deemed necessary and clicks ‘Proceed’ to invite Authors via the *Select Authors – Confirm Selection and Customize Letters* page
   1. Editor can then customize and send invitation letters to all selected Authors
      1. NOTE field values will be pulled into invitation letters using the Invitation Notes for Author field
   2. When the Editor clicks proceed on the *Select Authors – Confirm Selection and Customize Letters* page EM proxy-registers any selected Author which doesn’t have an existing EM account
   3. Displays a loading overlay while this process is running

## Glossary

**Author List File (ALF) –** The file to be uploaded by Editors containing a list of potential Authors which Editors may invite to submit via Editorial Manager’s invited workflows

**Primary ADDRESS record** – A record in the ADDRESS table where ADDRESS.PRIMARY = 1

**Confirmed Match** – A Confirmed Match is a match between an Author record found in the Author List File and the EM database determined using the logic defined by this specification

**Invitation Session (or Invite Session)** – This describes the start to finish process of selecting Authors for invitation and then inviting them. As defined in this specification an Author List File may be used once per invitation session. If a file has been uploaded an Editor can use the ‘Remove’ link in the interface to remove the file’s contents from the ‘Author Candidates’ grid and then choose to upload a new file.

**Validation** – The process by which the data held within an Author List File (ALF) is determined to be properly formatted, accurate and valid. (See Section 2.1.3 for full details).

## Related Specifications

**3.0-01** – Proposals and Commentaries

**3.1-11** – Enhancements to Proposals and Commentaries

**6.0-06** – Proposal Improvements

**7.0-05** – Invite Author Improvements

**7.0-67** – Invited Author Text Entry Box

**9.0-34** – ORCID (introduced the ORCID iD)

**9.0-34B** – ORCID (Introduced validation and checksum functionality)

**10.0-31** – Reviewer Discovery (layout and behind the scenes proxy registration)

**10.0-77** – Proxy Registration Security Enhancements

**11.0-01** – Drag and drop attach files

**12.0-37** – Required Fields for Proxy Registration

# Application Impact

The enhancements described in this specification affect the following areas:

1. Author Selection Summary page [authorSelectionSummary.aspx] (Section 2.2)
2. Select Authors – Confirm Selection and Customize Letters page [confirmAndCustomizeInvitedAuthors.aspx (Section 2.5)

## Author List File

The functionality introduced in this specification will require that a file be created outside the system which will then be ingested by Editorial Manager in order to populate a list of potential Authors to be selected. This file is called the Author List File (ALF). The ALF will contain the first name, last name and email address (multiple email addresses per record is supported) of potential Authors as well as optional fields for automatically selecting an Author for invitation, ORCID iD and notes.

The ALF will be uploaded, validated (See Section 2.1.3) and ingested into the system. Records in the file will be matched against existing EM people and address records to find existing accounts and for those without existing accounts, the records will be used to proxy-register accounts for new users in order to invite Authors. These potential Authors will first be displayed in an overlay on the *Upload Author List* page and then, once confirmed by an Editor, in ‘Author Candidates’ grid on the *‘Author Candidates’ Results page* (See Section 2.4) where an Editor can choose to exclude authors manually by deselect them and modify the number of invitations they should receive. Once an Editor proceeds to invite Authors, proxy registration will occur for Authors without identifiable existing accounts if these Authors have been selected. When the proxy registration occurs, data from the file will be mapped to specific tables and fields to populate the account with a baseline of metadata (See Section 2.1.2).

As defined in this specification an Author List File may be used once per invitation session. If a file has been uploaded an Editor can use the ‘Remove’ link in the interface to remove the file’s contents from the ‘Author Candidates’ grid. An Editor may then choose to upload additional ALFs once by starting a new invitation session once the prior session has concluded. Individual Authors can be excluded from the uploaded list by being deselected by an Editor, allowing Editors to correct any incorrect data which may have gotten through the validation.

The format of the ALF as well as validation and field mappings are outlined below.

**NOTE:** The format of the Author List File detailed below is based on what is needed today to implement the ability to upload and ingest a list of Authors into EM for invitation. The technical design of the upload, validation and ingest process must allow for additions implemented by future specifications if additional data fields are found necessary for inclusion in an Author List File.

Two upcoming specifications (13.0-33 – Assign Authors and 13.0-31 – Predefined Submission Title) will introduce functionality which will feed off of the data included in the defined Author List File as well as potential new data fields which would require the additional columns to be added to the Author List File. Beyond these examples, the technical design for the upload and ingest process must assume that as of yet undetermined future functionality may need to easily add data to the Author List File which is included in the validation being performed without requiring a redesign.

### Format of Author List File

[To be confirmed by Engineering]

1. Editorial Manager will support the uploading of an Author List File (ALF) by Editors in the following formats:
   1. Comma Separated Value (\*.csv, \*.txt)
   2. Tab Delimited File (\*.tab, \*.txt)
2. The file contains a list of people to be invited as potential Authors
   1. A limit of 200 records is enforced on the file [To be confirmed by Engineering]
      1. The enforced limit must be able to be easily changed on the backend in the event it needs to be changed due performance reasons or user feedback
   2. Six pieces of data will be usable in the file
      1. Three required fields
         1. First Name
         2. Last Name
         3. Email Address
            1. Multiple email addresses can be entered in this field with each value separated by a semi-colon
      2. Three optional fields
         1. NOTE
            1. No limit is enforced on the length of this field, though it’s appearance in the interface may be truncated depending on where the content is displayed (See Section 2.2)
            2. The value of this field is carried forward into the Invitation Notes for Author field which can be merged into Author Invitation letters
         2. ORCID iD
            1. May be entered as XXXXXXXXXXXXXXXX OR XXXX-XXXX-XXXX-XXXX
         3. INVITE
            1. A True/False field which will be used to determine whether or not an Author is selected for invitation

A value of TRUE selects the Author for invitation

FALSE results in the Author not being automatically selected for invitation

A null value is treated as a FALSE value and will result in the record not being selected by default

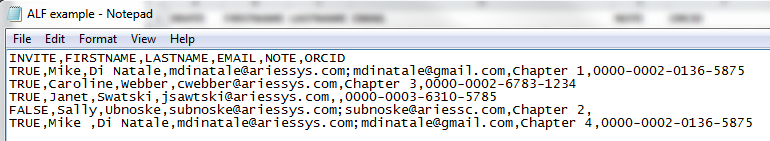
* + - * 1. In future development this field may have other applications

Spec 13.0-33 adds the ability to assign Authors and while a specification has not yet been written the intent here is that a FALSE value could indicate an assignment and a NULL value could result in no selections being made

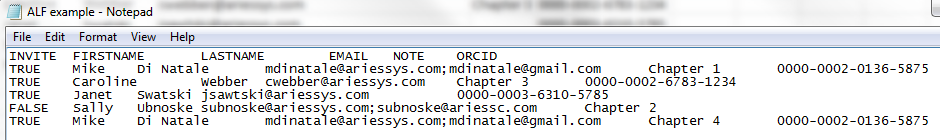
* + 1. Additional fields will be added via future development (Spec 13.0-31)
  1. Each field will be entered in its own column which must be identified using a specific column header
     1. Variant header text will be accepted
     2. Column headers:
        1. FIRSTNAME
           1. Variants: FIRST NAME, FIRST-NAME
        2. LASTNAME
           1. Variants: LAST NAME, LAST-NAME
        3. EMAIL
           1. Variants: E-MAIL, E MAIL
        4. NOTE
           1. Variant: NOTES
        5. ORCID
           1. Variant: ORCID ID
        6. INVITE
           1. Variant: INVITIATION
     3. Column header text is case insensitive
     4. A column for each required field must be present when the file is uploaded
        1. A missing required header will trigger a validation failure (See Section 2.1.3)
        2. Columns for optional fields can be excluded if no values are present for an optional field
        3. If no values for an optional field exist an empty column with header will also be acceptable
     5. A specific column order is not required as the headers will be used to identify the data so that it can be mapped correctly
     6. Data entered in the file without being placed under an appropriate header will be ignored and not ingested
  2. The file will contain one row for each Author record
  3. Blank rows in the file will be ignored if found
  4. Duplicate rows are allowed
     1. If a duplicate row is present by design the new functionality will interpret the intent to be that each listing represents an individual invitation to be sent to an Author
        1. The number of duplicates will be used to populate the Number of Invitation Letters field in the ‘Author Candidates’ grid (See Section 2.4)
        2. The intention is to allow the same Author to be given multiple assignments at the same time with different NOTE values associated
     2. A row is determined to be a duplicate if the FIRSTNAME, LASTNAME and EMAIL address match
        1. If the optional ORCID field is populated it must match as well
        2. If the ORCID field occurs in only one of the two otherwise matching records, the records are displayed as separate entities in the ‘Author Candidates’ grid as detailed in Section 2.4
           1. These records are marked in the grid as potential duplicates
     3. When Authors with duplicate rows in the ALF appear in the ‘Author Candidates’ grid they appear only once
        1. The Number of Invitation Letters field in the grid is populated with a count of an Author’s duplicate rows
           1. Note: The default value for Number of Invitation Letters field is 1

1. Future development (Spec 13.0-33 and Spec 13.0-31) will require additional columns for new data fields to be added to the Author List File
   1. Technical design must account for this future scenario

**Figure 2**: Partial Screenshot of comma separated value version of Author List File, displayed in Notepad, listing five potential authors in the correct format. Multiple email addresses are separated by a semicolon. Note that the NOTE and ORCID columns can contain null values because they are optional. All other fields are required.



**Figure 3**: Partial Screenshot of tab delimited version of Author List File, displayed in Notepad, listing five potential authors in the correct format. Multiple email addresses are separated by a semicolon. Note that the NOTE and ORCID columns can contain null values because they are optional. All other fields are required.



### Metadata Mapping at Proxy Registration

1. Six fields are used by the Author List File (ALF)
   1. Three fields are required
      1. FIRSTNAME
      2. LASTNAME
      3. EMAIL
   2. Three are optional
      1. NOTE
      2. ORCID
      3. INVITE
2. When a record from the ALF is used to proxy-register a user, account data from the uploaded file is mapped to the appropriate locations in the database
   1. FIRSTNAME
      1. Maps to PEOPLE.FIRSTNAME
   2. LASTNAME
      1. Maps to PEOPLE.LASTNAME
   3. EMAIL
      1. Maps to ADDRESS.EMAIL
         1. May contain multiple email addresses separated by semicolons
      2. This entry in the address table is set as the primary record.
         1. Also set ADDRESS.PRIMARY = 1
   4. ORCID
      1. Maps to PERSONAL\_IDENTIFIERS.ORCID
         1. If the publication is configured with ORCID iD set to Hidden under the Configure Proxy Registration page, then the ORCID iD is not added to the People Record created by this proxy registration.
            1. Any supressed data is ignored and noted as such on an overlay alert which appears following validation (See Figure 4)
      2. This entry in the Personal Identifiers Table for the ORCID iD is not considered Authenticated
         1. To record the ORCID as unauthenticated, also set PERSONAL\_IDENTIFIERS.ORCID\_IMPORT\_AUTHENTICATED = 0
3. The NOTE field is not stored as part of the people or address record
   1. During the invitation process NOTE data will be used to populate the Invited Author Notes field
   2. When invitations are created for Authors the NOTE field value will be mapped to INVITED\_AUTHORS.INVITED\_AUTHORS\_NOTES
4. INVITE is used to determine whether or not the ALF record is selected for invitation when ingested and displayed in the ‘Author Candidates’ grid (See Section 2.2)
   1. INVITE is a TRUE/FALSE field
   2. If TRUE the ‘Select’ checkbox for invitation will be checked off for the ingested record
   3. If the value is FALSE or NULL the ‘Select’ checkbox will not be checked off
      1. Future Development of Spec 13.0-33 may result in assignment actions keying off of the FALSE value for this field

### Validation of Author List File

In order to properly ingest and use the data from an Author List File in Editorial Manager the format and consistency of any ALF upload must be verified and validated. This process will be triggered automatically as Editors upload ALFs.

An uploaded ALF may fail validation or be accepted. Additionally, in some cases detailed below, parts of the file will be validated and eventually ingested while records with missing or malformed data will be excluded. An alert message will detail all errors to the user if they exist. In cases of a partial validation leading to only a portion of a file being ingested, Editors may remove the affected ALF and replace it with a corrected version in order to avoid potentially inviting Authors a second time when attempting to use the initially reject records at a later date. A limit of 200 records per ALF will be enforced initially. [The limit may be changed pending the result of engineering testing and experimentation during the technical design process.]

1. Validation begins when an ALF is uploaded via the *Upload Author List* page (See Section 2.3)
   1. Only one ALF can be uploaded and processed per invitation session
   2. Once the validation process has completed on the uploaded file an overlay is displayed showing the contents of the file
      1. All records contained within the ALF will be displayed
         1. If the number of errors result in the overlay expanding beyond the visible area of the screen the window’s scroll bar will allow a user to scroll down to see the remaining results
      2. Instructional text appears at the top of the overlay
         1. Test reads: “The Author Candidates found in the uploaded file are displayed below. If any errors exist within the provided data they are highlighted with a warning icon. Depending on the issue with the data, the offending record may not be imported. Rolling over the icon where an issue has been discovered will provide information about the issue. The 'Ignore' option can be used to exclude Authors before importing the list.

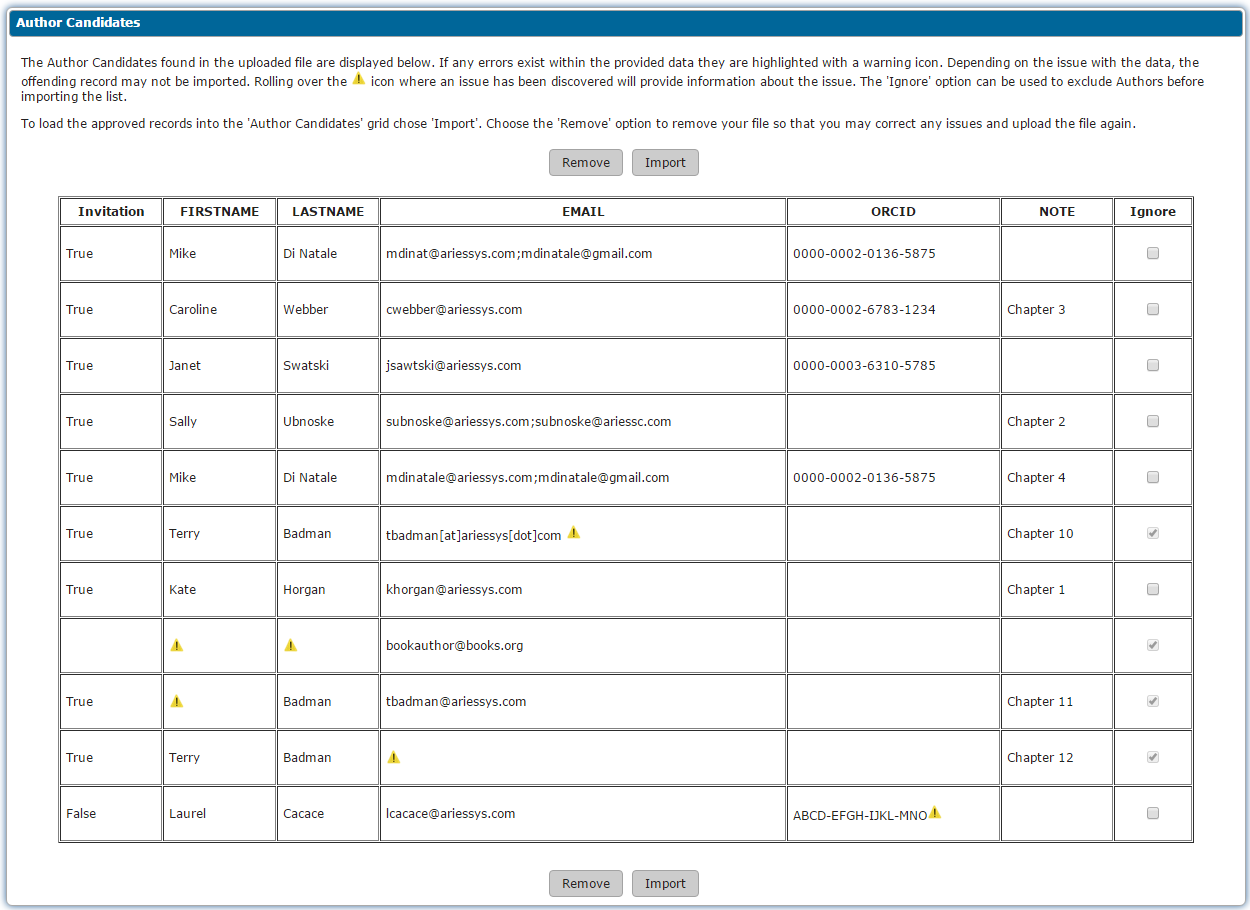
To load the approved records into the 'Author Candidates' grid chose 'Import'. Choose the 'Remove' option to remove your file so that you may correct any issues and upload the file again.”

* + 1. Fields which are missing or formatted incorrectly will be noted with a yellow triangle alert icon ()
    2. Rolling over the  icon will reveal information about the error
       1. When a FIRSTNAME value is not provided, rolling over the icon displays a tool tip
          1. Tool top reads: “First Name is a required field. This record cannot be imported.”
       2. When a LASTNAME value is not provided, rolling over the icon displays a tool tip
          1. Tool top reads: “Last Name is a required field. This record cannot be imported.”
       3. When an EMAIL value is not provided, rolling over the icon displays a tool tip
          1. Tool top reads: “Email address is a required field. This record cannot be imported.”
       4. When an EMAIL value is not formatted correctly, rolling over the icon displays a tool tip
          1. Tool tip reads: “This email address is invalid. This record cannot be imported.”
       5. When an ORCID iD is not formatted correctly, rolling over the icon displays a tool tip
          1. Tool tip reads: “This ORCID iD is invalid. The ORCID value will not be imported.”
    3. An ‘Ignore’ checkbox option will be present for each record displayed in the overlay
       1. The ignore option will let Editors exclude anyone who has been included at this point from the ‘Author Candidates’ grid
       2. The ignore option is selected and disabled for any record missing required data in order to prevent bad data from being used
    4. The overlay provides two options which are displayed at the top and bottom on a page
       1. ‘Remove’
          1. Clicking the ‘Remove’ option discards the uploaded ALF and allows the user to start the process over again
          2. A warning message with ‘Cancel’ and ‘OK’ options triggers when the ‘Remove’ button is clicked

Message reads: “This will remove the uploaded file and discard any existing selections. Are you sure?”

* + - 1. ‘Import’
         1. Clicking the ‘Import’ button dismisses the overlay and displays the ingested records within the ‘Author Candidates’ grid (See Section 2.1.4 for ingest process)
    1. If an Editor wishes to address errors in a file they can remove a file which has been found to have numerous records which will not ingest and then upload a corrected version of that file instead (See Section 2.3)

**Figure 4:** Mocked up overlay of Author records which displays once validation is completed.



**Figure 5**: Mocked up mouse-over text for missing FIRSTNAME value



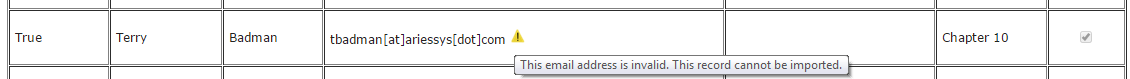
**Figure 6**: Mocked up mouse-over text for missing LASTNAME value



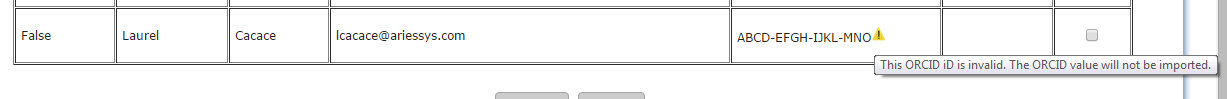
**Figure 7**: Mocked up mouse-over text for missing EMAIL value



**Figure 8**: Mocked up mouse-over text for invalid EMAIL value



**Figure 9**: Mocked up mouse-over text for invalid ORCID value



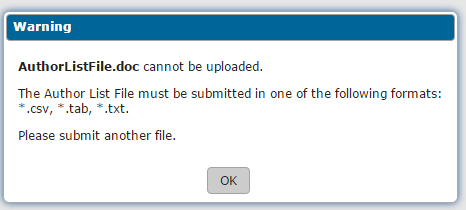
* + 1. Once Import is clicked the overlay is dismissed and the user is sent to the *‘Author Candidates’ Results* page where the ‘Author Candidates’ grid is populated and displayed
    2. Editors will be able to manually remove any additional Authors by deselecting them on the ‘Author Candidates’ grid
       1. Authors will be preselected for invitation by default if the INVITE value in the ALF for a record is set to TRUE and can be deselected and reselected while the Editor reviews the list of Authors in the ‘Author Candidates’ grid
       2. Editors will not be able to add Additional Authors via searching or an additional file until the invitation process has been completed
       3. See Section 2.4

1. If users close the *Upload Author List* page while a file is being uploaded or validated, the process is abandoned and the file is discarded
2. Several errors exist which will prevent the overlay from being displayed
   1. When one of these errors occurs the user will be returned to the *Upload Author List* page so that a new file can be uploaded
   2. When an ALF is uploaded the file extension is checked against supported formats to ensure that the file is of an acceptable type (Supported types include \*.csv, \*.tab, \*.txt)
      1. If the file type does not adhere to these formats the file fails validation
      2. If this failure occurs the system will discard the file
      3. This failure triggers an alert message ending the validation process
         1. The message reads:

“**{FILENAME}** cannot be uploaded.   
  
The Author List File must be submitted in one of the following formats: \*.csv, \*.tab, \*.txt.   
  
Please upload another file.”

* + - 1. {FILENAME} is replaced with the name of the file
         1. The name of the file appears in bold text

**Figure 10:** Mocked up warning message for failure due to Author List File being uploaded in unsupported file extension. In this example the file failed because \*.doc is not a supported format.

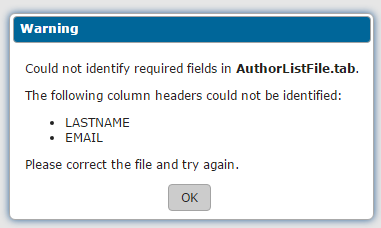


* 1. If the file is missing headers for any of the required fields (FIRSTNAME, LASTNAME, EMAIL) the file fails validation
     1. If this failure occurs the system will discard the file
     2. This failure triggers an alert message ending the validation process
        1. Message reads:   
           “Could not identify required fields in **{FILENAME}**. The following column headers could not be identified:
* {MISSING HEADERS}

Please correct the file and try again.

* + 1. {FILENAME} will display the name of the uploaded file
       1. The file name appears in bold text
    2. {MISSING HEADERS} displays a bulleted list of the required headers which were absent

**Figure 11:** Mocked up warning message for failure due to missing headers in the Author List File. In this example the file failed because the LASTNAME and EMAIL headers were not present in the file.

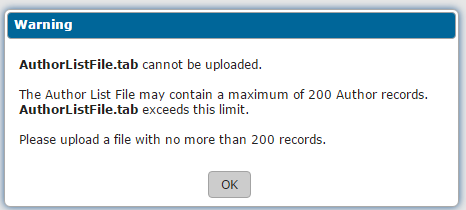


* 1. If INVITE, ORCID or NOTE headers are not found no error occurs as these fields are not required
     1. The system treats this absence as null values
  2. The number of rows are checked to make sure files above the enforced limit are not ingested
     1. Initial limit is 200 records [Pending Engineering testing/experiments during development]
     2. One row above the limit should always be allowed in order to account for the header row (e.g. 201 rows would be allowed if the final limit is 200)
     3. If the ALF file exceeds the limit an alert message ending the validation process is triggered
        1. The message reads:

“**{FILENAME}** cannot be uploaded.   
  
The Author List File may contain a maximum of 200 Author records. {FILENAME} exceeds this limit.

Please upload a file with no more than 200 records.”

* + - 1. {FILENAME} is replaced with the name of the file
         1. The name of the file appears in bold text

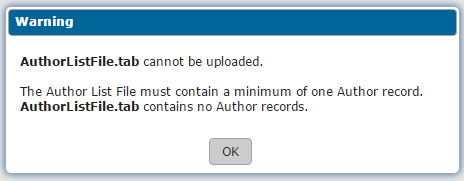
**Figure 12:** Mocked up warning message for failure due to Author record limit being exceeded in the Author List File.   
 

* 1. The ALF must contain a minimum of one Author record
     1. If the ALF file does not have any records in it (and thus fails to meet the minimum of one record) an alert message ending the validation process is triggered
        1. The message reads:

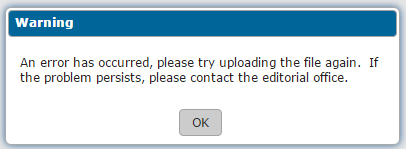
“**{FILENAME}** cannot be uploaded.   
  
The Author List File must contain a minimum of one Author record. {FILENAME} contains no Author records.”

* + 1. {FILENAME} is replaced with the name of the file
       1. The name of the file appears in bold text

**Figure 13:** Mocked up warning message for failure due to Author record minimum not being reached in the Author List File.



* 1. If there is an internal error that prevents the ALF from being uploaded, validated or ingested box is displayed with a message indicating there was a problem
     1. Message reads: “An error has occurred, please try uploading the file again. If the problem persists, please contact the editorial office.”
     2. When this error occurs the file is discarded

**Figure 14:** Mock-up of general error message to be used for internal errors with the Author List File.  


1. When the overlay is to be displayed, the contents of each record is validated
   1. Each row is checked to confirm the required data (FIRSTNAME, LASTNAME, EMAIL) is present (any order for the columns is supported)
      1. Individual records missing data will not prevent other, complete records from being ingested
         1. If individual rows do not meet the requirements of validation they are excluded from the ingest
         2. Correctly formatted rows will be ingested
         3. If FIRSTNAME, LASTNAME or EMAIL are missing for a record in the file the missing field is flagged with the  icon in overlay
      2. The EMAIL address value will be checked to confirm the address or addresses listed are valid email addresses
         1. The same check used when entering an email address in the interface will be used
            1. This check ensures that the @ symbol and a proper suffix are present (ex: WORDS@WORDS.WORDS, the system currently confirms the presence of the @ symbol and the period in the correct placements
         2. If the EMAIL value is not valid the row will be deemed improperly formatted and flagged with the  icon in the overlay
            1. A record with an invalid email address will not be ingested
      3. Records with bad or missing data will have the ignore option selected and disabled to prevent bad data from being used
      4. If all rows in the file are missing at least one of the required fields (FIRSTNAME, LASTNAME, EMAIL) the file cannot be imported
         1. The overlay will display with all records displayed and flagged appropriately
         2. The ‘Import’ button will be disabled so that the only option available to the user is to remove the file
2. If an ORCID iD has been provided in the optional ORCID field the ORCID must be validated in order to be included in the ingest of an Author List File
   1. ORCID iDs must be placed under the ORCID header
   2. The ORCID value is confirmed to be a 16-number digit
      1. The system will accept a 16-number digit as well as 16 digits grouped into 4-digit groups and separated by hyphens (i.e., formatted as either XXXXXXXXXXXXXXXX or XXXX-XXXX-XXXX-XXXX)
         1. The format of the record in the ALF is not the format the of ORCID stored in EM
         2. The file will be stored in EM as noted in Section 2.1.2 using the format used by existing ORCID functionality
      2. In addition to determining that the ORCID is well formed the checksum value will need to be confirmed as in existing functionality
         1. This validation was used in Spec 9.0-34B
   3. If the ORCID value cannot be validated it is excluded from the ingest and flagged with the  icon in the overlay
   4. If the provided ORCID iD is invalid the other fields in the record will still be ingested, without the ORCID iD value, because the ORCID iD is not required
3. Data found outside of the allowed columns is ignored and discarded

### Ingest of Author List File and Matching of Records to the EM Database

Once a file is uploaded and validated its usable contents is ingested into the system. The data is displayed in an ‘Author Candidates’ grid on the *‘Author Candidates’ Results* page once the overlay display of the ALF’s contents on the *Upload Author* List page is dismissed via the import option. Considerations must be taken when displaying the Author records in order to account for duplicate records within the Author List File as well as how Editorial Manager determines whether or not an ingested record matches an existing user record already in the database. The ‘Author Candidates’ grid populated by the ALF cannot be combined with other search results, Editors may search for Authors to invite individually before once the ALF has been used to send invitations.

The matching process begins once an Editor clicks the ‘Import’ button found on the overlay. A loading message displays in an overlay while the matching is in progress before the user is directed to the *‘Author Candidates’ Results* page (See Section 2.3). When attempting to match an Author record from the Author List File to an existing EM record several checks are performed in order to locate a unique record where possible (See Table 1 for order of priority). If the first check fails to identify a single, unique EM people record to associate with the Author record from the ALF the matches which have been found (if any) are run through the subsequent checks until an unique EM people record is identified. If multiple matches remain once all the checks are completed all matching records will be displayed in the 'Author Candidates' grid, flagged as possible duplicates and without the select box checked off by default, so that Editors can choose which if any will be included. If no matches are found at all the ALF record will be displayed and an account will be proxy registered once the Editor proceeds to customize invitation letters.

1. When Author records are ingested from an Author List File the system considers the contents of the list prior to displaying the Authors
   1. The ALF is checked for multiple entries for the same record
      1. These duplicate entries are determined to be duplicates if the FIRSTNAME, LASTNAME and EMAIL match other existing records
         1. If there is a partial match the records are displayed as separate entities in the ‘Author Candidates’ grid as detailed in Section 2.4
      2. If the optional ORCID field is populated it must match as well
      3. The NOTE field is able to vary between records without impacting the determination of duplicate entries
         1. All NOTE values will be displayed in the Author Name column of the ‘Author Candidates’ grid (See Section 2.4 , item 2)k)
   2. If multiple entries for the same record exist a count is taken
      1. This count becomes the Number of Invitation Letters value in the ‘Author Candidates’ grid
      2. If an author appeared *n* times in the ALF, they would appear only once in the ‘Author Candidates’ grid with Number of Invitation Letters set to *n*
      3. If between these matched records the INVITE value varies, the value used by a majority of the records is applied, if no one value is in the majority the record will be treated as though INVITE = NULL
   3. If there is an exact match on EMAIL between records in the ALF but the FIRSTNAME and LASTNAME values in each record do not match the respective values in the other records, they are treated as separate records when the system attempts to match the ALF records against existing EM records (See item 2)e)
      1. If multiple emails are included in the ALF record only one of them must be found in another record to meet this condition
2. The system attempts to match each Author record from the ALF to an existing, registered EM user
   1. Two determinations are possible – Confirmed Match and No Match.
   2. Confirmed matches are determined using email address and ORCID and subsequently use the first and last names if a unique record cannot be found
      1. The system starts with the highest priority match condition and works its way through all the conditions attempting to find a match
      2. If multiple matches are found at a high level priority, only those matches rather than the entire database are run through the subsequent checks in an effort to bring the potential matches down to a single unique record
   3. Matching occurs as follows:
      1. If an ORCID is present in the ALF and matches an Authenticated ORCID associated with an EM people record the ALF record is matched to the EM record
         1. ORCID is found in EM in the Personal Identifiers Table (PERSONAL\_IDENTIFIERS.ORCID)
         2. ORCID is Authenticated if PERSONAL\_IDENTIFIERS.ORCID\_IMPORT\_AUTHENTICATED = 1
      2. If an ORCID is present in the ALF and matches an unauthenticated ORCID associated with an EM people record, then the PEOPLE.LASTNAME value is checked against the ALF LASTNAME value
         1. If PEOPLE.LASTNAME matches the ALF LASTNAME value the ALF record is matched to the EM record
            1. If multiple matches are found an additional check using FIRSTNAME is performed
            2. If PEOPLE.FIRSTNAME matches as well and the result is a single unique EM record match the ALF record is matched to the EM record
         2. ORCID is unauthenticated if PERSONAL\_IDENTIFIERS.ORCID\_IMPORT\_AUTHENTICATED = 0
      3. If no ORCID value is present in the ALF, ALF EMAIL is used to determine a Confirmed Match
         1. If the ALF EMAIL value matches only a primary ADDRESS record for a single EM people record, a Confirmed Match is determined
            1. Primary ADDRESS record is determined by ADDRESS.PRIMARY = 1
         2. If the ALF EMAIL value matches a primary ADDRESS.EMAIL for more than one EM people record, additional checks are performed
            1. The ALF LASTNAME value is checked against the PEOPLE.LASTNAME value associated with the EM user’s ADDRESS.EMAIL

If there is a single exact match that EM record becomes the Confirmed Match

If multiple matches remain the ALF FIRSTNAME is checked against PEOPLE.FIRSTNAME

If there is a single match remaining after this additional comparison, that EM record becomes the Confirmed Match

* + - 1. If the ALF EMAIL value does not match a primary ADDRESS record, alternate ADDRESS records are checked
         1. Alternate ADDRESS record is determined by ADDRESS.PRIMARY = 0
         2. If an alternate ADDRESS record is found where the ADDRESS.EMAIL value and the ALF EMAIL value , the ALF LASTNAME is checked against PEOPLE.LASTNAME
         3. If both values are an exact match the EM record becomes the Confirmed Match

If multiple matches remain the ALF FIRSTNAME is checked against PEOPLE.FIRSTNAME

If there is a single match remaining after this additional comparison, that EM record becomes the Confirmed Match

* + - 1. If multiple email addresses exist in either the ADDRESS.EMAIL or ALF record each individual address is parsed out and checked independently
         1. Any match is sufficient for confirmation
         2. Example: ALF EMAIL value is [smith@email.com;smith@email.edu](mailto:smith@email.com;smith@email.edu) and ADDRESS.EMAIL is [smith@email.edu](mailto:smith@email.edu)

This is a confirmed match because [smith@email.edu](mailto:smith@email.edu) is present in both values

* + - * 1. Example: ALF EMAIL value is [jones@email.com](mailto:jones@email.com;jones@email.edu) and ADDRESS.EMAIL is [jones@email.com;jones@email.edu](mailto:jones@email.com;jones@email.edu)

This is a confirmed match because smith@email.com is present in both values.

* + 1. If after all checks against ORCID and EMAIL are completed multiple EM accounts match a record from the ALF all matching EM accounts are added to the ‘Author Candidates’ grid on the *‘Author Candidates’ Results* page (See Section 2.4)
       1. Red warning text will be displayed in the Author Name column of the ‘Author Candidates’ grid indicating that there are duplicate EM accounts
       2. These additions to the ‘Author Candidates’ grid will not be automatically selected for invite
          1. Editors will determine which match to invite with their manual selection using the information presented via the ‘Author Candidates’ grid (See Section 2.4)

An Editor should be able to determine which duplicate account is actively in use using the available statistics and rule out any errors by reviewing the information from the ALF that is presented

* + - 1. The Number of Invitation Letters count will be applied to all matching EM records
         1. Editors can adjust this value on the ‘Author Candidates’ grid manually (See Section 2.4)
      2. NOTE fields from all matching ALF records will be displayed for all matching EM records
         1. If multiple NOTE values are entered they are truncated and concatenated into a single cell for display on the ‘Author Candidates’ grid (See Section 2.4)
  1. If the previous logic does not find a match, a ‘No Match’ determination is reached
     1. Authors with no match to existing EM accounts will have accounts proxy-registered when the Editor clicks the proceed button to move on to the Select Authors – Confirm Selection and Customize Letters page [confirmAndCustomizeInvitedAuthors.aspx]
        1. Proxy-registration process occurs behind the scenes similar to the proxy-registration process used for Reviewer Discovery (See Section 2.4)
        2. The data carried forward during proxy-registration is detailed in Section 2.1.2
        3. If a record displayed in the ‘Author Candidates’ grid is not selected it will not be proxy-registered
  2. If two records from the ALF have the same email address, but have been treated as separate records because other data does not match as noted under item 1)b.iii in this section and both records have been given a No Match determination both records will be added to the ‘Author Candidates’ grid
     + 1. Red warning text will be displayed in the ‘Author Name’ column of the ‘Author Candidates’ grid indicating that there are duplicate ALF records with the same email address (See Section 2.4)
       2. These additions to the ‘Author Candidates’ grid will not be automatically selected for invite
          1. Editors will determine which if any record should be used and proxy-registered

1. Matches to EM records exclude inactive and merged users
   1. No warning or indication is provided that an account being created through proxy-registration may match a previous inactivated or merged record
2. When a match occurs the Board Member, Submissions Being Processed, Author Statistics, and Invitation Statistics fields in the ‘Author Candidates’ grid populate with data from the matching user’s record
   1. This is the same information displayed when an Author is found through the existing searching functionality
   2. The new functionality adds information to the Author Name column of the grid (See Section 2.4, 2)k)
3. The uploaded file and ingested information is discarded once invitations have been sent to the Authors

**Table 1:**  This table lists ways data from the Author List File is matched to the EM database in order to identify existing records for the people contained in the list. The various procedures are listed in their intended order of operations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Priority/** | **Match Type** | **Match Criteria** | | **Notes** |
| **Confidence** |  | **ALF field(s)** | **EM Field(s)** |  |
| 1 | Authenticated ORCID | ORCID | PERSONAL\_IDENTIFIERS.ORCID PERSONAL\_IDENTIFIERS.ORCID\_IMPORT\_AUTHENTICATED | If the ORCID is present in the ALF, EM checks for a matching ORCID in the Personal Identifiers Table. If the matching ORCID is Authenticated (PERSONAL\_IDENTIFIERS.ORCID\_IMPORT\_AUTHENTICATED=1) the corresponding EM people record is considered a match. |
| 2 | Non-Authenticated ORCID + Last Name | ORCID, LASTNAME | PERSONAL\_IDENTIFIERS.ORCID, PERSONAL\_IDENTIFIERS.ORCID\_IMPORT\_AUTHENTICATED, PEOPLE.LASTNAME | If the ORCID is present in the ALF, EM checks for a matching ORCID in the Personal Identifiers Table. If the matching ORCID is Non-Authenticated (PERSONAL\_IDENTIFIERS.ORCID\_IMPORT\_AUTHENTICATED=0) the LASTNAME value is checked against PEOPLE.LASTNAME if a unique match is found the corresponding EM people record is considered a match. |
| 3 | Non-Authenticated ORCID + Last Name + First Name | ORCID, LASTNAME, FIRSTNAME | PERSONAL\_IDENTIFIERS.ORCID, PERSONAL\_IDENTIFIERS.ORCID\_IMPORT\_AUTHENTICATED, PEOPLE.LASTNAME, PEOPLE.FIRSTNAME | If the ORCID is present in the ALF, EM checks for a matching ORCID in the Personal Identifiers Table. If the matching ORCID is Non-Authenticated (PERSONAL\_IDENTIFIERS.ORCID\_IMPORT\_AUTHENTICATED=0) the LASTNAME value is checked against PEOPLE.LASTNAME. If a unique match is still not identified the FIRSTNAME value is checked against PEOPLE.FIRSTNAME. If a unique match is found the corresponding EM people record is considered a match. |
| 4 | Primary Email | EMAIL | ADDRESS.EMAIL, ADDRESS.PRIMARY | If no matching ORCID value is found, or no ORCID value is present in the ALF, the EMAIL value is checked against emails stored in the Address Table. If EMAIL uniquely matches an ADDRESS.EMAIL record which is also marked as the primary address (ADDRESS.PRIMARY=1) then the corresponding EM people record is considered a match. |
| 5 | Primary Email + Last Name | EMAIL, LASTNAME | ADDRESS.EMAIL, ADDRESS.PRIMARY, PEOPLE.LASTNAME | If multiple matches are found on EMAIL against the primary email addresses in the Address Table, the system also runs a check on LASTNAME. If matching on EMAIL and LASTNAME results in a unique result the corresponding EM people record is considered a match. |
| 6 | Primary Email + Last Name + First Name | EMAIL, LASTNAME, FIRSTNAME | ADDRESS.EMAIL, ADDRESS.PRIMARY, PEOPLE.LASTNAME, PEOPLE.FIRSTNAME | If multiple matches are found on EMAIL and LASTNAME against the primary email addresses in the Address Table and People Table, the system also runs a check on FIRSTNAME. If matching on EMAIL, LASTNAME, and FIRSTNAME results in a unique result the corresponding EM people record is considered a match. |
| 7 | Alternate Email + Last Name | EMAIL, LASTNAME | ADDRESS.EMAIL, ADDRESS.PRIMARY, PEOPLE.LASTNAME | If a unique match has not been found by matching on ORCID or primary Email the attempts to match on alternate emails (ADDRESS.PRIMARY=0). An attempt is made to match the ALF record to an EM record using EMAIL and LASTNAME where the email is listed as an alternate address in the ADDRESS table. If a unique result is found the corresponding EM people record is considered a match |
| 8 | Alternate Email + Last Name + First Name | EMAIL, LASTNAME, FIRSTNAME | ADDRESS.EMAIL, ADDRESS.PRIMARY, PEOPLE.LASTNAME, PEOPLE.FIRSTNAME | If a unique match has not been found by matching on EMAIL and LASTNAME while looking at alternate emails (ADDRESS.PRIMARY=0). An attempt is made to match the ALF record to an EM record using EMAIL, LASTNAME, and FIRSTNAME where the email is listed as an alternate address in the ADDRESS table. If a unique result is found the corresponding EM people record is considered a match |
| 9 | Multiple Matches found | N/A | N/A | If after all attempts to locate a unique EM record have been made multiple matches still exist all matching records will be displayed in the 'Author Candidates' grid flagged as possible duplicates without the select box being checked out so that Editors can choose which if any will be included |

## Author Selection Summary page [authorSelectionSummary.aspx]

The *Author Selection Summary* page is available to Editors working with Proposals and Commentaries by clicking the *Solicit Commentary* Action Link, clicking the *Invite Authors* Action Link for an existing Proposal, or choosing the ‘Assign to Myself and Invite Authors’ option when creating a Proposal. These options are available when an Editor has the Invite Authors/Solicit Commentaries permission.

Results from different Author search and selection methods are not combined in the existing functionality and selections of authors across different search methods are not carried over when switching from one option to another. This will not be enhanced in this specification. If the Editor switches to another Author search method after uploading, validating and ingesting an Author List File the file and uploaded information are discarded. Likewise if an Editor had been searching for Authors prior to deciding to upload a file any previous selections would be discarded.

**NOTE:** Currently, Editorial Manager does not support the direct assignment of Authors. Spec 13.0-33 (tentatively intended for 13.1) will address the assignment of Authors and will account for how that functionality should impact the Author List File and its interactions with the *Author* *Selection Summary* page.

**Current Functionality:**

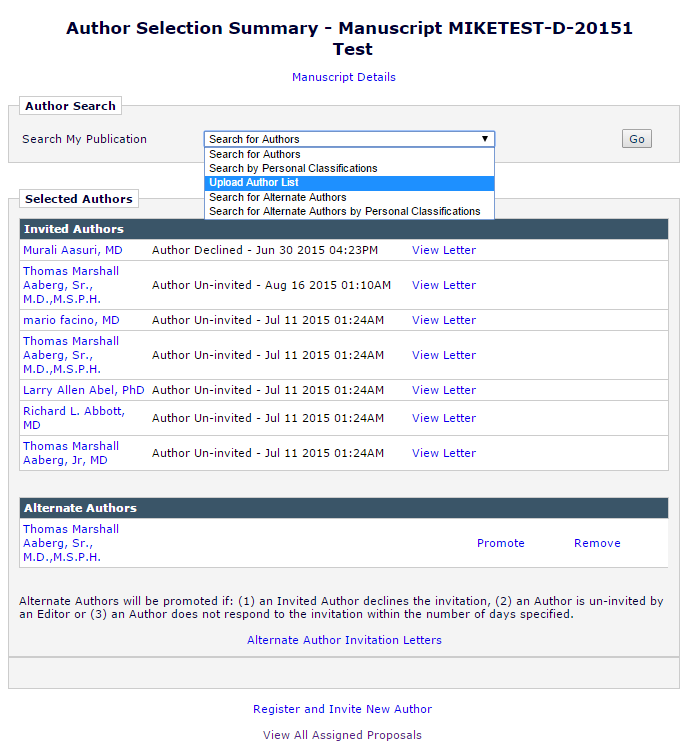
1. Editors must search for Authors to invite them to submit using the ‘Author Search’ options on the *Author Selection Summary* page
   1. Editors have four options for searching in the Author Search drop-down menu
      1. ‘Search for Authors’
      2. ‘Search by Personal Classifications’
      3. ‘Search for Alternate Authors’
      4. ‘Search for Alternate Authors by Classifications’
   2. Alternate authors must be selected separately from invited authors
   3. Alternate authors are not linked to specific (primary) authors
      1. This functionality is not enhanced in this specification
   4. Selections made before switching to another search method are not retained
      1. This functionality is not enhanced in this specification
2. If an Editor cannot find a desired Author in the system they may choose to proxy-register a new user from the *Author Selection Summary* page or from Author search results

**New functionality:**

In order to accommodate the new functionality introduced by this specification the layout and functionality of the *Author Selection Summary* page is enhanced.

1. The Author Search drop-down menu on the *Author Selection Summary* page is enhanced with a new search method:
   1. ‘Upload Author List’
      1. Placed in drop-down menu after the ‘Search by Personal Classifications’ option
      2. Will allow an Author List File to be uploaded and the Authors ingested from that file to be invited to submit manuscripts
   2. When this new option is selected an  icon is placed to the right of the drop menu
   3. When switching to any of the existing options from the new Upload Author List option the  icon is no longer displayed placed to the right of the drop menu
      1. Any Author selections made prior to switch between search options are not retained

**Figure 15:** Mockup of enhanced Author Search drop-down menu showing the new Upload Author List option.

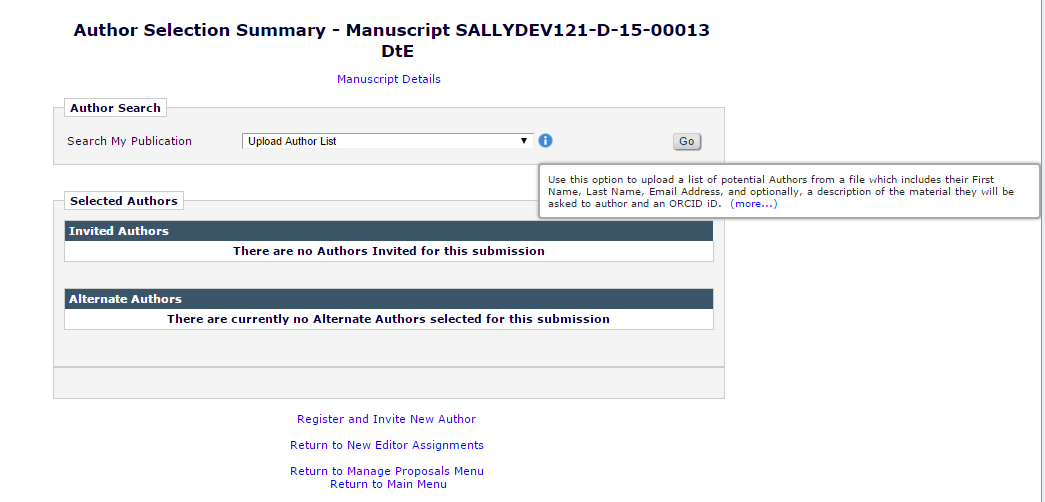


**Figure 16:** Mockup of enhanced *Author Selection Summary* page once user selects the new Upload Author List option. The ‘I’ icon will produce a tool tip message leading to information about how the file to be uploaded should be formatted. (Tool tip shown in Figure 18)



1. When the  icon is displayed:
   1. Hovering over the icon with the mouse cursor produces a tool tip box
   2. The tool tip remains displayed while the mouse cursor hovers over the icon or the tool tip box
   3. The tool tip box can be expanded and collapsed by a user and initially appears in its collapsed form
   4. Tool tip box explains the purpose of the option and can be expanded to display a more complete explanation of the process
      1. Text of tooltip: Use this option to upload a list of potential Authors from a file which includes their First Name, Last Name, Email Address, and optionally, a description of the material they will be asked to author and an ORCID iD.

**Figure 17:** Mocked up tool tip box which appears when the mouse cursor rolls over the  icon. This represents the collapsed version of the tool tip.



* 1. When the ‘more’ link is clicked the tool-tip box expands, showing detailed information
     1. Text of expanded tool tip:

Use this option to upload a list of potential Authors from a file which includes their First Name, Last Name, Email Address, and optionally, a description of the material they will be asked to author and an ORCID iD.

To upload a file of people to invite as Authors, you can either click the 'Choose File' button to browse your computer, or drag and drop a file onto the folder image below. Only a single file may be selected at a time.

**Caution:** The file must be in one of the following formats:

* Comma Separated Value (\*.csv, \*.txt)
* Tab Delimited File (\*.tab, \*.txt)

The first row must be a header with three to six columns:

Required:

* FIRSTNAME
* LASTNAME
* EMAIL

Optional:

* INVITE
* NOTE
* ORCID

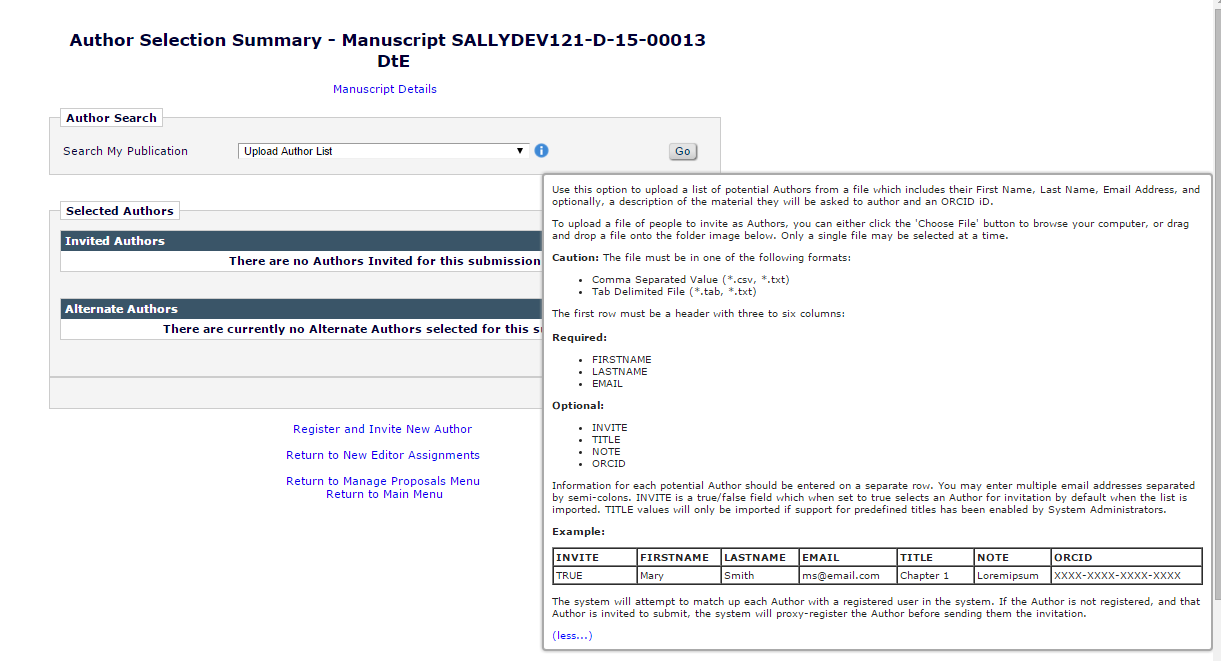
Information for each potential Author should be entered on a separate row. You may enter multiple email addresses separated by semi-colons. INVITE is a true/false field which when set to true selects an Author for invitation by default when the list is imported.

**Example:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **INVITE** | **FIRSTNAME** | **LASTNAME** | **EMAIL** | **NOTE** | **ORCID** |
| TRUE | Mary | Smith | [ms@email.com](mailto:ms@email.com) | Chapter 1 | XXXX-XXXX-XXXX-XXXX |

The system will attempt to match up each Author with a registered user in the system. If the Author is not already registered in the system, and that Author is invited to submit, the system wills proxy-register the Author before sending their invitation.

* 1. Clicking the ‘less’ link will restore the tool tip box to its original, collapsed form

**Figure 18:** Mocked up tool tip box which appears when the mouse cursor rolls over the  icon and a user clicks the ‘more’ link to expand the box. 

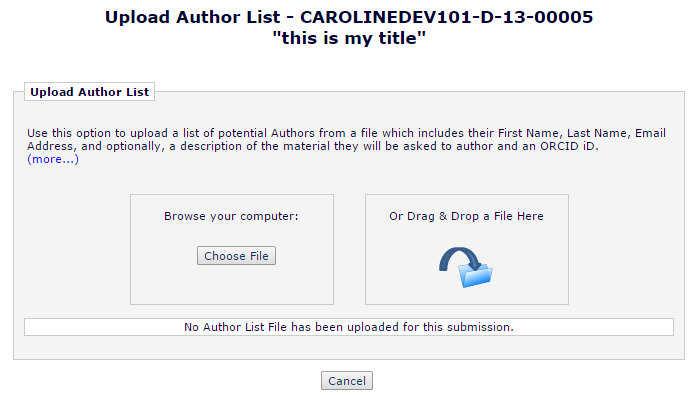
* 1. When the ‘Upload Author List’ option is selected the ‘Go’ button may be clicked to proceed the user to a new page where they will be able to upload the Author List (See Section 2.3)
     1. This is consistent with the existing functionality where a user must select a search method and then click ‘Go’ in order to proceed to the relevant search page

## Upload Author List page [new page]

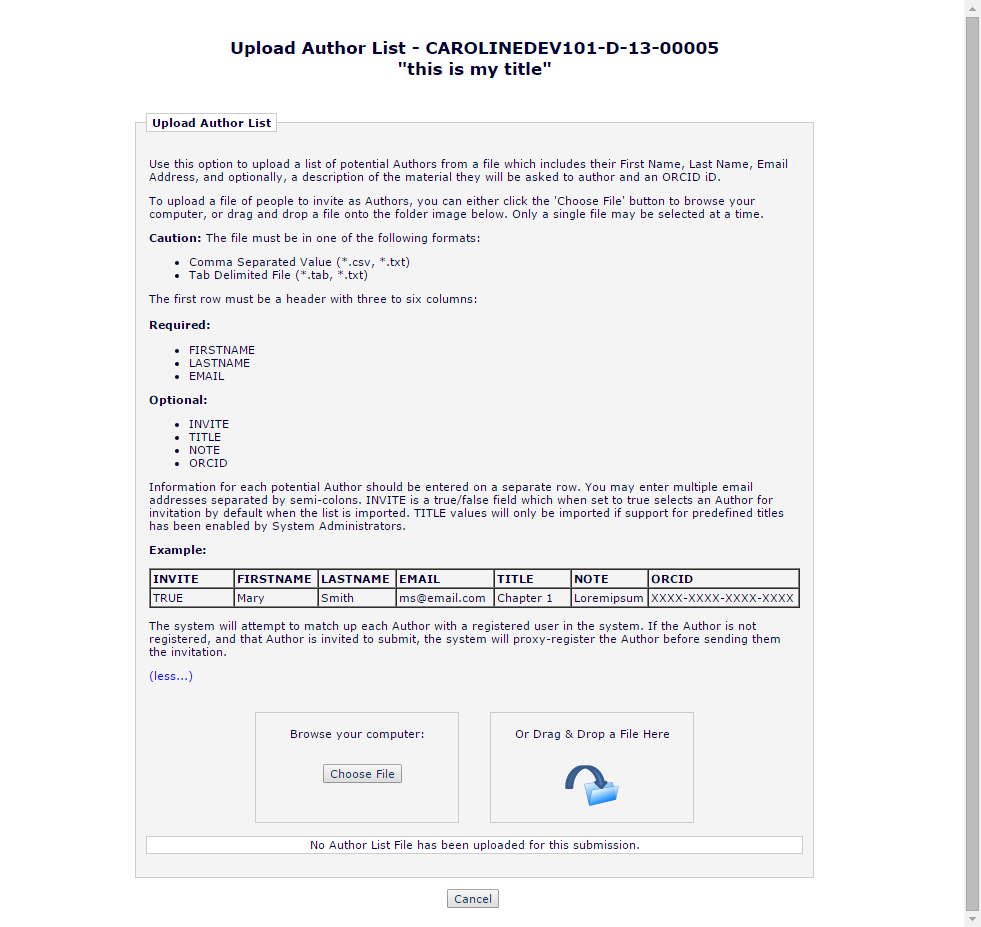
The existing Author Search options direct a user to different search pages depending on the method of search that is selected. When ‘Search for Authors’ is selected users are directed to searchAuthorsByNameAddr.asp and when ‘Search by Personal Classifications’ is used, users arrive at searchAuthorsByPersonalClassifications.aspx. To facilitate the uploading of the Author List File a new page is created to serve as a bridge between the *Author Selection Summary* page and the new *‘Author Candidates’ Results* page showing a full vetted ‘Author Candidates’ grid.

1. New page will be called the *Upload Author List* page
2. The *Upload Author List* page is accessed when a user selects ‘Upload Author List’ from the Author Search drop down on the *Author Selection Summary* page and then clicks the ‘Go’ button
3. The *Upload Author List* page allows the uploading of an Author List File
   1. Users will be able to browse for a file locally
   2. Users will be able to drag and drop a file onto to the page
   3. Only one ALF can be uploaded per invitation session
4. The *Upload Author List* page will be laid out as follows:
   1. Page header will Display “Upload Author List”
   2. Heading on the page will display ‘Author List File – ’ followed by the standard metadata about a submission displayed in a header (including Manuscript Number and Article Title)
   3. An ‘Upload Author List’ grid is placed on the page
      1. Instructions detailing the proper formatting for an Author List File are placed within the grid
      2. The text used is the same text used for the ‘Upload Author List’ tool tip from the *Author Selection Summary* page
         1. The instructional text is expandable/collapsible using more/less links
         2. The text is divided between more/less in the same manner as the tool tip
   4. Below the instructions two boxes are added side by side for the two upload options
   5. The left-side box contains the option to choose a local file to upload
      1. A ‘Choose File’ button is placed within the box which will allow the user to browse their computer for the Author List File
      2. Text above the button reads “Browse your computer:”
      3. When the ‘Choose File’ button is clicked users are prompted to browse their computer to select a file
         1. Only one file may be selected for upload per invitation session
         2. During and after the upload and validation of a file the Choose File button is disabled and greyed out
            1. Following the upload a file can be removed by the Editor
            2. If the file is removed the button will become available for use again
   6. The right-side box contains the drag and drop option
      1. A drag and drop area is established by the box so that a file may be uploaded using drag and drop (**Note:** drag and drop functionality was introduced by 11.0-01, existing code should be leveraged)
      2. Only one file may be selected for upload per invitation session
         1. If multiple files are dragged and dropped the file which would be considered first in the existing drag and drop functionality begins to upload, other files are ignored
      3. During and after the upload and validation of a file the drag and drop option becomes disabled and greyed out
         * 1. Following the upload a file can be removed by the Editor
           2. If the file is removed the option will become available for use again
   7. Below the upload options is a space to show information about the file
      1. When no file has been uploaded the space contains a message reading, “No Author List File has been uploaded for this submission.” (See Figure 19)
      2. When a file has been uploaded the section is labeled ‘File in Progress’
         1. Status information about the upload and validation of the file is displayed in this space
   8. A ‘Cancel’ button is present at the bottom of the page
      1. Clicking ‘Cancel’ will return a user to the *Author Selection Summary* page

**Figure 19:** Mock-up of *Upload Author List* page prior to a file being uploaded and with the instructions collapsed.

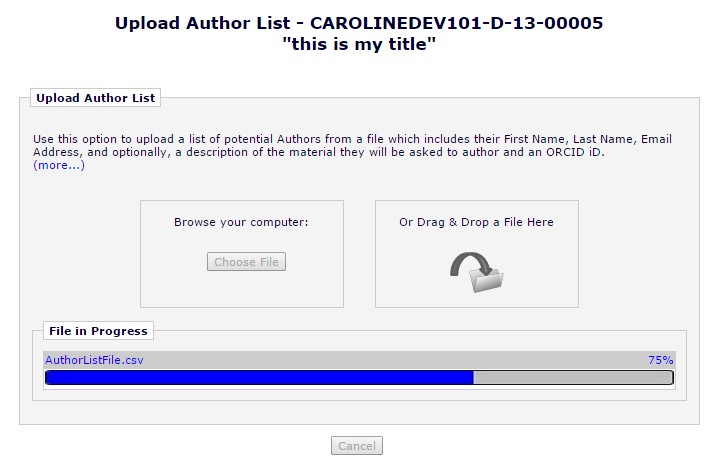


**Figure 20:** Mock-up of *Upload Author List* page prior to a file being uploaded and with the instructions expanded.



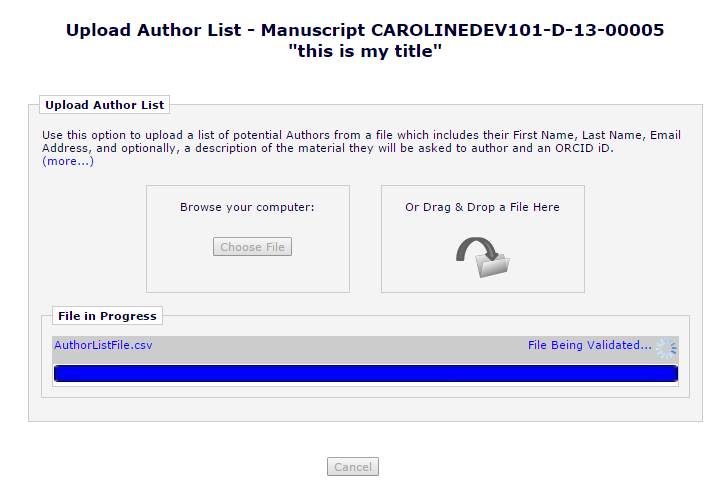
1. When file is selected or dropped on the page, the upload and validation process begins immediately
   1. The system validates the file’s file type
   2. File will not be uploaded if it is of an incorrect type (See Section 2.1.3 for validation information)
2. Upload and validation progress is displayed within the new ‘File in Progress’ section of the page
   1. The ‘File in Progress’ section displays information about the uploaded ALF
      1. When no file has been uploaded the space contains a message reading, “No Author List File has been uploaded for this submission.” (See Figure 19)
      2. When a file has been uploaded the section is labeled ‘File in Progress’
         1. While the ALF is uploading:
            1. The name of the file is displayed opposite the completion percentage as a numerical value
            2. A progress bar is displayed below the name to indicate the percentage of the file upload completed, similar to what exists currently when attaching items for submission
            3. The file upload options are disabled
            4. The ‘Cancel’ button is disabled (See Figure 21)

**Figure 21:** Mocked-up *Upload Author List* page showing a file upload in progress.



* + - 1. Once the file reaches 100 percent uploaded, the validation process begins (See Section 2.1.3 for validation process logic)
         1. While validation is in progress a blue spinner appears where the completion percentage for the upload was displayed (See Figure 22)
         2. The file upload options are disabled
         3. The ‘Cancel’ button is disabled
         4. Once validation is completed, the spinner and progress bar disappear

**Figure 22:**  Mocked-up *Upload Author List* page showing a file validation in progress.

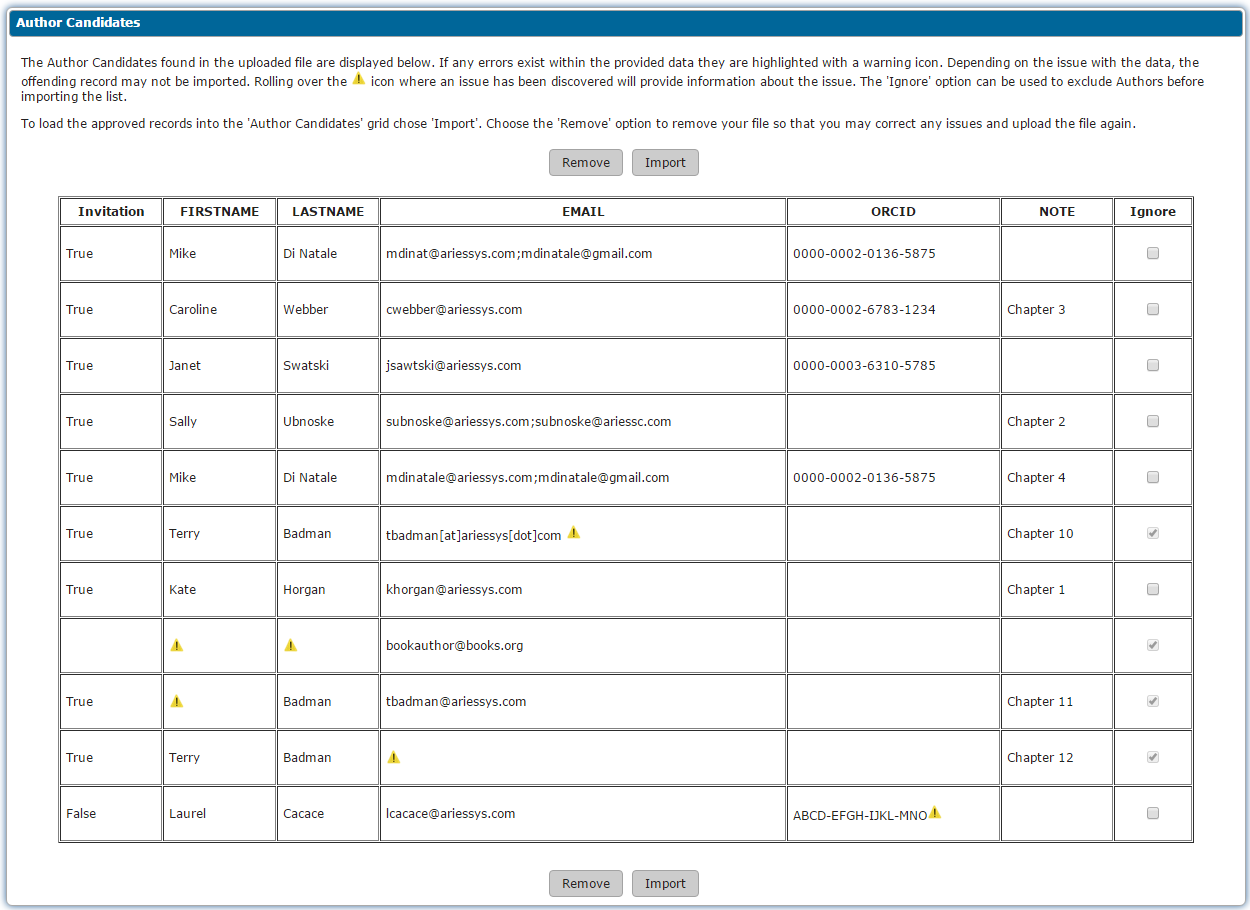


1. If an uploaded file contains any usable data, following validation an overlay appears on the screen (See Section 2.1.3)
   1. The overlay contains text explaining its purpose
   2. All records in the file will be displayed in a grid inside the overlay
   3. Fields which have not passed validation will be flagged with an  icon displayed to the right of any text entered in the field
      1. Hovering over the icon with the cursor will produce a tool tip explaining the issue with the field
      2. The text of these tool tips is detailed in Section 2.1.3
   4. An ‘Ignore’ column will appear on the far right of the grid
      1. Each entry in the ‘Ignore’ column for a record will contain a check box
      2. The checkbox is checked off and disabled for any record missing required data
         1. Records with missing required data are not ingested
      3. Users will be able to check the ignore box to prevent a record from added to the ‘Author Candidates’ grid if they have passed validation
   5. There are two buttons used to dismiss the overlay which appear at the top and bottom of the overlay
      1. A ‘Remove’ button which removes the file so that an Editor can uploaded another file in order to correct the discovered errors
         1. Corrections to the data in the original file would occur outside the system
            1. A warning message with ‘Cancel’ and ‘OK’ options triggers when the ‘Remove’ button is clicked

Message reads: “This will remove the uploaded file and discard any existing selections. Are you sure?”

* + 1. A ‘Import’ button which proceeds with the ingest, including only files which passed validation and have not been marked by an Editor to be ignored
       1. Clicking ‘Import’ will move the user away from the new *Upload Author List* page and direct the user to the new *‘Author Candidates’ Results* page (See Section 2.4)
          1. Prior to moving a user to the *‘Author Candidates’ Results* page the matching of Author List File records to existing EM records occurs once the ‘Import’ button is clicked
          2. While EM is matching records from the ALF to the EM database an overlay is displayed indicating that records are being processed (See Figure 24)
          3. Once the matching is complete, users are moved to the *‘Author Candidates’ Results* page where the records from the ALF are displayed in the ‘Author Candidates’ grid

**Figure 23:** Mockup of the overlay which would appear over the *Author Selection Summary* page at the end of the validation process in order to highlight any issues with the uploaded Author List File.



**Figure 24:** Mockup of overlay displayed while EM matches records from the ALF against the existing EM database.



## ‘Author Candidates’ Results page [UploadedAuthorSearchResults.aspx]

In the existing functionality an ‘Author Candidates’ grid is displayed on one of two the possible search results pages displayed to users when searching for Authors. The page users arrive at depends on whether a search by name or search by classification match was used by the Editor. These pages are reused by Alternate Author searches. Two these destination pages for the results are searchAuthorsByNameAddrResults.asp and searchAuthorsByPersonalClassResults.asp.

To remain consistent with the existing functionality, a new results page will be created in order to display showing a fully vetted and validated ‘Author Candidates’ grid. This new results page will remain nearly identical to the existing results pages, but supress the searching options and display additional information in the Author Name column depending upon the data found in the Author List File and the system’s ability to match Authors to existing EM records.

Users will arrive on this new results page after clicking the ‘Import’ button within the ‘Author Candidates’ overlay that is displayed on the *Upload Author List* page.

1. The *‘Author Candidates’ Results* page [UploadedAuthorSearchResults.aspx] is displayed once an Editor chooses to import an uploaded Author List file from the *Upload Author List* page.
   1. The new page opens in the same window which previously displayed the *Upload Author List* page
2. The layout of the *‘Author Candidates’ Results* page is similar
   1. A Heading appears at the top of the page identifying the page as ‘Author Candidates’ Results and containing the standard submission metadata including Manuscript Number and Article Title
   2. The search options found on the existing results pages are removed
   3. A new section containing information about the uploaded Author List File is added below the heading
      1. The section is labeled ‘Author List File’
         1. The file is displayed in a grid with four headers
         2. Headers use white text and EM blue for a background color
            1. File Name

Displays the file name and extension of uploaded ALF

* + - * 1. Size

Displays the size of the uploaded file in a manner consistent with the size displayed when uploading a submission item

* + - * 1. Last Modified Date

Displays the last modified date of an uploaded file in a manner consistent with the last modified date displayed when uploading a submission item, formatted based on the publication’s configured date format.

* + - * 1. Actions

A ‘Remove’ option is available under ‘Actions’ to remove the uploaded ALF

Removing the ALF will return a user to the *Upload Author File* page and remove the file from the system

The user will then be able to upload a new Author List File

Any Author selections made prior to removing the ALF will be lost

A warning message with ‘Cancel’ and ‘OK’ options triggers when the ‘Remove’ button is clicked

Message reads: “This will remove the uploaded file and discard any existing selections. Are you sure?”

Clicking ‘OK’ proceeds with the removal of the file

Clicking ‘Cancel’ leaves the file in place

* + 1. If the file has been ingested with known errors, an  icon will display to the right of the file name to indicate that there were issues with the uploaded and ingested file (See Figure 25)
       1. Rolling over the  icon will provide an explanation that there were errors (See Figure 25)
          1. Tool tip reads: “This file contains incomplete or invalid records. Click for more information.”
       2. Clicking icon will display an overlay detailing the issues which had been found in the record (See Figure 26)
          1. This overlay is separate and different, but similar to what was displayed following validation on the *Upload Author List* page

Ignore option is disabled because the ALF has already been ingested

Import button is replaced with ‘OK’

‘Clicking ‘OK’ will dismiss the overlay

Remove remains present

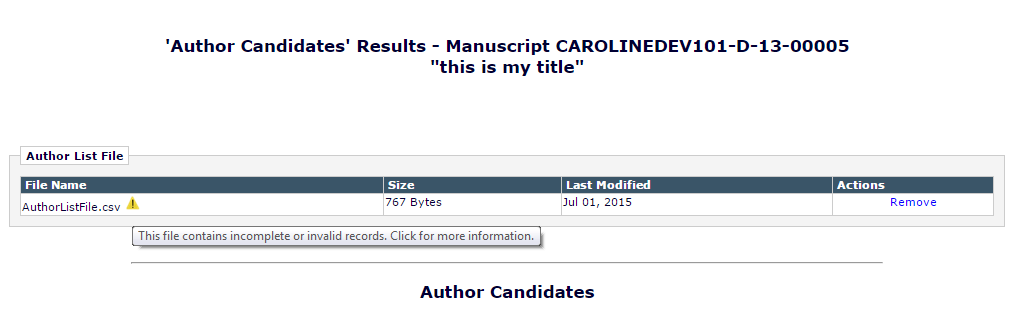
Clicking ‘Remove’ will remove the ALF file so that the Editor can start the process over

A warning message with ‘Cancel’ and ‘OK’ options triggers when the ‘Remove’ button is clicked

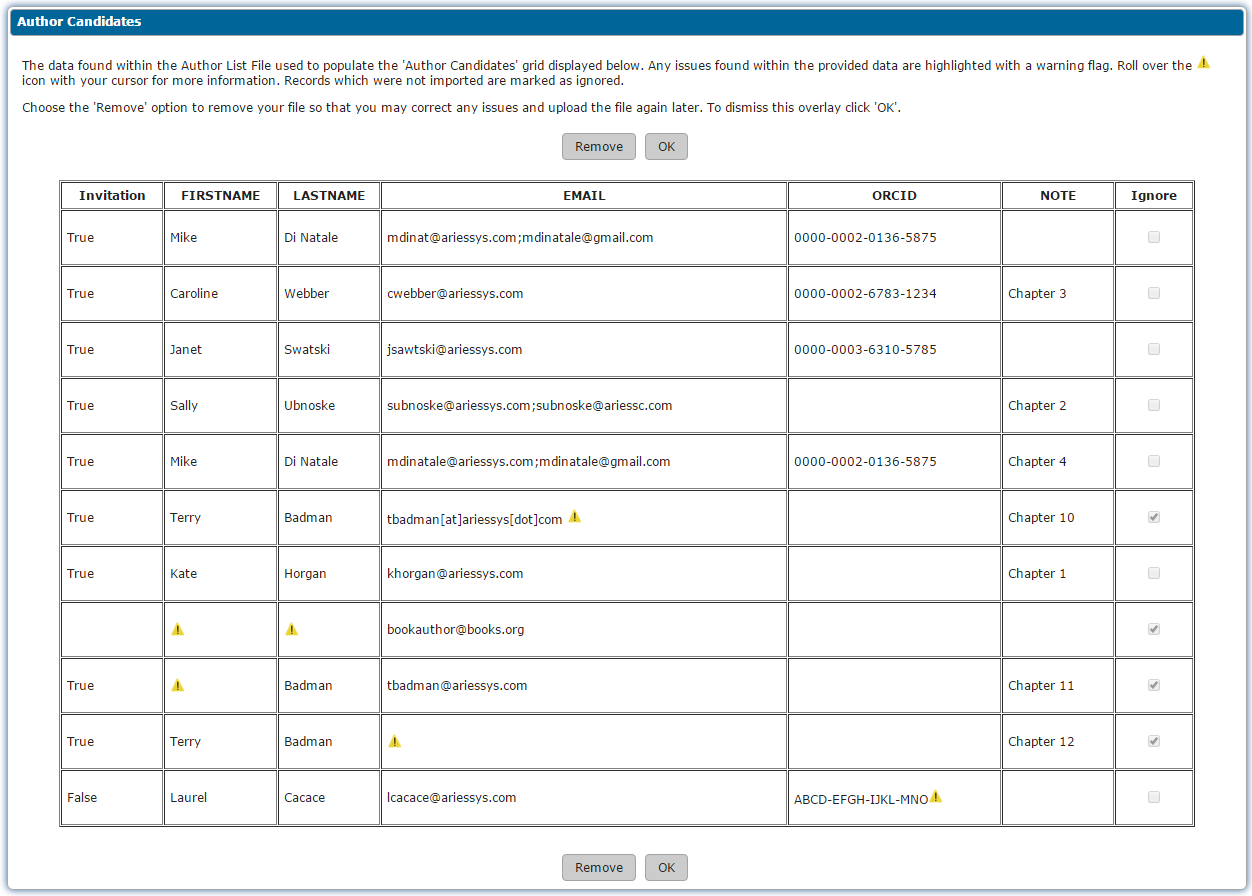
Message reads: “This will remove the uploaded file and discard any existing selections. Are you sure?”

The instructional text is changed to: “The data found within the Author List File used to populate the 'Author Candidates' grid displayed below. Any issues found within the provided data are highlighted with a warning flag. Roll over the icon with your cursor for more information. Records which were not imported are marked as ignored.

Choose the 'Remove' option to remove your file so that you may correct any issues and upload the file again later. To dismiss this overlay click 'OK'.”

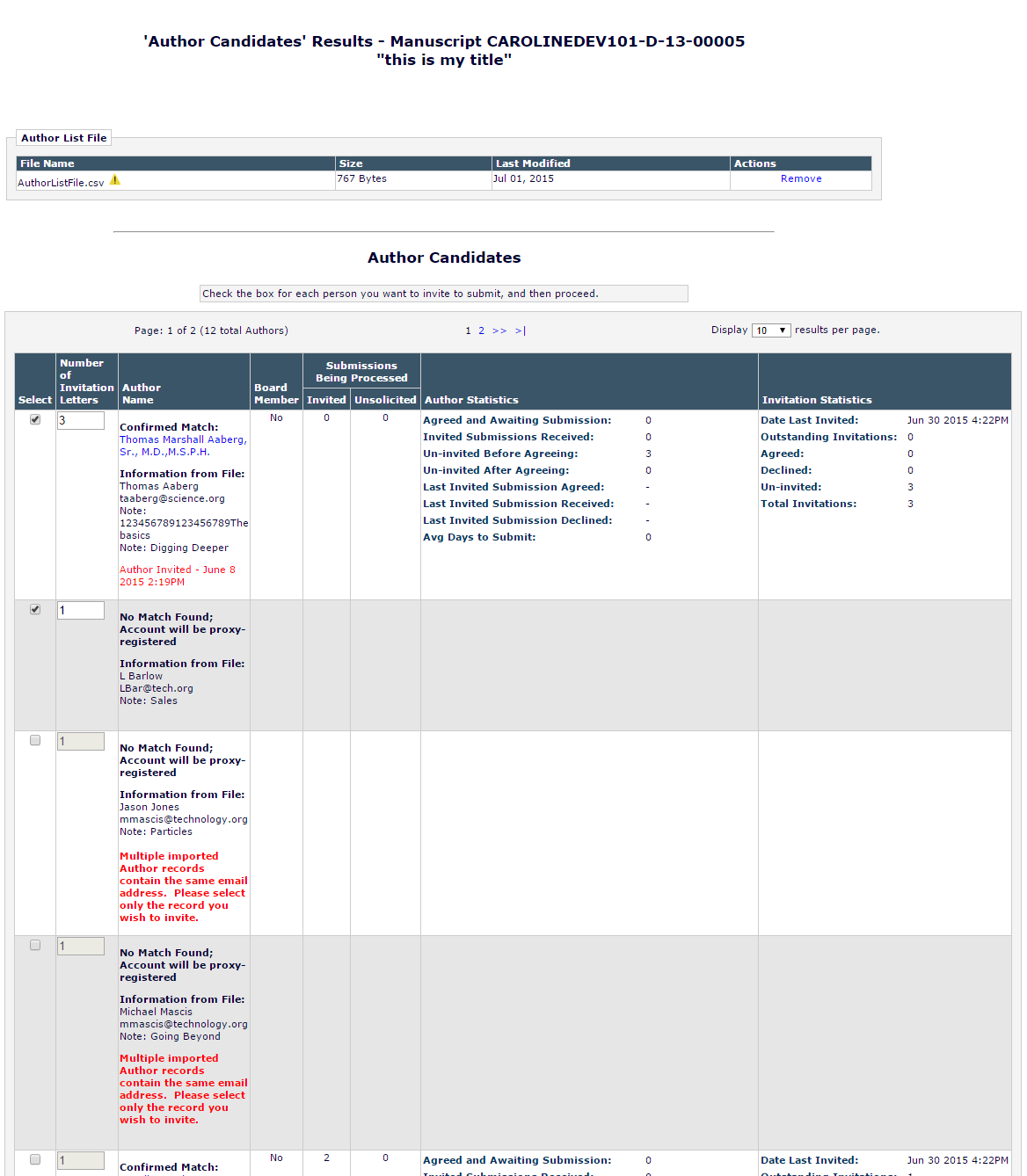
**Figure 25:** Partial layout of *‘Author Candidates’ Results* page showing the Author List File. An  icon is placed next to the file name to indicate an issue. A tool tip is displayed when rolling over the  icon. Clicking the icon will display an overlay listing the contents of the file. Upload options remain disabled once a file is uploaded unless the file is removed.  


**Figure 26:** Mock-up of the overlay which appears over the *‘Author Candidates’ Results* page when the  icon is clicked. This overlay is similar to what is displayed on the *Upload Author File* page following validation, but with different text and button labels. All the ‘Ignore’ check boxes are displayed because the file has already been ingested.



* 1. Below the Author List File information the ‘Author Candidates’ grid will be displayed
     1. The ‘Author Candidates’ grid on the *‘Author Candidates’ Results* page the same layout as the ‘Author Candidates’ grid which appears on the Author Search Results pages (searchAuthorsByNameAddrResults.asp and searchAuthorsByPersonalClassResults.asp)

**Figure 27:** Mock-up of the the *‘Author Candidates’ Results* page. The Number of Invitation Letters field is only enabled when the record is selected.



* 1. The ‘Author Candidates’ grid displays the contents of an ALF has been uploaded, validated, and ingested (See Section 2.1.3 for validation and Section 2.1.4 for ingest rules) within the interface
     1. Each unique Author ingested from an ALF receives their own row in the ‘Author Candidates’ grid
     2. If multiple entries for an Author are present in an ALF the duplicates are treated as additional invitations.
        1. The initial value to be displayed in the Number of Invitations Letters field in the ‘Author Candidates’ grid is ‘1’. For each duplicate entry, the number increments by 1. (See Section 2.1.4)
  2. The ‘Author Candidates’ grid is populated in the order of the records as they appeared in the ALF
  3. The ‘Author Candidates’ grid’s ‘Author Name’ column is enhanced to provide information about the ingested Author List File
     1. If there is confirmed a match from the Author List File to a registered user in the EM database (See Item 2 in Section 2.1.4 for how a match is defined)
        1. “Confirmed Match:” is displayed at the top of the Author Name cell, followed by the Author’s name and degree/title information (as stored in the EM PEOPLE table) on the next line (This matches how names are displayed in search results in the existing functionality)
           1. The Author’s name is displayed as a link
           2. Clicking the Author’s name will bring up their *People Information* page (reviewer\_info.asp)
           3. If the Editor has permission to view or add People Flags the People Flags icon will appear next to the name
        2. “Information from File:” is displayed bolded following the Author’s name on a new line
        3. This is immediately followed by the potential Author’s full name (FIRSTNAME LASTNAME)from the ALF file and the email address(es), as well as the optional ORCID and the contents of the NOTE field
           1. Each item begins on a new line
           2. If an Authenticated ORCID iD is present in the EM record it is displayed as a link to the ORCID public profile, https://orcid.org/[ORCID-ID]

Where [ORCID-ID] is replaced by the ORCID iD value tied to the EM account

Authenticated ORCID iDs display the ORCID icon per existing functionality

* + - * 1. Each NOTE field text value is preceded by “Note:”
        2. “Note:” is not displayed if no value exists for the field
        3. If an Author appeared in the ALF multiple times (duplicate rows), a row for each populated NOTE field will appear starting on a new line

The order of the NOTE field placement is determined by the order the NOTE values appear in the ALF

Each NOTE field text value is preceded by “Note:”

The NOTE field value is truncated at 25 characters and if truncated followed by an ellipsis

If the NOTE is less than or equal to 25 characters it is not truncated

* + - * 1. If there are multiple EM accounts which are matches for an Author record found in the ALF, text will be inserted into the ‘Author Name’ column indicating the potential duplicate

Text appears following the displayed information from the file

Text is displayed bolded in red

Text reads: “Multiple existing user accounts matched this record. Please select the account you wish to invite.”

See Figure 28 for example

* + - 1. The ‘Author Name’ column will continue to display pending invitations, accepted invitations, declined invitations, and assignments as an Alternate Author per existing functionality
         1. This information is displayed below the ALF file information and any warnings displayed by EM introduced by this specification
    1. If EM has found No Match from the Author List to the EM database:
       1. “No EM Match Found; Account will be proxy-registered” is displayed in EM blue at the top of the ‘Author Name’ cell
       2. “Information from File:” in EM blue follows a line below
          1. This is immediately followed by the potential Author’s full name (FIRSTNAME LASTNAME) from the ALF file, the email address(es), ORCID iD and the contents of the NOTE field if present
          2. If an ORCID iD is present it is displayed as a link to the ORCID public profile, https://orcid.org/[ORCID-ID]

Where [ORCID-ID] is replaced by the ORCID iD value present in the ALF

* + - * 1. NOTE field text is preceded by “Note:”

“Note:” is not displayed if no value exists for the field

* + - * 1. If an Author appeared in the file multiple times, a row for each populated NOTE field value will appear starting on a new line

Order of the NOTE field placement determined by the order the NOTE values appear in the ALF

* + - * 1. If there are multiple ALF records which have the same email address but no other matching data and the ALF Author records do not match existing EM accounts, each will be given its own row in the ‘Author Candidates’ grid

Warning text will be inserted into the ‘Author Name’ column indicating the duplicate email addresses

Text appears following the displayed information from the file

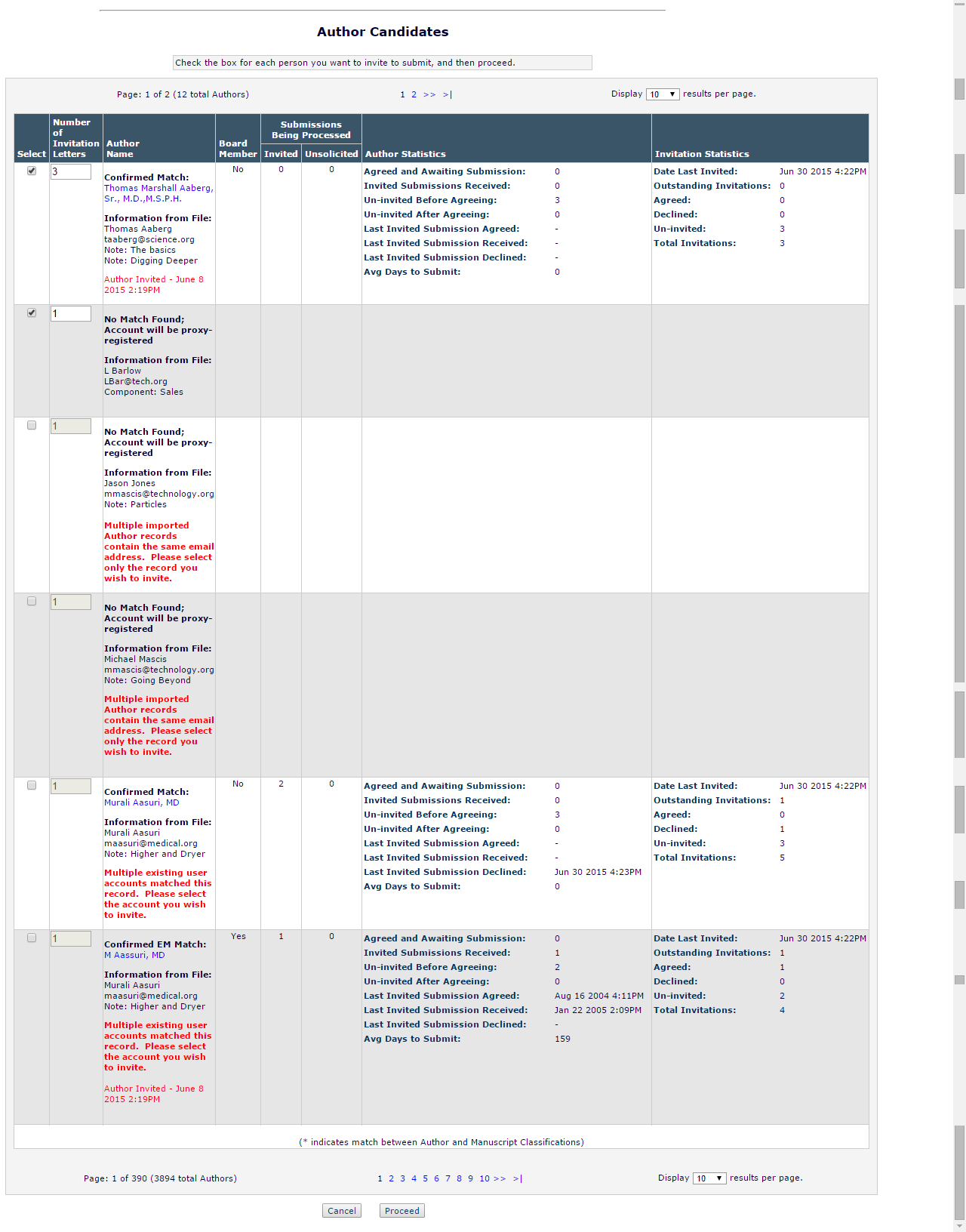
Text is displayed bolded in red

* + - * 1. Text reads: “Multiple imported Author records contain the same email address. Please select only the record you wish to invite.”

See Figure 28 for example

* + - * 1. These records will not be selected by default regardless of the associated INVITE value

**Figure 28:** Mock-up of ‘Author Candidates grid’ Enhancements to the ‘Author Name’ column are shown, including warnings related to duplicate matches and records.



1. The Select box for Authors listed in the ‘Author Candidates’ grid is checked off for invitation by based on the INVITE value found in the ALF
   1. If INVITE = TRUE then the Author record is selected for invitation
   2. If INVITE = FALSE or is null the Author is not selected for invitation
   3. **EXCEPTIONS** 
      1. Where multiple existing EM user accounts have been matched to an Author record from the ALF and the matches have been marked as duplicates
      2. Where multiple ALF records have the same email address but other info is different and no EM match found
   4. In these cases the Select box is NOT checked off by default
      1. Editors can select and unselect Authors at will
2. If the ‘Cancel’ button at the bottom of the page is clicked the uploaded ALF is discarded an the user is returned to the *Upload Author List* page where they can upload a new file
   1. A warning message with ‘Cancel’ and ‘OK’ options triggers when the ‘Remove’ button is clicked
      1. Message reads: “This will remove the uploaded file and discard any existing selections. Are you sure?”
         1. Clicking ‘OK’ proceeds with the removal of the file
         2. Clicking ‘Cancel’ leaves the file in place
3. With Authors selected for invitation the Editor can click the ‘Proceed’ button at the bottom of the page
   1. Clicking ‘Proceed’ while inviting Authors will bring the Editor to the *Select Authors – Confirm Selection and Customize Letters* page (confirmAndCustomizeInvitedAuthors.aspx) where they can proceed with the invitation process
      1. This page is enhanced in Section 2.5
      2. Proxy registrations of accounts for Authors without an existing EM account occurs when the‘Proceed’ button is clicked
         1. Required fields for proxy-registration are ignored when proxy-registration is carried out this using the new functionality
            1. Similar to Reviewer Discovery registrations (10.0-31)
         2. While proxy-registration occurs an overlay with blue spinner appears over the page
            1. The overlay contains text reading “Please wait while proxy registration is performed”
            2. This is the same overlay used when proxy-registering accounts via the Reviewer Discovery tool. (Spec **10.0-31,** existing code should be leveraged)

**Figure 29:** Mock-up of the overlay which will appear over the *Author Selection Summary* page when the ‘Proceed’ button is clicked and accounts need to be proxy-registered.



* + - 1. If a user who is not already registered in the system is to be invited more than once, he will be proxy-registered only once
      2. Letters are not sent to the user being proxy-registered notifying them of their new registration
         1. The expectation is that the first contact from the system the author receives will be the Author Invitation letter
         2. **EXCEPTION:** If the Hosting Administration setting ‘Enforce Notification at Proxy Registration’ is enabled will receive a notification when the account is registered

The letter sent will be the letter configured by the publication for the ‘Proxy Register New User’ event in ActionManager for the Author Role

* + - 1. If no accounts require proxy-registration the overlay is not displayed
      2. **Note:** When the FRA for this spec was reviewed at the Delphic Summit, it was noted that the proxy-registrations could be handled asynchronously however it was determined that the proxy-registrations should be handled synchronously, as in existing functionality for Reviewer Discovery, because the proxy registration process could fail for one or more users, and it would be difficult to resolve the situation once the user is navigated away from this page
  1. Once an Editor has invited Authors they may return to the *Author Selection Summary* page to select additional Authors using the existing search options or using the new Upload Author List option
     1. This allows for multiple Author List Files to be used for a single Proposal or Commentary, but keeps the restriction of one file per invitation session in place

1. If the *Author Selection Summary* page is closed prior to the ‘Proceed’ button being clicked the ALF file and ingested data is discarded

## Select Authors – Confirm Selection and Customize Letters page [confirmAndCustomizeInvitedAuthors.aspx]

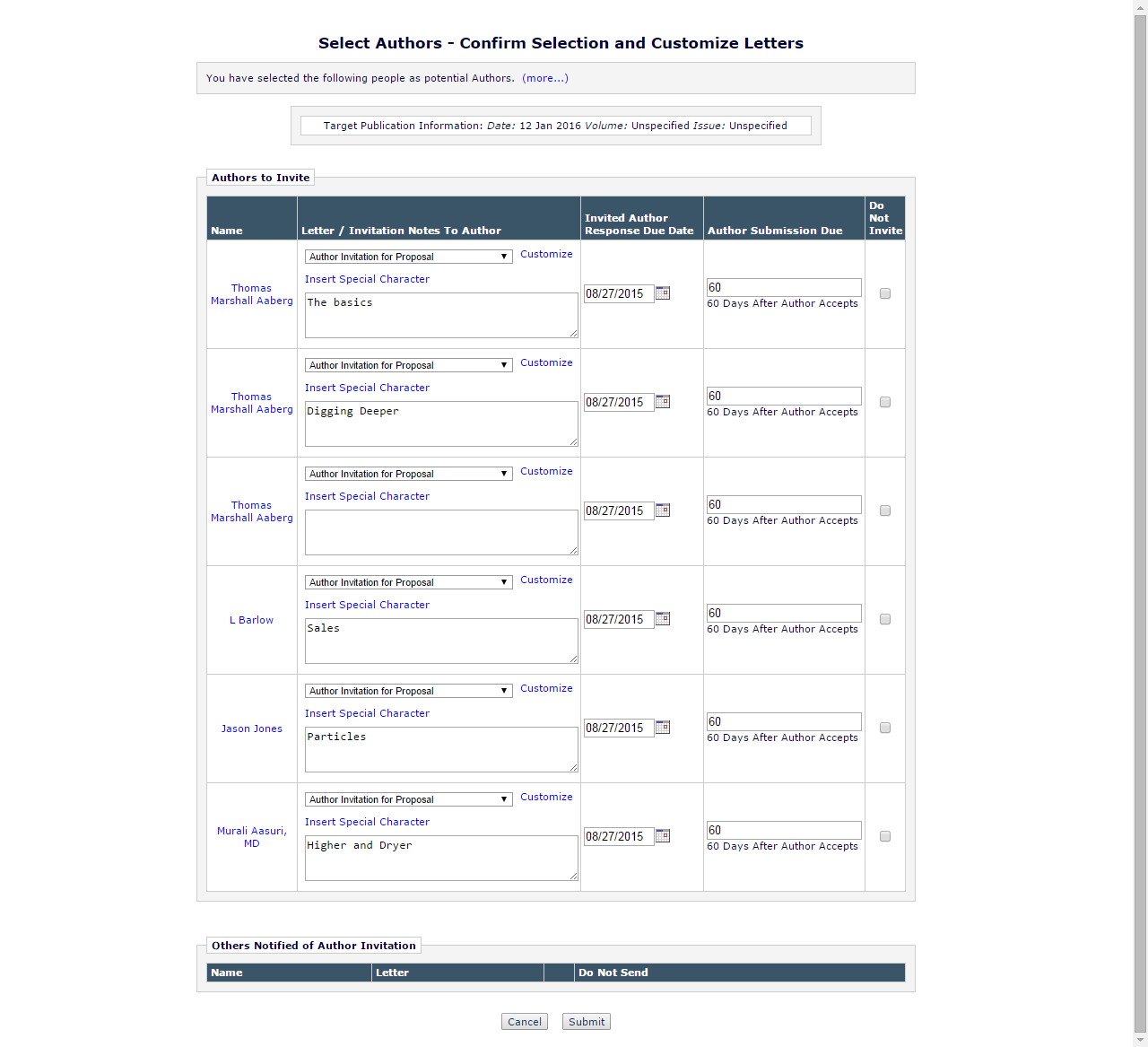
Users arrive at on the *Select Authors – Confirm Selection and Customize Letters* page when proceeding past the results page of an Author search. In the existing functionality two such pages exist (searchAuthorsByNameAddrResults.asp and searchAuthorsByPersonalClassResults.asp). This specification adds an additional results page, *‘Author Candidates’ Results* page, which will also direct users to the *Select Authors – Confirm Selection and Customize Letters* page.

The *Select Authors – Confirm Selection and Customize Letters* page is enhanced to incorporate the NOTE field from the Author List. The NOTE field value for each invitation will be written to INVITED\_AUTHORS.INVITED\_AUTHORS\_NOTES. The text will be displayed in the box for Invited Author Notes on the *Select Authors – Confirm Selection and Customize Letters* page which will save it to INVITED\_AUTHORS.INVITED\_AUTHORS\_NOTES when the invitation is sent and enable the NOTE field text to be used with the existing merge field, %INVITATION\_NOTES\_TO\_AUTHOR%.

**New Functionality:**

1. The *Select Authors – Confirm Selection and Customize Letters* page is enhanced
   1. For each optional NOTE for an invited Author, one of that Author’s invitations will display the value of the NOTE field in the ‘Invitation Notes to Author’ field
      1. Order of the NOTE field placement is determined by the order of the NOTE in the Author List File – i.e. the first invitation created for an author will display the first NOTE text value associated with that person as entered in the ALF; the second invitation the second NOTE text, until all NOTE values have been included.
      2. Authors may have more invitations than NOTE values
         1. Empty NOTE values will display once all NOTE values are included
   2. Invitation Notes to Author field must be configured to appear for the NOTE value to be carried forward
      1. Feature is activated under ‘Edit Additional Invitation Details’ in PolicyManager (existing page)
      2. The Editor retains the ability to edit and add to this text box if the NOTE field text value has been carried forward
         1. This includes editing the text ingested by from the ALF
   3. If a NOTE field was left blank no text will display in the ‘Invitation Notes to Author’ field for that invitation
   4. As in existing functionality, all text in the ‘Invitation Notes to Author’ field can be added to the customized letter with the merge field %INVITATION\_NOTES\_TO\_AUTHOR%
   5. Clicking the ‘Cancel’ button on this page returns a user to ‘*Author Candidates’ Results* page
      1. When returned to the ‘*Author Candidates’ Results* page the ‘Author Candidates’ grid remains populated with the contents of the Author List File
   6. Clicking the ‘Submit’ button sends invitations to the selected Authors per existing functionality

Figure 30 – Partial mockup of the *Select Authors – Confirm Selection and Customize Letters* page. If one or more optional NOTE field values were included for an Author, each NOTE is listed in a separate invitation. For example: Thomas Marshall Aaberg has three invitations, two of which had values in the corresponding NOTE field.



1. Other than populating the ‘Invitation Notes to Author’ field with the NOTE field value no functionality on the page is altered
2. When the user clicks the ‘Submit’ button, the system continues with the existing functionality, i.e. the invitation letters are sent and the *Author Invitation Confirmation* page is displayed

# Database Impact

This section will be updated after coding with input from Engineering if it has been determined that there will be impact to the database.

## DB Content Changes

This section will be updated after coding with input from Engineering if it has been determined that there will be impact to the database.

## UDB Impact

This section will be updated after coding with input from Engineering if it has been determined that there will be impact to the database.

# Multi-Lingual Considerations

# Performance Processing Impact

*Consideration guidelines:*

* *1,000,000 submissions per year*
* *5,000,000 people records*
* *NOTE: Expectation that all testing be on PONECLONE*

## Performance Requirements

### Data Server

### Web Server

### Network

### Client Machine

## Engineering Assessment

### Engineering Test Results

# Merge Field Impact

# Upgrade Script Impact

None.

# Site Cleanout

# Report Implications

## Canned Reports

## General Data Export

## Custom Reports

## Enterprise Analytics Reporting

*Describe any or all of the following:*

* *New tables to be added to the Data Sources tab*
* *New views to be added to the data sources tab*
* *New standard EAR reports, and which category they fall into*
* *Impact on scheduling or emailing of EAR reports*

## Cross-Publication EAR

# Hosting Documentation Impact

# Portal (IJRS, etc.) Impact

The proxy-registration of Authors from the Author List File during the invitation process will trigger an IJRS Synch for a publication in an IJRS Group; this is a new trigger point for this functionality.

# Hosting Administration Impact

## Hosting Administration Audit Trail

*Any changes made to any Hosting Administration code should include invocation of the HA Audit Trail code module to ensure that appropriate data about administration changes are recorded to the HA Audit Trail*

# Sys Admin Audit Trail

# Import/Export Impact

*Should consist of elements/data 1) that would need to be added to the generic Aries Export schema/XML and/or 2) available for Aries import and added to the “Select Fields for Import” page and Aries Import schema and script.*

## Transmittal Impact

## Aries Export Impact

## Aries Import Impact

# Transfer Impact

# Cross-Publication Impact

*Anything not covered in the reporting section (i.e. journal acronym change, cross-pub login, consideration for additional use cases, etc.)*

# Documentation and Training

Translators must be informed that the headers referred to in the help text should not be translated as using the English text for the headers is necessary for the process to work.

# Operations’ Considerations

## Journal Rename

## Journal Redirects

# Utilities

*Consider any or all of the following with Engineering consultation:*

* *UDB Utility*
* *Upgrader*
* *IJRS Sync Tool*
* *Deleter*
* *Cloner*
* *Migrator*
* *Translation Utility*

# Audit/Monitor/Diagnostics

*Used when going out to a 3rd party system to detect if the system is up. Add to an existing “Tembria dashboard”; indicate who should get e-mail notices if 3rd party goes down.*

# Journal Configuration

No additional configuration is needed to use the functionality introduced in this specification

# Style Sheet impact

# Deployment Considerations

## Associated Revenue Model

## Sales and Marketing Rollout Anticipation

## Customer Service Rollout Preparation

## Release Target

# Future Development

1. Future development planned for 13.1 will add the ability to assign authors in addition to inviting them. Spec 13.0-33 addresses this future development.
2. Future development planned for 13.1 will add the ability for the title of an invited submission to be predetermined by an Editor. This functionality will be incorporated into the Author List File upload process. Spec 13.0-31 addresses this future development.
3. Future development of additional headers or variant headers to the ALF for the purpose of supporting additional languages. Engineering believes adding variant header text would require a mini-spec level of effort at most.
4. Future development may allow the uploading of a list for inviting Reviewers or Editors.
5. Future development may allow the uploading of multiple Author List Files during the same invitation session.
6. Future development may allow for Editors to use a mix of search methods when inviting Authors.

# Mobile Considerations

*Functional considerations for smartphones and tablets (if applicable)*

# Use Cases and Test Scenarios

## Use Cases

1. Publications will use the new functionality introduced in this specification as part of a Books workflow inviting Authors to submit chapters and sections as part of a larger books project. In a books project the authors are generally determined and solicited well be before the project begins, allowing publication staff to compile a list of Authors and their assignments well in advance. Feedback from publishers has indicated that this would entail inviting as many as 150 to 200 authors per book.

## Specification Test Cases

1. **Confirm that the enhanced layout for the *Author Selection Summary* page is appearing when the new ‘Upload Author List’ option is selected**
   1. Go to the *Author Selection Summary* page
   2. Select ‘Upload Author List’ from the Author Search Drop Down
   3. Confirm that the  icon appears to the right of the drop down menu
      1. Confirm that hovering over the icon reveals a tool tip jQuery box
         1. Confirm that the initial tool tip is collapsed
         2. Click the ‘more’ link to confirm that the tool tip box expands to its expanded version
         3. Click the ‘less’ link to confirm that the tool tip returns to its collapsed version
         4. Take the mouse off the tool tip to confirm that it disappears
   4. Select another Author search option and confirm that the icon no longer appears
2. **Confirm that the *Upload Author List* page is accessible and that the ‘Choose File’ upload option found on the page is functional**
   1. Go to the *Author Selection Summary* page
   2. Select ‘Upload Author List’ from the Author Search Drop Down
   3. Click the ‘Go’ button
   4. Confirm that the *Upload Author List* page loads
   5. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in Section 2.1.1
   6. Click the ‘Choose File’ button on the *Upload Author List* page
      1. Confirm that the user is prompted to select a file to upload as the Author List File from their local computer
      2. Browse for the prepared Author List File and choose to upload it
         1. While the file is uploading:
            1. Confirm that the ‘Choose File’ button is disabled and greyed out
            2. Confirm that the drag and drop area is disabled and greyed out
            3. Confirm that an ‘File in Progress’ box has replaced the “No Author List File has been uploaded for this submission” text on the *Upload Author List* page
            4. Confirm that while the file is uploading the ‘File in Progress’ box will display the following information:

That the name of the file is listed

That the percentage of the upload completed is displayed

That a progress bar showing how much of the file has been uploaded is displayed

* + - * 1. Confirm that once the file uploaded is completed the validation process begins

The full progress bar remains displayed

The displayed percentage is replaced with a blue spinner

* 1. Confirm that once the upload and validation is complete an overlay will appear displaying the contents of the file
     1. Confirm the ‘Remove’ button removes the file and restarts the process then repeat steps A through C
        1. When clicking the Remove button an ‘Are you Sure?’ warning should appear
        2. Confirm both the ‘Cancel’ and ‘OK’ options function
     2. Confirm the ‘Import’ button ingests the file, dismisses the overlay and sends the user to the *‘Author Candidates’ Results* page
  2. Confirm that the ALF is displayed on the *‘Author Candidates’ Results* page in an ‘Author List File’ section
     1. The name, size, last modified date and a ‘Remove’ link are present
     2. Confirm that the ‘Remove’ link functions
  3. Confirm that the ‘Author Candidates’ grid is now display on the *‘Author Candidates’ Results* page below the ‘Author List File’ section
     1. Confirm that the NOTE value contents from the Author List File appear as expected in the Author Name column of the grid
  4. Click the ‘Remove’ link
     1. Confirm that this removes the uploaded file
     2. Confirm that once the file is removed the user is returned to the *Upload Author List* page
        1. The ‘Author Candidates’ grid will no longer display
        2. The “No Author List File has been uploaded for this submission” text displays
        3. The upload options are now enabled and no longer greyed out

1. **Confirm that the *Upload Author List* page is accessible and that the ‘Drag and Drop File’ upload option found on the page is functional**
   1. Go to the *Author Selection Summary* page
   2. Select ‘Upload Author List’ from the Author Search Drop Down
   3. Click the ‘Go’ button
   4. Confirm that the *Upload Author List* page loads
   5. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in Section 2.1.1
   6. Drag and drop the prepared Author List File on to the ’Drag and Drop File’ area on the *Author Selection Summary* page
      1. Confirm that the upload process begins immediately
         1. While the file is uploading:
            1. Confirm that the ‘Choose File’ button is disabled and greyed out
            2. Confirm that the drag and drop area is disabled and greyed out
            3. Confirm that an ‘File in Progress’ box has replaced the “No Author List File has been uploaded for this submission” text on the *Upload Author List* page
            4. Confirm that while the file is uploading the ‘File in Progress’ box will display the following information:

That the name of the file is listed

That the percentage of the upload completed is displayed

That a progress bar showing how much of the file has been uploaded is displayed

* + - * 1. Confirm that once the file uploaded is completed the validation process begins

The full progress bar remains displayed

The displayed percentage is replaced with a blue spinner

* 1. Confirm that once the upload and validation is complete an overlay will appear displaying the contents of the file
     1. Confirm the ‘Remove’ button removes the file and restarts the process then repeat steps A through C
        1. When clicking the Remove button an ‘Are you Sure?’ warning should appear
        2. Confirm both the ‘Cancel’ and ‘OK’ options function
     2. Confirm the ‘Import’ button ingests the file, dismisses the overlay and sends the user to the *‘Author Candidates’ Results* page
  2. Confirm that the ALF is displayed on the *‘Author Candidates’ Results* page in an ‘Author List File’ section
     1. The name, size, last modified date and a ‘Remove’ link are present
     2. Confirm that the ‘Remove’ link functions
  3. Confirm that the ‘Author Candidates’ grid is now display on the *‘Author Candidates’ Results* page below the ‘Author List File’ section
     1. Confirm that the NOTE value contents from the Author List File appear as expected in the Author Name column of the grid
  4. Click the ‘Remove’ link
     1. Confirm that this removes the uploaded file
     2. Confirm that once the file is removed the user is returned to the *Upload Author List* page
        1. The ‘Author Candidates’ grid will no longer display
        2. The “No Author List File has been uploaded for this submission” text displays
        3. The upload options are now enabled and no longer greyed out

1. **Confirm that uploading a new Author List File after removing a different Author List File displays only the correct list of Authors in the ‘Author Candidates’ grid**
   1. Prepare two unique Author List Files for testing which are formatted correctly, without errors per the requirements outlined in Section
   2. Upload the first of the two files on the *Upload Author List* page and observe that the ‘Author Candidates’ grid is populated on the *‘Author Candidates’ Results* page
   3. Remove the file
   4. Upload the second of the two files and confirm that only the Authors from the second list are displayed in the ‘Author Candidates’ grid and that nothing has been retained from the previous list
2. **Confirm that all supported file types can be uploaded**
   1. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in Section 2.1.1
   2. Save a version of this file in Comma Separated Value format( \*.csv) and Tab Delimited (\*.tab, \*.txt)
   3. Upload each file type using both the ‘Choose File’ and ‘Drag and Drop’ options and confirm that they can be ingested
3. **Confirm that error appears when uploading an unsupported file type as the Author List File**
   1. Prepare a file of an unsupported type which will be used for attempting to upload an Author List File on the *Upload Author List* page
   2. Drag and drop the prepared Author List File on to the ’Drag and Drop File’ area on the *Upload Author List* page
      1. Confirm that an warning message appears in the jQuery style box indicating that the file you have uploaded is of an unsupported type
      2. See Figure 10 for example
      3. Click ‘OK’ to dismiss the warning message
   3. Use the ‘Choose a File’ option to select the prepared Author List File on the *Upload Author List* page
      1. Confirm that an warning message appears in the jQuery style box indicating that the file you have uploaded is of an unsupported type
      2. See Figure 10 for example
      3. Click ‘OK’ to dismiss the warning message
4. **Confirm that an error message occurs when uploading an Author List File of the correct file type with required headers missing**
   1. Several Author List files will need to be prepared all should be prepared containing data which is correct per the requirements outlined in Section 2.1.1
   2. Each of these files will have at least one missing of the three required headers (FIRSTNAME, LASTNAME, EMAIL), combinations needed for testing include:
      1. Missing FIRSTNAME only
      2. Missing LASTNAME only
      3. Missing EMAIL only
      4. Missing FIRSTNAME and LASTNAME
      5. Missing FIRSTNAME and EMAIL
      6. Missing LASTNAME and EMAIL
      7. Missing FIRSTNAME, LASTNAME and EMAIL
   3. Upload each file using the ‘Choose a File’ option on the *Upload Author List* page
      1. Confirm that an warning message appears in the jQuery style box indicating that the file you have uploaded is missing required headers
         1. Confirm that the warning correctly identifies the missing headers
      2. See Figure 11 for example
      3. Click ‘OK’ to dismiss the warning message
   4. Repeat using the ‘Drag and Drop’ option
5. **Confirm error message appears when Author List File contains more records than allowed per an Author List File**
   1. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in Section **Error! Reference source not found.** but contains more than the 200 record limit per Author List File
   2. Upload the file using the ‘Choose a File’ option on the *Upload Author List* page
      1. Confirm that an warning message appears in the jQuery style box indicating that the file you have uploaded is over the limit for number of records
      2. See Figure 12 for example
      3. Click ‘OK’ to dismiss the warning message
   3. Repeat using the ‘Drag and Drop’ option
6. **Confirm error message appears when Author List File does not contain the minimum number of records**
   1. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in Section 2.1.1**Error! Reference source not found.** but contains more than the no Author records, jus the required headers
   2. Upload the file using the ‘Choose a File’ option on the *Upload Author List* page
      1. Confirm that an warning message appears in the jQuery style box indicating that the file you have uploaded does not contain any Author records
      2. See Figure 13 for example
      3. Click ‘OK’ to dismiss the warning message
7. **Confirm that the warning icon appears correctly in the overlay when required fields are missing from Author records in the Author List File or are improperly formatted**
   1. Prepare an Author List File for testing, this file incorporate some of the errors outlined in Section **Error! Reference source not found.** which will trigger the warning icons in the overlay display following validation
      1. The file must be formatted correctly and of the correct file type
      2. There are five different triggers for the warning flags
         1. Missing FIRSTNAME
         2. Missing LASTNAME
         3. Missing EMAIL
         4. Invalid EMAIL
         5. Invalid ORCID iD
      3. The file should be constructed to include one of each the listed triggers across
   2. Once the file is prepared it upload the file using the ‘Choose a File’ on the *Upload Author List* page
   3. Once the file validates an overlay containing the records will display
   4. Confirm that the missing fields are flagged with the icon
   5. Confirm that the invalid fields are flagged with the icon
   6. Confirm that each icon has the appropriate tool tip text appear when a cursor rolls over the icon
      * 1. When a FIRSTNAME value is not provided, rolling over the icon displays a tool tip
           1. Tool top reads: “First Name is a required field. This record cannot be imported.”
        2. When a LASTNAME value is not provided, rolling over the icon displays a tool tip
           1. Tool top reads: “Last Name is a required field. This record cannot be imported.”
        3. When an EMAIL value is not provided, rolling over the icon displays a tool tip
           1. Tool top reads: “Email address is a required field. This record cannot be imported.”
        4. When an EMAIL value is not formatted correctly, rolling over the icon displays a tool tip
           1. Tool tip reads: “This email address is invalid. This record cannot be imported.”
        5. When an ORCID iD is not formatted correctly, rolling over the icon displays a tool tip
           1. Tool tip reads: “This ORCID iD is invalid. The ORCID value will not be imported.”
   7. Repeat using ‘Drag and Drop’ option to upload file
8. **Confirm that proxy-registrations take place once the Editor proceeds past the ‘Author Candidates’ grid**
   1. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in 2.1.1
      1. This list should contain Authors who are not are in the system already, unique names, email and ORCID iDs should be used
   2. Upload file using ‘Choose a File’ option on the *Upload Author List* page
   3. Confirm that the upload occurs and that file is validated
   4. Confirm that the overlay with all the records appears
   5. Click the ‘Import’ button
      1. Confirm that the overlay is dismissed
      2. Confirm that the user is directed to the *‘Author Candidates’* *Results* page
   6. Confirm that the Authors are marked in the ‘Author Candidates’ grid on the *‘Author Candidates’* *Results* page as having no existing match to the EM database
   7. Click Proceed
   8. Confirm that an overlay appears and indicates that Proxy Registration is taking place
   9. Confirm that you have arrived on the Select Authors – Confirm Selection and Customize Letters page [confirmAndCustomizeInvitedAuthors.aspx] once proxy registration was complete
   10. Check the database to confirm the Authors have had accounts created
   11. Confirm that no letters were sent to the Authors
   12. Repeat process with a second file using the ‘Drag and Drop’ option to upload the file
9. **Confirm that letters are sent during proxy registration when HA setting is enabled**
   1. Enable the ‘Enforce Notification at Proxy Registration’ setting for the test site in Hosting Administration under Set Journal Features
   2. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in Section 2.1.1
      1. This list should contain Authors who are not are in the system already, unique names, email and ORCID iDs should be used
   3. Upload file using ‘Choose a File’ option on the *Upload Author File* page
   4. Confirm that the upload occurs and that file is validated
   5. Confirm that the overlay with all the records appears
   6. Click the ‘Import’ button to display the ‘Author Candidates’ grid on the *‘Author Candidates’* *Results* page and dismiss the overlay
   7. Confirm that the Authors are marked in the ‘Author Candidates’ grid as having no existing match to the EM database
   8. Click Proceed
   9. Confirm that an overlay appears and indicates that Proxy Registration is taking place
   10. Confirm that you have arrived on the Select Authors – Confirm Selection and Customize Letters page [confirmAndCustomizeInvitedAuthors.aspx] once proxy registration was complete
   11. Check the database to confirm the Authors have had accounts created
   12. Confirm that letters were sent to the Authors
   13. Repeat process with a second file using the ‘Drag and Drop’ option to upload the file
10. **Confirm proper handling of duplicates with the Author List File**
    1. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in Section 2.1.1
       1. The file will contain duplicate information to test the ability to match records within the file and identify duplicates
          1. The Author List File is checked for multiple entries for the same record
          2. These duplicate entries are determined to be duplicates if the FIRSTNAME, LASTNAME and EMAIL match other existing records
             1. If there is a partial match the records are displayed as separate entities in the ‘Author Candidates’ grid
       2. The file should contain three records which have identical FIRSTNAME, LASTNAME, and EMAIL values
       3. The file should contain two records where the EMAIL value matches, but the names do not
    2. Upload the Author List File using the ‘Choose a File’ option on the *Upload Author List* page
    3. Confirm that the file uploads and validates
    4. Confirm that the overlay displays listing the file’s contents
    5. Click ‘Import’
       1. Confirm that the overlay is dismissed
       2. Confirm that user is directed to the *‘Author Candidates’ Results* page
    6. Confirm that the record that was included three times appears once in the ‘Author Candidates’ grid and has a value of ‘3’ for the Number of Invitation Letters
    7. Confirm that the records that matched on email only appear separately in the ‘Author Candidates’ grid and has a value of ‘1’ for the Number of Invitation Letters
    8. Repeat using ‘Drag and Drop’ option to upload file
11. **Confirm matching process properly matches Author Records to existing records in the EM database**
    1. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in Section 2.1.1
       1. This list should include Authors who are already registered in EM in various configurations in order to test the matching functionality
       2. There are several matching types that can occur and a specific order to the match (See Table 1)
       3. All records in the file should be set to INVITE =TRUE
       4. Include in the file an Author record which contains and ORCID iD which matches an Authenticated ORCID associated with an EM account
       5. Include in the file an Author record which contains a ORCID iD which matches and Non-Authenticated ORCID associated with an EM account and a LASTNAME which matches the same EM account
       6. Include in the an Author record which matches two EM accounts on Non-Authenticated ORCID and LASTNAME, but uniquely matches a single account when FIRSTNAME is included
       7. Include in the file an Author record which matches two EM accounts on Non-Authenticated ORCID, LASTNAME, and FIRSTNAME but uniquely matches a single account when EMAIL is included
       8. Include in the file an Author record which contains no ORCID and includes an EMAIL value which matches a unique EM account
       9. Include in the file an Author record which contains no ORCID and matches two EM accounts on EMAIL, but uniquely matches a single account when LASTNAME is included
       10. Include in the file an Author record which contains no ORCID and matches two EM accounts on EMAIL and LASTNAME, but uniquely matches a single account when FIRSTNAME is included
       11. Include in the file an Author record which contains no ORCID and matches a unique EM account where EMAIL matches the alternate email address in the Address Table and LASTNAME matches the last name found in the People Table for that account
       12. Include in the file an Author record which contains no ORCID and matches a two EM account where EMAIL matches the alternate email address in the Address Table and LASTNAME matches the last name found in the People Table for that account, but uniquely matches an account when FIRSTNAME is included
       13. Repeat steps f through j with an ORCID included, but one which matches no existing EM records
       14. Include in the file an Author record which matches two EM accounts where a unique match cannot be made between the two
    2. Upload this file using the ‘Choose a File’ option on the *Upload Author List* page
    3. Confirm that the file uploads and validates
    4. Confirm that the overlay displays listing the file’s contents
    5. Click ‘Import’
       1. Confirm that the overlay is dismissed
       2. Confirm that the *‘Author Candidates’ Results* page appears
    6. Confirm that the expected matches to existing EM records have been made
       1. All examples except 14.a.xiv should be matches to a specific EM account
       2. In the case of example 14.a.xvi both matching accounts should appear in the ‘Author Candidates’ grid but should be flagged as potential duplicates an NOT selected by default to be invited
    7. Repeat uploading file using ‘Drag and Drop’ option
12. **Confirm that Invitations Letters are sent from the Select Authors – Confirm Selection and Customize Letters page** 
    1. Enable the Invitation Notes to Author field on the test site
    2. Prepare an Author List File for testing which is formatted correctly, without errors per the requirements outlined in Section 2.1.1
       1. This file should include NOTE field values for the Author records
    3. Upload file using ‘Choose a File’ option on the *Upload Author List* page
    4. Confirm that the upload occurs and that file is validated
    5. Confirm that the overlay with all the records appears
    6. Click the ‘Import’ button
       1. Confirm that the overlay is dismissed
       2. Confirm that the user is directed to the *‘Author Candidates’ Results* page
    7. Confirm that the Authors are marked in the ‘Author Candidates’ grid
    8. Click ‘Proceed’
    9. Confirm that you have arrived on the Select Authors – Confirm Selection and Customize Letters page [confirmAndCustomizeInvitedAuthors.aspx] Check the database to confirm the Authors have had accounts created
    10. Confirm that any NOTE values from the file have been pulled into the Invitation Notes to Author field
    11. Proceed to send invitations
    12. Confirm that invitations were sent and that %INVITATION\_NOTES\_TO\_AUTHOR% pulled the in the comments properly
        1. Confirm that NOTES which are customized on the page are then properly pulled into the letter via the merge field
    13. Repeat process with a second file using the ‘Drag and Drop’ option to upload the file
13. **Confirm that test cases 10, 13, 14 can be run using a single Author List File**
    1. Repeat test cases 10, 13, 14 with the test cases for each incorporated into a single Author List File to confirm that these disparate cases can be detected during the same upload and ingest process

# Appendix A: Technical Specification

**13.0-18**

Select Authors from Uploaded List

**EM Version: 13.0**

# Contributors:

|  |  |
| --- | --- |
| Primary Engineer | Andrew Pokrovski |
| Senior Engineer | Andrew Pokrovski |
| Coding Engineer(s) | Matt Van Voorhies,  Jean Gelinas |

# Document Revision History:

|  |  |  |
| --- | --- | --- |
| 09/09/2015 | Matt Van Voorhies, Andrew Pokrovski | Created Technical Specification |
| 10/08/2015 | Jean Gelinas | V2 - reorg |
| 10/09/2015 | Jean Gelinas | V3 – reorg 4.6 & 4.7 after discussion with Andrew. |
| 11/11/2015 | Matt Van Voorhies | V4 - Update |

# Summary:

This spec will enhance the existing Author Selection process by adding functionality for uploading and pre-registering a list of authors from an external file.

# General Design:

There are 3 major pieces to the implementation of the Author Upload functionality:

1. The File Import process and related UI enhancements
2. The Parsing of the Author List File
3. The Author List File Results and related UI enhancements

## The File Import Process (authorSelectionSummary.aspx) - Matt

The file import process occurs on the authorSelectionSummary.aspx page. The current page is updated to include a file import button and drag-drop area for importing the Author List File (ALF). The File Import Process is similar to the interface used for adding Submission files during Manuscript creation and hence the existing code will be leveraged as much as possible for implementation of the AuthorSelectionSummary page.

### Pages

#### AuthorSelectionSummary.aspx(.cs)

AuthorSelectionSummary.aspx includes minor changes to add a new option to the existing drop-down list.

At about line 195 in authorSelectionSummary.aspx.cs a “searchModeItem” is created and assigned the constant value: Constants.AUTHOR\_SEARCH\_MODE\_UPLOAD\_AUTHOR\_LIST

Logic is added at about line 779 in the btnGoSearch\_Click event handler to redirect to the new UploadAuthorList.aspx page which implements section 2.3 of the functional spec.

An Information icon using existing image file img/info\_icon.png is added to the right of the drop-down along with a link tag for the tooltip. A spacer icon using existing image file img/spacer.gif is added. Toggling between these icons is done via a JQuery function tied to the existing drop-down. The spacer is used to keep the page layout from varying when the icon appears.

An Aries:CollapsibleTextControl is added in a new div at the end of the HTML form (about line 262) and hidden. The control requires registration at about line 3 of the aspx file using the existing WebUserControls/CollapsibleText.ascx file. Initialization of the control (using the instruction translation text used in UploadAuthorList.aspx (see section 4.1.1.2)) is done in the .cs file at about line 69. A character limit of 230 is set and the existing tokens for “more…” and “less…” are used.

A JQuery tooltip is used to implement the tooltip functionality. The JQueryUI tooltip function is called in the JQuery OnReady function and the div created with the collapsible text control inside it is used as the content. The tooltip’s “Open” and “Close” callbacks are overridden to allow the tooltip to stay open while the mouse remains in either the bounds of the icon, or within the tooltip content window so the user can click the expansion buttons without the tooltip disappearing.

#### + UploadAuthorList.aspx

This new page is described in the functional spec in section 2.3. It utilizes the JQuery File Upload Plugin and passes a list of Authors to the “Search results” style page described in section 2.4 of the functional spec.

The file import will utilize the JQuery File Upload Plug-In, similar to *AttachManuscriptFiles.aspx*. In order to do this the page is updated as follows:

* JQuery libraries are included at about line 23
* JQuery OnReady function is added to script block to initialize the File Upload Plug-in
* A div for the Dynamic Upload (using the plug-in) with class ‘deliveryPanel’ is added. This panel has the fileInput ‘input’ HTML as well as the HTML for the Drag/Drop area.
* A div is added for Static Upload (for browsers that don’t support the drag-drop plug-in) with a standard HTML input of type ‘file’
* A div for the progress Panel is added similar to the HTML in *AttachManuscriptFiles.aspx* at about line 286.

The error messages are displayed like standard ‘alert’ message boxes, but for a more consistent look they are implemented as JQuery modal dialogs. In order to do this the page is updated as follows:

* A div for the warning message is added at about line 232
* A javascript function for setting the content to a provided string message and opening the dialog is added at about line 565
* Appropriate CSS styles are added at about line 32. Styles from cssSiteStyle.aspx and cssJQueryDialog.aspx are also required, so these files are imported.

### Code

The file upload process uses the following classes to parse and process the uploaded Author List file.

#### Aries.EditorialManager

##### Constants.cs

This existing class is updated to add a new constant for the Author Search Mode for Upload Author List at about line 1217.

public const int AUTHOR\_SEARCH\_MODE\_UPLOAD\_AUTHOR\_LIST = 5;

##### AuthorListFileInfo.cs

This new class is created to store information about an uploaded Author List File.

Properties:

Public string FileName

The ALF file name.

Public int Length

The ALF file size in bytes.

Public DateTime UploadTime

The ALF upload day/time.

Public string ContentType

The ALF file content type.

Public bool HasErrors

A flag indicating whether the file has any records with errors.

#### AEM.Web.Command.Upload

##### + UploadAuthorListFile.cs

This class is similar to the UploadSubmissionFile class used for uploading submission files. This is a new class introduced in this spec.

The main functionality of this file that differs from a standard file upload handler file is contained in the override of the ProcessFile.

protected override void ProcessFile(Stream inputStream, string filename, long fileLength, string contentType, DateTime lastModified)

This method checks the file for the valid/accepted extensions and creates a copy for processing. The method then passes this copied file to the ParseAuthorListFile method. If any errors occur, the UploadStatusInfo.ErrorCode is set to the appropriate error message and the method returns prematurely.

protected void ParseAuthorListFile(string filePath)

This method parses the ALF as follows:

* The file is attempted to be opened as a text file. If this fails, UploadStatusInfo.ErrorCode is set and the method returns.
* The first line of the file is read. This line MUST contain the column headers, in any order. The line is split using either a tab or semi-colon as a delimiter. A hashtable is used to map the column to the index in the delimited list.
* The hashtable is checked for the required columns FIRSTNAME, LASTNAME and EMAIL. If any of these columns were not found in the column list, an error is generated specifying the specific missing columns. The error message string is set using the translation token for the error message and the generated list of column names. The error message string, if applicable, is set into UploadStatusInfo.ErrorCode.
* The remaining lines in the file are read and parsed using the column hashtable. A new UploadedAuthorInfo object is created for each line. Default values are set for any missing items. It is not an error at this time for required values to be missing.
* A GUID is generated for each author entry and set in the UploadedAuthorInfo object.
* A count is maintained as each line is parsed. If the count exceeds 200, the parsing is terminated.
* If either there are zero author entries, or more than 200 author entries, the appropriate error message is set into UploadStatusInfo.ErrorCode and the method returns.
* If no errors occur and everything is valid, the UploadStatusInfo.ValidationResult is set and the UploadedAuthorInfo object is stored in cache.

UploadAuthorStatusInfo

This class is serialized into the JSON return object for the File Upload process. The class definition is contained in the source file for UploadAuthorListFile.

Properties:

Public string FileName

The file name that was uploaded.

Public string ErrorCode

The error code if the upload failed.

Public ValidationResult ValidationResult

The validation results of the file upload process.

Public List<UploadedAuthorInfo> AuthorInfoList

A list of UploadedAuthorInfo objects containing the parsed output generated by parsing the Author File List.

UploadedAuthorInfo

This class represents an individual Author from the uploaded Author List File.

Properties:

Public Guid AuthorID

Public string FirstName

Public string LastName

Public string EmailAddress

Public string AuthorNote

Public string OrcID

Public bool Invite

Public bool AuthorRegistered

#### Aries.EditorialManager.Web

##### UploadScheme.cs

This existing class is updated to provide for the new Upload type - AuthorListFile

At about line 100 the following item is added to the enum UploadScheme:

AuthorListFile = 10

## Author List Overlay Dialog - Matt

The Author List Overlay Dialog appears after the user has uploaded the Author List File. This overlay is modeled after similar dialogs used in EM. The overlay presents the parsed list of Authors using the JSON object list returned by the File Upload Plugin. The overlay itself is presented as a JQuery dialog.

### Code

#### + AuthorListFileAuthors.js

This new javascript library file contains the JQuery function called when the file upload and parsing process has completed.

A JQuery dialog is opened with the function defined in this library using *AuthorListFileAuthors.ashx* as the Ajax target and passing the list of Author objects.

The dialog function call passes the parameters:

modal: true,

autoOpen: false,

resizable: false,

closeOnEscape: false,

width: $contentBody.width() \* 0.80,

dialogClass: 'wizard',

open: function () {

$dlgAuthorListFileAuthors.dialog('option', 'position', {

my: 'left top',

at: 'left+10%' + ' top+' + (authorListAuthorsCount > 1 ? '40' : '45'),

of: $contentBody

});

### Pages

#### + AuthorListFileAuthors.ashx

This ashx file is added for this spec and defines the Web Handler for the AuthorListFileAuthors dialog that is displayed as an overlay.

The class AuthorListFileAuthors is defined as follows:

public class AuthorListFileAuthors : BaseHttpHandler, IHttpHandler

The class implements the following methods:

public override void ProcessRequest(HttpContext context)

This method calls the base class method, sets the response context to ‘application/json’ and then loads the WebUserControl AuthorListFileAuthors and generates the HTML and serializes it in the HTTP response.

#### + AuthorListFileAuthors.ascx(.cs)

This is a Web User Control that contains the HTML for displaying the Author list. An Asp Panel and Asp Table are used to display the list of Author objects.

The class is defined as follows:

public partial class WebUserControlsAuthorListFileAuthors : System.Web.UI.UserControl

The following constructors are defined:

WebUserControlsAuthorListFileAuthors()

WebUserControlsAuthorListFileAuthors(List<AuthorListFileAuthors> authorList)

This constructor sets the local member variable with the provided author list.

The following methods are defined:

protected override void OnLoad(EventArgs e)

This method populates a table within an ASP panel on the User Control page that contains fields for each of the Author properties defined in the Author List File

### Misc

The following miscellaneous updates are implemented.

#### Web.config

The web.config file requires the following changes to support the new upload type.

A key is added to the *uploadSchemes* section as follows:

<uploadSchemes>

...

<add key="10" value="Aries.EditorialManager.Web.Command.Upload.UploadAuthorListFile"/>

</uploadSchemes>

## Parsing the Author List File - Matt

Parsing the Author File List is done as part of the file upload process. This section details the specific processes for how the file is parsed.

### Code

#### Aries.EditorialManager.Web.Command.Upload

The first step in parsing the file is determining the file type. The acceptable file types are comma separated text file, tab separated text file or Excel spreadsheet file.

Parsing the comma separated and tab separated text is done by opening the file and reading the first few bytes.

##### + UploadAuthorListFile.cs

The parsing occurs in the AEM.Web.Command.Upload.UploadAuthorListFile class in the method:

protected override void OnProcess()

The following method is used to parse the comma separated or tab separated file:

Protected void ParseTextAuthorListFile()

This method uses File.ReadLine() to extract the first line from the source (the uploaded or the converted Excel file) which is used to determine the column order. For both the column row and each Author row, String.Split is used to break apart the provided values. The column order is stored in a hashtable of key = column value=index.

The remainder of the file is read, line by line and each line is stored in a list of *UploadedAuthorInfo* objects.

This list is then returned via the Ajax call in the *UploadAuthorStatusInfo*.

## Author List File Results Page - Matt

### Code – Andrew (completed)

#### Aries.EditorialManager.Framework.Journal.PeopleObjects.Person

##### Mapper.cs

**+ FindByPersonalIdentifierOrcid(List<string> orcIDs, bool authenticated)**

New function (along with supporting CollectionQuery and stored proc) that takes a list of orcid strings and returns all people records with matching orcids. The authenticated parameter controls whether or not we limit ourselves to authenticated orcids only.

**+ FindByEmailMatch(List<string> emails)**

New convenience function that takes a list of strings and invokes FindByEmailMatch(string email) after transforming the string list into a comma separated list (or however else that’s stored)

#### Aries.EditorialManager.Framework.Journal.AuthorObjects.AuthorInvitationStats

##### Mapper.cs

+ FindByPeopleIDs(List<int> peopleIDs)

New function that takes a list of people IDs and finds author invitation statistics records for those people.

##### Entity.cs

Must add DateLastInvited property, driven by DATE\_LAST\_INVITED from the underlying table. Also, of course, supporting code in mapper and stored procs.

#### Aries.EditorialManager.Framework.Journal.AuthorObjects.InvitedAuthor

##### Mapper.cs

+ FindByPeopleIDs(List<int> peopleIDs)

New function that takes a list of people IDs and finds all author invitation records for those people.

#### Aries.EditorialManager.Framework.Journal.SubmissionObjects.Submission

##### Mapper.cs

**+ Dictionary<int, int> GetUnsolicitedSubmissionCounts(List<int> peopleIDs)**

New function that takes a list of people IDs and returns the count of “unsolicited” submissions in the pipeline for each given person, as per logic in UDF\_getAuthorInvitationStatistics:

SELECT COUNT(1)

FROM dbo.DOCUMENT D WITH (NOLOCK)

INNER JOIN dbo.ROLEAUTH RA WITH (NOLOCK)

ON RA.DOCUMENTID = D.DOCUMENTID

WHERE D.RELATED\_DOCUMENTID IS NULL

AND RA.PEOPLEID = PEOPLE.PEOPLEID

AND D.DDISPOSIID IS NULL

The return value needs to map people IDs to the ‘unsolicited submission count’.

## AuthorCandidate data structure - Jean

### Code

#### Aries.EditorialManager

##### AuthorCandidate.cs

Constructor:

**AuthorCandidate(UploadedAuthorInfo authorFromFile, int authorCandidateID)**

* Set FirstName,LastName,EmailAddress,ORCID to values incoming from authorFromFile.
* Set CandidateID to incoming authorCandidateID.
* Set Selected property based on incoming UploadedAuthorInfo.Invite
* Set NumberOfInvitationLetters to 1 if incoming UploadedAuthorInfo.Invite == true
* Set any author notes to UploadedNotes.

**AuthorCandidate()**

Empty constructor necessary for cache serialization.

Properties:

**CandidateID** - Represents the manufactured ID of this candidate in the listing of matched candidates

**EmailAddress** – \_authorFromFile.EmailAddress

**FirstName** - \_authorFromFile.FirstName

**Hash**

* FirstName,LastName,EmailAddress,ORCID
* Makes for easier searching of dictionaries/lookups

**ImportMatchesMultiplePeople** – indicates this person could match multiple em people

**LastName** - \_authorFromFile.LastName

**MatchedPersonID** – MatchedPerson.IdentityValue

**MatchedPersonIsBoardMember** – MatchedPerson.Board

**MultipleImportsWithSameEmail**

* Import record has no EM matches – one or more
* One of the email addresses is duplicated in another non matcheable import record
* Not otherwise a duplicate record in the file

**NumberOfInvitationLetters**

* Number of invitation letters to send
* Initially set at constructor to 1

**ORCID** - \_authorFromFile.OrcID

**ORCIDIsAuthenticated** – MatchedPerson.HasAuthenticatedOrcid

**PotentiallyMatchedPeopleID**

* Each potential match will have a separate author candidate object created
* For this candidate object, the potential person match

**Selected**

* is this author selected to have an invitation letter sent
* initially determined by \_authorFromFile.Invite

**UploadedNotes** – list of notes from the file

## Imported Author Manager class – Jean

This class encapsulates functionality needed to:

* build lists of AuthorCandidates
* mass-proxy register unregistered authors and generate InvitedAuthorsXml (including notes!) so that it can be harmlessly parsed by AuthorConfirmAndCustomizeEMPage.ParseSelectedAuthorsXml

### Code

#### Aries.EditorialManager

##### ImportedAuthorManager.cs

All static class

**CONSTANTS**:

authorCandidatesCacheDataLocation – “AuthorCandidateList”

uploadedAuthorListCacheDataLocation – “UploadedAuthorList”

authorListFileInfoCacheDataLocation – “AuthorListFileInfo”

**PROPERTIES**:

**int NumberOfCandidates** – returns number of candidates created after call to MatchUploadList.

**METHODS**:

**private static bool AlreadyMatched\_HashCheck(UploadedAuthorInfo uploadedAuthor, List<AuthorCandidate> processedAuthors)**

* Check if current upload author has been already processed, based on the Hash property and is, ergo, a duplicate from the file.

**private static bool DuplicateEntityCheck(AuthorCandidate matchedAuthorCandidate, List<AuthorCandidate> processedAuthors)**

* Check that processed authors does not already have an author candidate that has been em entity matched same as incoming author candidate. If so, don’t create a new author candidate, just aggregate number of invitations and upload notes.

**private static SessionCacheManager GetSessionCacheManager(string journal, string emSessionID)**

* Convenience function to return a session cache manager for retrieving or updating data in the session cache.

**public static List<AuthorCandidate> GetAuthorCandidatesFromCache(string journal, string emSessionID)**

* gets a cache manager and retrieves author candidates from cache

**public static Ilookup<int,AuthorInvitation> GetAuthorInvitations (List<int> authorIDs, DatabaseMapperManager mapperManager)**

* call to InvitedAuthorMapper.FindByPeopleIDs to get author invitation data.

**public static Dictionary<int, PersonEntity> GetAuthors(List<int> authorIDs, DatabaseMapperManager mapperManager)**

* call to PersonMapper.FindByPeopleIDS to get em person data.

**public static Dictionary<int,AuthorInvitationStatsEntity> GetAuthorStats(List<int> authorIDs, DatabaseMapperManager mapperManager)**

* call to AuthorInvitationStatsMapper.FindByPeopleIDs to get author invitation stats data.

**public static Dictionary<int,int> GetUnsolicitedSubmissionsCount(List<int> authorIDs, DatabaseMapperManager mapperManager)**

* call to SubmissionMapper.GetUnsolicitedSubmissionCounts to get duh, unsolicited submission counts.

**public static List<AuthorCandidate> GetMatchedCandidates(string journal, string emSessionID, int startIndex, int endIndex)**

* meant to be called by results page to return the current page’s author candidates
* gets a cache manager and retrieves author candidates from cache and filters based on start and end index passed in.

**public static List<UploadedAuthorInfo> GetUploadedAuthorListFromCache(string journal, string emSessionID)**

* gets a cache manager and retrieves uploaded author info from cache

**public static void MatchUploadList (DatabaseMapperManager, List<UploadedAuthorInfo> uploadedAuthors)**

* performs matching algorithm and returns listing of author candidates to cache

**public static void PlaceAuthorCandidatesInCache(List<AuthorCandidate> authorCandidates, string journal, string emSessionID)**

* gets a cache manager and places passed in list to cache

**public static void PlaceUploadedAuthorsInCache(List<UploadedAuthorInfo> uploadedAuthors, string journal, string emSessionID)**

* gets a cache manager and places passed in list to cache

**public static void ProxyRegSelectedUnregisteredUsers(string journal, string emSessionID, DatabaseMapperManager mapperManager, int operatorID)**

* gets a cache manager and retrieves current list of author candidates from cache
* In loop
  + proxy registers all selected/non-matched (Selected == true, HasMatchedPerson == false) persons in em and returns created person entities using RegistrationManager.ProxyRegisterUser.
  + Updates author candidate list with returned person entity.
* Places the author candidates back into cache.

**public static void SaveGridChanges(List<AuthorCandidateGridData> gridData, string journal, string emSessionID)**

* retrieves current author candidates list from cache and updates data based on passed in author candidate grid data from the results grid.
* Returns the updates data to the cache
* Allows saving from either the server (the results page) or from the client.

**public static string SerializeCandidatesForConfirmAndCustomize(string journal, string emSessionID)**

* gets a cache manager and retrievs current list of author candidates from cache
* creates and populates an xml structure to be passed to the ConfirmAndCustomizeInvitedAuthors page with data from author candidates list.
* Creates one ‘record’ for each invitation – if author candidate has more than one invitation, breaks into single invitation records.
  + Each invitation gets ONE of the UploadedNotes in the *note* element of the record.

Xml data structure:

<PeopleSearchXml>

<InvitedAuthorsXml>

<author>

<numLetters>AuthorCandidate.NumberOfInvitationLetters

</numLetters>

<id>AuthorCandidate.MatchedPersonID</id>

<note>AuthorCandidate.UploadedNotes</notes>

</author>

</InvitedAuthorsXml>

</PeopleSearchXml>

**Support Objects**:

**AuthorCandidateGridData** – object indicating data that can change in author candidate grid.

intCandidateID – candidate id given when list was created in *MatchUploadList*

bool Selected – whether candidate is selected for invite in the grid

int NumberOfInvites – number of invites indicated in the grid

Constructor:

**public AuthorCandidateGridData (int candidateID, bool selected, int numberOfInvites)**

**public AuthorCandidateGridData()** – necessary for JSON serialization

##### Matching Logic

**MatchUploadList(DatabaseMapperManager mapperManager, string journal, string emSessionID)**







## AuthorCandidateGrid - Jean

### Data

* + - 1. TRANSLATION
* Pages.Editorial.AuthorCandidateGrid.AuthorInvited – “Author Invited:”
* Pages.Editorial.AuthorCandidateGrid.ConfirmedMatch – “Confirmed Match:”
* Pages.Editorial.AuthorCandidateGrid.InformationFromFile – “Information from File:”
* Pages.Editorial.AuthorCandidateGrid.MultipleMatchedEmails – “Multiple imported Author records contain the same email address. Please select only the record you wish to invite.”
* Pages.Editorial.AuthorCandidateGrid.NoMatchFound – “No Match Found; Account will be proxy-registered.”
* Pages.Editorial.AuthorCandidateGrid.Note – “Note:”
* Pages.Editorial.AuthorCandidateGrid.Declined – “Declined:”

### Code

#### Aries.EditorialManager

##### Constants.cs

* **CURRENT\_UPLOAD\_AUTHORS\_RESULTS\_PAGE\_CACHE\_KEY** (UploadAuthorsResults\_CurrentPage)– to be used to track, in the cache, on which page the candidate grid is at currently so that, upon refresh (paginator will refresh on page change and page size) that value can be reset in the paginator.

#### Aries.EditorialManager.Web

##### WholeNumbersOnly\_KeyDown.js

* Add to Resources
* Contains function WholeNumbersOnly\_KeyDown which will be attached to keydown event of invitation number text boxes.

#### common\WebUserControls

##### AuthorCandidateGrid.ascx

**startup()**

* function to be called at document.ready
* hook invite number text boxes keydown event to WholeNumbersOnly\_KeyDown function that will be placed by Page\_Load – selector(.inviteTextBox)
* hook selected check boxes change event to selectCheckbox\_Changed

**selectCheckbox\_Changed(e)**

* handles change event of select check boxes
* selector (.cbSelected)
* Enable invite text box and place value to 1 if checked
* Disable invite text box and clear value if unchecked

**table UploadedAuthorResultsGrid**

* table containing author candidate rows.
* All rows have class *candidateRow*.
* Row uniquely identified through *data-candidate* attribute value of which is the AuthorCandidate.CandidateID.
* All select checkboxes have class *colSelected*.
* Select checkbox uniquely identified through *data-candidate* attribute value of which is the AuthorCandidate.CandidateID.
* All invitation text boxes have class *invitesTextBox*.

**TopPaginator/BottomPaginator**

* Top and bottom of grid – GridPaginator.ascx

##### AuthorCandidateGrid.ascx.cs

To be similar to InvitedReviewersGrid.ascx.

Uses common/WebUserControls/GridPaginator.ascx

Top div includes GridPaginator

Middle div is HTMLTable – *UploadedAuthorResultsGrid*

Bottom div includes GridPaginator

**Private Fields**:

**\_currentPage** – current page for grid paginators

**\_pageSize** – currently chosen page size for grid paginators

**\_confirmedMatchText** – Pages.Editorial.AuthorCandidateGrid.ConfirmedMatch

**\_informationFromFileText** – Pages.Editorial.AuthorCandidateGrid.InformationFromFile

**\_multipleMatchedEmailsHTML** – Pages.Editorial.AuthorCandidateGrid. MultipleMatchedEmails

**\_noMatchFoundHTML** – Pages.Editorial.AuthorCandidateGrid.NoMatchFound

**\_noteText** – Pages.Editorial.AuthorCandidateGrid.Note

**\_agreedAndAwaitingSubmissionText** – Pages.Editorial.ResultsForAuthors. TextAgreedAndAwaitingSubmission

**\_invitedSubmissionReceivedText** – Pages.Editorial.ResultsForAuthors. TextInvitedSubmissionsReceived

**\_uninvitedBeforeAgreeingText** – Pages.Editorial.ResultsForAuthors. TextUnivitedBeforeAgreeing

**\_uninvitedAfterAgreeingText** – Pages.Editorial.ResultsForAuthors. TextLastInvitedSubmissionAgreed

**\_lastSubmissionAgreed** – Pages.Editorial.ResultsForAuthors. TextLastInvitedSubmissionAgreed

**\_lastSubmissionReceived** – Pages.Editorial.ResultsForAuthors. TextLastInvitedSubmissionReceived

**\_lastSubmissionDecline** – Pages.Editorial.ResultsForAuthors. TextLastInvitedSubmissionDeclined

**\_avgDaysToSubmit** – Pages.Editorial.ResultsForAuthors.TextAverageDaysToSubmit

**\_dateLastInvitedText** – Common.Text.DateLastInvited

**\_outstandingInvitationsText** – Pages.Editorial.ResultsForAuthors. TextOutstandingInvitations

**\_agreedInvitationsText** – Pages.Editorial.ResultsForAuthors.TextAgreed

**\_declinedInvitationsText** – Pages.Editorial.AuthorCandidateGrid.Declined

**\_uninvitedInvitationsText** – Pages.Editorial.ResultsForAuthors.TextUnivited

**\_totalInvitationsText** – Pages.Editorial.ResultsForAuthors.TextTotalInvitations

**\_yes** – Common.Text.Yes

**\_no** – Common.Text.No

**\_peopleFlagManager** – Aries.EditorialManager.FlagManager – for use in creating custom flags of successfully matched persons.

**Properties**:

**protected EMPage OwnerPage** – allows convenience to containing EMPage object of containing page.

**public EditorRoleManager** Aries.EditorialManager.Framework.Journal.RoleManager.EditorRolePermissions.Entity – \_peopleFlagManager will need this to determine if flags can be shown.

**public int EndIndex** – holds the paginator’s end index value when fired.

**public bool PaginationChanged** – allows containing page to know that pagination has changed and retrieve of a new set of data for the grid is necessary.

**public int RecordsCount** – set only – allows external setting of paginator RecordsCount property.

**public int StartIndex** – holds the paginator’s start index value when fired.

**Paginator Events**:

**GridPaginator\_PageSelected**

* Fires when the grid paginator control’s page number changes, either by pressing the actual number page link or pressing the next/previous buttons.
* Set paginators.CurrentPage to e.SelectedPage
* Do a DataBind on both paginators – not sure why this would be necessary although InvitedReviewersGrid.ascx is doing the same so, when in Rome…
* Capture the start and ending index indicated by the paginator into StartIndex and EndIndex properties.
* Store the current page value in the cache to be retrieved once the owner page refresh occurs, which will be caused by paginator. Will allow the containing page to indicate to the paginator, what page it is currently to display. Save in location name created in Constants.cs, previously.
* Set PaginationChanged = true – used as an indicator in the containing page that pagination has changed and that it should be reset (essentially, there is now a ‘new’ paginator that needs to be recreated/repopulated.

**GridPaginator\_PageSizeSelected**

* Capture the selected page size, if it exists from e.SelectedPageSize or, if it doesn’t, default to a page size of 10 records - place into pageSize local variable.
* Set PageSizeForGrid in StateManager – not quite sure why, but, again, InvitedReviewersGrid is doing it so…
* Set PageSize of both paginators
* Do a DataBind on both paginators – again, not sure why but if you saw someone jump off of a bridge, you’d do it too, right?
* Capture the start and ending index indicated by the paginator into StartIndex and EndIndex properties for later usage.
* Store the current page value in the cache to be retrieved once page refresh caused by paginator
* Set PaginationChanged = true

**Page Events**:

**Page\_Load**

* Retrieve translation values from resource loader into local fields.
* Create html string for multiple matched emails to be used in grid creation.
* Create html string for no match found to be used in grid creation.
* Set \_currentPage retrieved from cache (CURRENT\_UPLOAD\_AUTHORS\_RESULTS\_PAGE\_CACHE\_KEY). – set it to 1 if there is no current page – will be used in Page\_PreRender to reset the CurrentPage of the paginators.
* Set \_pageSize from StateManager.PageSizeForGrid (gues this is why we saved it previously in the paginator events, yes?) – this will be used in Page\_PreRender to reset the PageSize property of the paginators.
* Register any client scripts
  + CUSTOM\_FLAGS\_SCRIPT – for grid creation and adding people flags where appropriate
  + WHOLE\_NUMBERS\_ONLY\_TEXTBOX\_KEYDOWN – created a new function to handle invite text box KeyDown event to prevent all but numeric input.
  + ORCID – includes functions to popup a page to show ORCID data, if authenticated.

**Page\_PreRender**

* Only if PaginationChanged:
  + Set paginators.CurrentPage = \_currentPage
  + Set paginators.PageSize = \_pageSize
  + Do a DataBind on paginators

**Page Methods**:

**ShowData(**

**List<AuthorCandidate> candidatesToShow,**

**Dictionary<int, PersonEntity> matchedAuthors,**

**Ilookup<int,AuthorInvitation> authorInvitationsList,**

**Dictionary<int, AuthorInvitationStatsEntity> authorInviteStats,**

**Dictionary<int,int> unsolicitedSubmissionCounts)**

Externally available function to allow containing page to display the correct records requested by the paginator. Containing page will determine, based on StartIndex and EndIndex, the candidate records to request in the grid.

Builds html to append to UploadedAuthorResultsGrid (HTMLTable).

Each table row will contain attribute *data-candidate* with value being AuthorCandidate.CandidateID.

Various private helper functions:

*BuildAuthorStats* – builds the author statistics cell for a particular candidate.

*BuildConfirmedPerson* – builds the html to display a candidate that has been successfully matched against an EM entity, in the AuthorName cell. Uses \_flagManager to get custom people flags/creator, if possible

*BuildInfoFromFile* – builds the file info portion of the AuthorName cell.

*BuildInviteStats* – builds the invite statistics cell for a particular candidate.

## UploadedAuthorSearchResults - Jean

### Data

* + - 1. TRANSLATION
* Pages.Editorial.UploadedAuthors.UploadedAuthorSearchResults – “Uploaded Author Search Results”
* Pages.Editorial.UploadedAuthors.AuthorListFile – “Author List File”
* Pages.Editorial.UploadedAuthors.CandidateHeading – “Author Candidates”
* Pages.Editorial.UploadedAuthors.Instructions – “Check the box for each person you want to invite to submit and then proceed.”
* Pages.Editorial.UploadedAuthors.HeaderText = “’Author Candidates’ Results – Manuscript “
* Pages.Editorial.UploadedAuthors.ProxyWaitText = “Please wait while proxy-registration is performed”
* Pages.Editorial.UploadedAuthors.RemoveFileWarning = “This will remove the uploaded file and discard any existing selections. Are you sure?”
* Pages.Editorial.UploadedAuthors.ProxyRegError = “The following error occurred while attempting to proxy register authors:”
* Pages.Editorial.UploadedAuthors.ProxyRegErrorTitle = “Proxy registration error”
* Pages.Editorial.UploadedAuthors.RemoveFileError = “The following error occurred while attempting to remove uploaded author data:”
* Pages.Editorial.UploadedAuthors.RemoveFileErrorTitle=”Remove file error”
* Pages.Editorial.UploadedAuthors.FileWarningTooltip = “This file contains incomplete or invalid records. Click for more information.”

### Page

#### UploadedAuthorSearchResults.aspx

Include AuthorCandidateGrid.ascx

**startup()**

* function to be called at document.ready

**proceed\_click**

* Creates an JSON object to mimic ImportedAuthorManager.AuthorCandidateGridData for each author candidate grid row (AuthorCandidateGrid control) – into array.
* Opens jquery proxy wait dialog ($dialogProxyWait) – modal
* Makes ajax call to UploadedAuthorSearchResults.aspx/ProxyRegSelectedAuthorCandidates – passing array created previously to proxy register any necessary selected candidates.
* Proxy register fails – return false to prevent proceed button server event from firing.
* Proxy register success – remove proxy wait dialog and return true to allow proceed button server event to fire.

**removeFile**

* Show warning message with OK/Cancel.
* ajax call to RemoveUploadedAuthorsFileData that calls to remove all uploaded info from cache.
* Return to file upload page.

**showOverlay** – shows file list authors page/overlay through function $fn.authorListFileAuthors.

**paginatorClick** – save data before paginator makes server post.

**saveGridData** – pass grid data to SaveGridChanges on server via ajax to save any changes.

#### UploadedAuthorSearchResults.aspx.cs

Took paginator/grid control handling direction from SearchReviewers.aspx which includes the InvitedReviewersGrid.ascx – on which the AuthorCandidateGrid.ascx is based.

**Properties:**

**UploadAuthorListURL** – builds url to return to upload author list.

**Page Events**:

**Page\_Load**

* Get submission data using EMPage.CurrentDocumentID
* Set header text to Pages.Editorial.UploadedAuthors.HeaderText + Submission.ManuscriptNumber + line break + Submission.Title.
* Create file data row in tblFileData.
* Do a DataBind – not sure if this is necessary but other search results pages with grid controls on them do it so…

**Page\_PreRender**

* If AuthorCandidateGrid.PaginationChanged, call SaveChanges() – save current grid page changes to cached dataset.
* Gather necessary author candidates from cache, depending on current AuthorCandidateGrid.StartIndex and AuthorCandidateGrid.EndIndex
* Gather matched author data.
* Gather author invitation data.
* Gather author stats data.
* Gather unsolicited submission counts data.
* Pass gathered data to AuthorCandidateGrid.SetData to populate the current grid page.

**Button Events**:

**btnProceed\_Click**

* proxy registration should have happened by now, so all selected author candidate items should have a person object by now. As well, all grid changes will have taken place prior to proxy registration.
* CurrentSession.CurrentPeopleSearchXML = ImportedAuthorManager.SerializeCandidatesForConfirmAndCustomize – creates xml and places it in session object location for use by ConfirmAndCustomizeInvitedAuthors.aspx.
* Call ConfirmAndCustomizeInvitedAuthors.aspx?authorSearchMode = Constants.AUTHOR\_SEARCH\_MODE\_UPLOAD\_AUTHOR\_LIST.

**Page Functionality**:

**private string BuildFileGrid()**

* return html string for file data grid row
* show warning graphic if import file errors
* get data from ImportedAuthorManager.GetAuthorListFileFromCache()

**Web Methods**:

**public static void ProxyRegSelectedAuthorCandidates (List<ImportedAuthorManager.AuthorCandidateGridData gridData)**

* Create a new EMData object
* Call *ImportedAuthorManager.ProxyRegSelectedUnregisteredUsers*.

**public static void RemoveUploadedAuthorsFileData()**

* Create a new EMData object
* Call ImportedAuthorManager.ClearAuthorListFileCache

**public static void SaveGridChanges(List<ImportedAuthorManager.AuthorCandidateGridData> gridData)**

* Create a new EMData object.
* ImportedAuthorManager.SaveGridChanges

## ConfirmAndCustomizeInvitedAuthors - Jean

### Page

#### AuthorConfirmAndCustomizeEMPage.cs

This is basis of ConfirmAndCustomizeInvitedAuthors.aspx page.

**New**:

**private string \_authorSearchMode**

holds passed in authorSearchMode from query string.

**private static List<KeyValuePair<int,string>> \_peopleNotes**

will hold a listing of peopleIDs (KEY) and related note passed in (VALUE) from the UploadedAuthorSearchResults page through note element in the author element of <InvitedAuthorsXml> portion of <PeopleSearchXml>.

See *ParseSelectedAuthorsXml* below for population.

See *GetAuthorInvitationNotes* below for usage.

**InitializeBaseObjects()**

*fromSearchResultsPage = TRUE*

determined if Request does not have ‘fromLetterCustomizationPage’ in query string which should be the case when called by *UploadedAuthorSearchResults.aspx*

*\_pageXml = CurrentSession.CurrentPeopleSearchXML*

This should already be populated in the call from *UploadedAuthorsSearchResults*.*aspx at the* btnProceed\_Click event by calling *ImportedAuthorManager.SerializeCandidatesForConfirmAndCustomize* – which will populate the <*InvitedAuthorsXml*> portion of the <*PeopleSearchXml*>

**CHANGE:**

* Capture authorSearchMode from queryString into \_authorSearchMode private field to be used later in StoreSessionVariables() and InitializeHiddenFields
* Use \_authorSearchMode in InitializeHiddenFields parameter – fromSearchResultsPage == true && \_authorSearchMode != Constants.AUTHOR\_SEARCH\_MODE\_UPLOAD\_AUTHOR\_LIST
* In the case of coming from the UploadedAuthorSearchResultsPage, fromSearchResults will be TRUE and second condition will be FALSE, so, parameter value passed to InitializeHiddenFields will be false, meaning that the majority of the hidden fields will be populated either from the current session object and/or the \_pageXml object, which is the correct way of populating these fields in this case.

**StoreSessionVariables()**

* When \_authorSearchMode = Constants.AUTHOR\_SEARCH\_MODE\_UPLOAD\_AUTHOR\_LIST (meaning the page has been arrived at from the UploadedAuthorSerachResults.aspx page) –
  + set the eventID node of <PeopleSearchXml> to EventType.InviteAuthorsForProposal.
  + DO NOT populate the \_pageXml<InvitedAuthorsXml> - it will already exist as it was set via the session object when we serialized the selected author candidates at ImportedAuthorManager.SerializeCandidatesForConfirmAndCustomize.
* Set AuthorSearchMode node of <PeopleSearchXml> to \_authorSearchMode

**InitializeHiddenFields**

Just populate hidden field hdnAuthorSearchMode from \_authorSearchMode captured at InitializeBaseObjects, no matter the input parameter.

**ParseSelectedAuthorsXml**

This function will be called within the ConfirmAndCustomizeInvitedAuthors.Page\_Load event.

At this event, it is asking for just an array of people ids that came in through CurrentSession.CurrentPeopleSearchXML, that it can use in the call to \_ELM.AddRecipients.

Take advantage of this call and the loop through all selected author candidates and capture the <note> element value in a KeyValuePair (KEY = peopleID & VALUE = <note>) and place in \_peopleNotes list for later use in creating populating the notes text box with a value in *GetAuthorInvitationNotes*.

**GetAuthorInvitationNotes**

This function is arrived at via

Page\_Load->generateRecipientListHTML->

GenerateLettersTableCell->GetAuthorInvitationNotes

GenerateLettersTableCell creates the author cell which may include a notes text box if it is visible through config settings.

GenerateLettersTableCell calls GetAuthorInvitationNotes for a value for the notes text box if it cannot find a note value in an existing invitation (\_xmlManager.InvitationNotes) to see if one can be gathered.

In the case of arrival to this page via UploadedAuthorSearchResults, there will be nothing in \_xmlManager, since there will not yet be any invitations, so GetAuthorInvitationNotes will be called for each created row.

In this case, after the check of the database for a note through \_authorInvitationManager.GetInvitedAuthorForDocIDAndEventLetterID (which will return null in this case) – go through the \_peopleNotes field KeyValuePair listing created previously (at ParseSelectedAuthorsXml), and see if there is a note that can be found related to the personID (meaning there was a related note in the uploaded authors file for this person). Use this value at the existing note.

#### ConfirmAndCustomizeInvitedAuthors.aspx.cs

**cancelButtonClick**

* Server.Transfer to UploadedAuthorSearchResults.aspx

# DB Content

FIELD

Short Description:

Long Description:

Engineering Notes:

# Translation Tokens

Common.Buttons.Import - Import

Common.Text.FirstName - FIRSTNAME

Common.Text.Ignore - Ignore

Common.Text.Invitation - Invitation

Common.Text.LastName - LASTNAME

Common.Text.Note - NOTE

Pages.Editorial.AuthorCandidateGrid.AuthorInvited - Author Invited

Pages.Editorial.AuthorCandidateGrid.ConfirmedMatch - Confirmed Match

Pages.Editorial.AuthorCandidateGrid.Declined - Declined:

Pages.Editorial.AuthorCandidateGrid.InformationFromFile - Information from File:

Pages.Editorial.AuthorCandidateGrid.MultipleMatchedEmails - [Error message for Multiple Emails]

Pages.Editorial.AuthorCandidateGrid.NoMatchFound - [Error message for no matching user]

Pages.Editorial.AuthorCandidateGrid.Note - Note:

Pages.Editorial.UploadAuthorList.EmailHeader - EMAIL

Pages.Editorial.UploadAuthorList.FileBeingValidated - [Validation message with spinner gif]

Pages.Editorial.UploadAuthorList.LegendUploadAuthorList - Upload Author List

Pages.Editorial.UploadAuthorList.NoAuthorListFileUploaded

- [No author list file label text]

Pages.Editorial.UploadAuthorList.OrDragAndDropAFileHere - [Drag/Drop area text]

Pages.Editorial.UploadAuthorList.OverlayInstructions - [Overlay Instructions before importing]

Pages.Editorial.UploadAuthorList.OverlayInstructionsPostImport

- [Overlay instructions for candidates grid/after file import is completed]

Pages.Editorial.UploadAuthorList.PageInstructions

- [Page instructions for upload page and info icon on Author Search page]

Pages.Editorial.UploadAuthorList.TooltipEmailInvalid - [Tooltip text for invalid email on overlay]

Pages.Editorial.UploadAuthorList.TooltipEmailRequired - [Tooltip text for missing email on overlay]

Pages.Editorial.UploadAuthorList.TooltipFirstNameRequired - [Tooltip text for missing first name]

Pages.Editorial.UploadAuthorList.TooltipLastNameRequired - [Tooltip text for missing last name]

Pages.Editorial.UploadAuthorList.TooltipOrcIDInvalid - [Tooltip text for invalid ORCID value]

Pages.Editorial.UploadAuthorList.WarningExceedsAuthorLimit - [Alert text for too many authors in ALF]

Pages.Editorial.UploadAuthorList.WarningGeneralError - [Alert text for general/unknown error]

Pages.Editorial.UploadAuthorList.WarningMissingReqFields

- [Alert text for missing required columns in ALF]

Pages.Editorial.UploadAuthorList.WarningNoAuthorRecords - [Alert text for ALF with no author records]

Pages.Editorial.UploadAuthorList.WarningUnsupportedFileExt - [Alert text for ALF with incorrect extension]

Pages.Editorial.UploadedAuthors.AuthorListFile - Author List File

Pages.Editorial.UploadedAuthors.CandidateHeading - Author Candidates

Pages.Editorial.UploadedAuthors.FileWarningTooltip - [Tooltip text for ALF with errors]

Pages.Editorial.UploadedAuthors.HeaderText - [Author candidates results header text]

Pages.Editorial.UploadedAuthors.Instructions - [Instructions for Uploaded Authors page]

Pages.Editorial.UploadedAuthors.ProxyRegError - [Alert text for proxy registration failure]

Pages.Editorial.UploadedAuthors.ProxyRegErrorTitle - [Title for proxy registration error alert]

Pages.Editorial.UploadedAuthors.ProxyWaitText - [Wait message for proxy registration]

Pages.Editorial.UploadedAuthors.RemoveFileError

- [Error message for removing uploaded authors failure]

Pages.Editorial.UploadedAuthors.RemoveFileErrorTitle - Remove file error

Pages.Editorial.UploadedAuthors.RemoveFileWarning - [Confirmation box text for discarding ALF file]

Pages.Editorial.UploadedAuthors.UploadedAuthorSearchResults- Uploaded Author Search Results

Pages.Editorial.UploadedAuthors.WarningNoAuthorsToImport

- There are no Authors specified to Import. Please select at least one author record.

**~ End of Specification ~**