Intralinks Integration Adapter Sample XML Files

Reference Guide

Intralinks 24x7x365 support US: + (1) 212 543 7800 UK: +44 (0)20 7623 8500.
See Intralinks login page for other national numbers

Table of Contents

Click below to navigate to the chapter or section you want.

Chapter 1: Introduction to the Intralinks Integration Adapter

The Integration Adapter can be used to transfer documents and/or contact information from a back-end system to Intralinks, ensuring that the information in both systems is up to date and accurate and reducing the amount of time spent on data entry.

The Integration Adapter can also be used to download documents and associated metadata from Intralinks exchanges and store it at a location of your choice.

All Integration Adapter functions are configured using an XML file. The same XML file can be used to perform file upload and/or download functions.

About this guide

This guide is a reference guide that includes sample XML files. See additional guides for reference information on the data dictionary and troubleshooting information.

Instructions and additional information about using ILIA can be found in the Knowledgebase of the **Intralinks Support Center.**

This guide is intended for:

- IT personnel who are responsible for implementing the Integration Adapter.
- The system administrator, the person within your organization who will be using the Intralinks Integration Adapter to upload information to Intralinks exchanges or to download documents and folders from Intralinks exchanges.

Chapter 2: Sample XML Files

Introduction

The Intralinks Integration Adapter transfers information using XML files. These files must be formatted using the schema described in this chapter and the next. XML must be in UTF-8 format.

- The operation types supported in the input XML file are:
  - **Create**—creates exchanges, users, groups, documents, and folders specified in that batch.
  - **Remove**—removes documents, folders, users and empty groups from specified exchanges.
  - **Update**—updates documents or groups and the role of existing users in that batch.
  - **Synchronize**—ensures that users on Intralinks exchanges match the users in your organization's back-end system.
  - **Rollback**—will undo any additions to documents and folders plus any associated metadata on Intralinks exchanges introduced by the output file specified in the rollback command.
  - **Download**—downloads documents and folders from exchanges and saves them on a local system.

    A full download downloads all files, even those that may already exist in the storage location.

    An incremental download will compare the requested download with what already exists in the storage location and download those items that have not already been downloaded.

    A timebound download allows you to download documents that were added, documents that were modified or both new and modified documents within a specified time frame.
  - **Close**—closes an exchange.
  - **Archive**—orders an archive copy of the exchange.
- Each input file can have multiple batches.
- Each batch can have multiple jobs.
- A job can have only one operation type.
- A job can have multiple exchanges.
  - However, the same exchange should not appear more than once in the same XML Input file.
- Each exchange can have multiple groups, users, folders and documents.

Sample XML files

Sample XML files are provided below.

Date format

- Setting a date format

Exchanges

- Create an exchange
- Add groups and users to an exchange
- Set the public/private content value
- Change the phase on an existing exchange
- Close an exchange
- Order an archive

Working with documents, folders and reports

- Create folders
- Add folders and documents to an exchange
- Add document with a custom field to an exchange
- Add documents with notes
- Update documents
- Update folder properties
- Set ownership of a new document
- Change ownership of an existing document
- Set sort order of folders and subfolders
- Update the existing sort order of folders and subfolder
- Set sort order for documents
- Update sort order for documents
- Filter a list of documents using the “Submitted By” or “Submitted On” fields
- Filter a list of documents using the “Effective Date Range” field

Removing folders, and subfolders and documents

- Removal scenarios
- Remove documents and folders
- Remove documents and leave folder
- Remove a list of documents and folders by ID
- Remove a list of documents by specified criteria

Folder and document permissions

- Set permissions for a folder
- Replace permissions for a folder
- Set permissions for a document
- Replace permissions for a document
- Permission documents to the All Users group

Groups and users

- Synchronize users in an exchange
- Synchronize groups and users in an exchange
- Update user roles on an exchange
- Set member roles of new users
- Set member roles of existing users
- Add groups and users to an exchange
- Remove a user from an exchange
- Remove all group members from an exchange
- Remove two users from two different groups
- Sending custom remove alerts for deleted users
- Suppress the “Welcome to the Exchange” alert for new users
- Create multiple groups
- Remove groups from an exchange
- Change default folder for groups on create or update
- Updating group metadata

Custom Fields

- Assign values to exchange custom fields
- Update values to exchange custom fields
- Assign values to group custom fields
- Change custom fields of an existing document
- Enable/disable custom fields when creating an exchange
- Update the availability of custom fields at the exchange and group levels and then publish them

Reports

- Access report
- Access reports for an exchange

- Access report using a group filter
- Access report using a group custom field filter
- Access report using a user filter
- Access report using a folder name (ILP)
- Access report using a folder ID (ILP)
- Access report using a document name filter (ILP)
- Access report using a document created date range filter (ILP)
- Access report using a document creator filter (ILP)
- Access report using a document EffectiveDate filter (ILP)
- Access report using a document effective date range filter (ILP)
- Access report using a document extension filter (ILP)
- Access report using a document modified date range filter (ILP)
- Access report using a document custom field filter (ILP)
- Access report using the last alert failure filter
- Document-centric report
- User-centric report
- User Comparison report

Rollback

- Rollback files

Downloads

- Download all files from an exchange
- Download all exchange files using a file title filter
- Download all files from a folder
- Download files from a folder using a custom field filter
- Download all exchange files using a custom field filter
- Download all files from all exchanges

Miscellaneous Procedures

- Use PassThrough fields
- Set the user protection period in the Input XML file
- Override email notification
- Change the name of the Output XML file
- Question and Answer functionality

Setting a date format

Because date formats vary from country to country, the DateFormat attribute should be used to define the format of the date fields used in your input XML files. If the DateFormat is not specified the date strings will be interpreted using the local setting of your operating system.

Examples of DateFormat strings

Date Format	Input File Date
MM/dd/yyyy	05/03/2010
dd-MM-yyyy	03-05-2010
dd MMM, yyyy	05 May, 2010

For more information on date format strings, see http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx.

DateFormat attribute

The DateFormat attribute allows the customer to tell ILIA what the format of the date strings are in the input file. For example, in the input file they can provide an EffectiveDate for a document or they can provide a ModifiedDateFrom and ModifiedDateTo in the DocumentCriteria. They can write the date string as “05/06/2010”. Without specifying the DateFormat this can be interpreted in different ways depending on the current culture setting of the system running ILIA. In the US this would be May 6, 2010. In most European countries this would be June 5, 2010. In some countries it may not even convert properly (they may not recognize the slash separator).

In the US the default date format is “MM.dd.yyyy so the input file would be:

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="MM.dd.yyyy">

  <Identity UserId="user@company.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Download">

      <Workspace Id="131591">

        <Folders>

          <Folder Name="Test EIS" />

        </Folders>
```

```
        <DocumentCriteria ModifiedDateFrom="01.01.2009"
ModifiedDateTo="08.13.2010" />

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

In Germany the default date format is "dd.MM.yyyy" so the input file would be:

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="dd.MM.yyyy">

  <Identity UserId="user@company.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Download">

      <Workspace Id="131591">

        <Folders>

          <Folder Name="Test EIS" />

        </Folders>

        <DocumentCriteria ModifiedDateFrom="01.01.2009"
ModifiedDateTo="13.08.2010" />

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Create an exchange

Exchanges are created from a template (identified by an ID number). You will need to work with your Intralinks representative to get this template ID number.

- The Workspace “Name” attribute is optional except when creating a new exchange. The Workspace “ID” attribute is required in all other conditions to identify the appropriate exchange except when creating a new exchange.
- Set the name, description and host.
- You can set the phase of the exchange when you create it. The default phase is “Hold”.
- When you create an exchange using a template that has the Custom Fields setting enabled, the custom fields will be automatically published.
- You can't create two exchanges with the same name in the same input XML file.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="100">

    <Job Id="1" OperationType="Create">

      <Workspace Name="TestCreate3" Template="105401" Host="IL
Testing" Description="This is a test exchange" Create="true"
Phase="Open”>

        <Folders>

          <Folder Name="Folder 3" Create="true">

            <Documents>

              <Document Name="Test.pdf"
LocalPath="C:\temp\Temp.pdf" />

              <Document Name="CRA UI Screens.docx"
LocalPath="C:\Temp\CRA UI Screens.docx" />

            </Documents>

          </Folder>

        </Folders>

      </Workspace>
```

```
    </Job>
  </batch>
</ExecuteData>
```

Return to Sample XML files index.

Add groups and users to an exchange

- To create users you should provide the following information:
  - Valid email address
  - First name
  - Last name
  - Phone number
  - Organization
  - Role (optional)

    If a role is not provided, the default will be assigned based on your selection in the configuration settings.
- Please see Validation Rules in Appendix B before entering the items above.
- If a user already exists in the Intralinks Global User Directory, the information you enter here will not change the existing listing.
- Updating users is limited to changing a person's role on the exchange.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///Z:/ILIA/Schemas/
CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1234567" Name="Test Workspace">

        <Groups>

          <Group Name="P_1">

            <Users>
```

```
                <User EMailId="test_1@intralinks.com"
FirstName="EIS_test1"

                LastName="testser_1" Phone="6171111111"
Org="Intralinks"

                Role="Publisher" />

              </Users>

            </Group>

            <Group Name="P_2">

              <Users>

                <User EMailId="Test_2@intralinks.com"
FirstName="EIS_test_2"

                LastName="testuser_2" Phone="6171111111"
Org="Intralinks"

                Role="Publisher" />

              </Users>

            </Group>

          </Groups>

        </Workspace>

      </Job>

    </batch>

</ExecuteData>
```

Return to Sample XML files index.

Set the public/private content value

These examples will set the public/private content value of an exchange. The value will be set only if the public/private content setting is enabled for the exchange, otherwise it is ignored.

Note that as the user submitting the XML file, you can declare yourself as public or private (Pr/Pu) for a specific exchange. If not declaring yourself as public or private within an exchange, the default value “private” will be assigned. If you are creating a document, then you should declare the document as public or private.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1234567" PvPDeclaration="PrivateOnly">

        <Folders>

          <Folder Name="My Root Folder Test A\Folder
Level&#xD;&#xA;1\My Sub Folder">

            <Documents>

              <Document Name="Base Document Name"
LocalPath="C:\Documents and
Settings\ILIAUser\My&#xD;&#xA;Documents\Base Document Name.doc"
Visibility="private"></Document>

            </Documents>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Change the phase on an existing exchange

Phases are used to ensure that no one has access to an exchange until the owner is ready for them to use it. There are three possible phases: Hold, Open, and Preparation. Most exchanges are initially created in the Hold phase. You can only change the phase on an ILP exchange.

You also have the option to suppress the alert that is sent to exchange users who become active when the phase is changed.

This example changes the phase to “Open” for exchanges 1243581 and 2253599 and suppresses the alert for each exchange.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/
Documents%20and%20Settings/msieg/Desktop/
CSAInputXMLSchema.xsd">

  <Identity UserId="user@intralinks.com"/>

  <batch Id="6R9379HI">

    <Job Id="adhoc" OperationType="Update">

      <Workspace Id="1243581" Name="ILIA test" Phase="Open"
SuppressWelcomeAlert="true"/>

      <Workspace Id="2253599" Name="ILIA test2" Phase="Open"
SuppressWelcomeAlert="true"/>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Close an exchange

Users with the Manager Plus and Hidden Manager Plus roles can close an exchange from ILIA. Exchanges can only be closed on a future date.

- Provide OperationType="Close" and list of exchanges to close in the input file.
- The Workspace “Id” attribute is required.
- Provide DealClose=”true” and “DealClosingDate” (MM/DD/YYYY) in the input file.
- Other optional attributes are:
  - DealComplete=Yes/No/NotApplicable
  - DealClosingRange=Range_0M_250M/Range_250M_500M/ Range_500M_750M/Range_750M_1B/Range_1B_10B/ Range_10B_50B/Range_50B_Plus
  - DealClosingCurrency=USD/EUR/GBP/AUD/CAD/CNY/HKD/INR/JPY/ NZD/PLN/RON/RUB/TWD/BRL/CLP/COP/NOK/DKK/SEK/KPW/KRW

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

        <Identity UserId="ILIAuser@intralinks.com"/>
```

```
        <batch Id="100">

                <Job Id="1" OperationType="Close">

                    <Workspace Id="14428685" DealClose="true"
DealClosingDate="03/14/2023" DealComplete="Yes"
DealClosingRange="Range_0M_250M" DealClosingCurrency="USD"  />

                </Job>

        </batch>

</ExecuteData>
```

Return to Sample XML files index.

Order an archive

Users with a Manager Plus or Hidden Manager Plus role can order archives. Archives can either be downloadable or shipped.When the archive has been ordered, an “OrderId” is added to the output file for tracking archive generation.

- Provide OperationType=”Archive”
- The following attributes are for downloadable archives:
  - Id (Mandatory), user needs to provide Workspace Id
  - ArchiveIncludeAuditInformation (Optional), user can provide true/false. If this attribute is set to true, audit information will be included in the generated archive
  - ArchiveViewpointUser (Mandatory), email address of the user who is part of the exchange that will be used as viewpoint user for generating the archive
  - ArchiveType (Mandatory), should be set to "Downloadable" for ordering downloadable archives
  - ArchiveFormat (Mandatory), can set to either ZIP/VHD
  - ArchiveRecipientEmail (Mandatory), email address of the user who is part of the exchange that will receive notifications for archive generation
  - ArchivePrimaryDownloadLocation (Mandatory), primary country from where archive will be downloaded. Here are the list of acceptable country values - United_States/Afghanistan/Albania/Algeria/ American_Samoa/Andorra/Angola/Anguilla/Antarctica/Antigua/ Argentina/Armenia/Aruba/Ascension_Island/Australia/Austria/ Azerbaijan/Bahamas/Bahrain/Bangladesh/Barbados/Barbuda/Belarus/ Belem/Belgium/Belize/Benin/Bermuda/Bhutan/Bolivia/Bosnia/ Botswana/Bouvet_Island/Brazil/British_Virgin_Islands/ Brunei_Darussalm/Bulgaria/Burkina_Faso/Burundi/Cambodia/ Cameroon/Canada/Cape_Verde/Cayman_Islands/ Central_African_Republic/Chad/Chile/China/CIS/Christmas_Island/ Cocos_Islands/Colombia/Comoros/Congo/Cook_Islands/Costa_Rica/ Croatia/Cuba/Cyprus/Czech_Republic/Denmark/Diego_Garcia/

Djibouti/Dominica/Dominican_Republic/Ecuador/Egypt/El_Salvador/
Equatorial_Guinea/Eritrea/Estonia/Ethiopia/Falkland_Islands/
Faroe_Islands/Fiji/Finland/France/French_Guiana/French_Polynesia/
French_Southern_Territories/Fyrom/Gabon/Gambia/Georgia/
Germany/Ghana/Gibraltar/Grand_Cayman/Greece/Greenland/
Grenada/Guadeloupe/Guam/Guantanamo_Bay/Guatemala/Guernsey/
Guinea/Guinea_Bissau/Guyana/Haiti/
Heard_Island_And_McDonald_Islands/Herzegovina/Honduras/
Hong_Kong/Hungary/Iceland/India/Indonesia/Iran/Iraq/Ireland/
Isle_Of_Man/Israel/Italy/Ivory_Coast/Jamaica/Japan/Jersey/Jordan/
Kazakhstan/Kenya/Khmer_Republic/Kiribati_Republic/Korea/Kuwait/
Kyrgystan/Laos/Latvia/Lebanon/Lesotho/Liberia/Libya/Liechtenstein/
Lithuania/Luxembourg/Macao/Macedonia/Madagascar/Malawi/
Malaysia/Maldives/Mali/Malta/Marshall_Islands/Martinique/Mauritania/
Mauritius/Mayotte/Mexico/Micronesia/Midway_Islands/Moldova/
Monaco/Mongolia/Montenegro/Montserrat/Morocco/Mozambique/
Myanmar/Namibia/Nauru/Nepal/Netherlands/Netherlands_Antilles/
New_Caledonia/New_Zealand/Nicaragua/Niger/Nigeria/Niue/
North_Korea/Norfolk_Island/North_Mariana_Islands/Norway/Oman/
Pakistan/Palau/Palestinian_Territories/Panama/Papua_New_Guinea/
Paraguay/Peru/Philippines/Pitcairn/Poland/Portugal/Puerto_Rico/
Qatar/Reunion_France/Romania/Russia/Rwanda/
Sao_Tome_And_Principe/Saint_Kitts_And_Nevis/Saint_Lucia/
Saint_Martin/Saint_Vincent_And_The_Grenadines/San_Marino/
Saudi_Arabia/Senegal/Serbia/Seychelles/Sierra_Leone/Singapore/
Slovakia/Slovenia/Solomon_Islands/Somalia/South_Africa/
South_Georgia_And_The_South_Sandwich_Islands/Spain/Sri_Lanka/
St_Helena/St_Pierre_And_Miquelon/Sudan/Suriname/
Svalbard_And_Jan_Mayen/Swaziland/Sweden/Switzerland/Syria/
Tahiti/Taiwan/Tajikistan/Tanzania/Thailand/Timor_Leste/Togo/
Tokelau/Tonga/Trinidad_And_Tobago/Tunisia/Turkey/Turkmenistan/
Turks_And_Caicos_Islands/Tuvalu/Uganda/Ukraine/
United_Arab_Emirates/United_Kingdom/
United_States_Minor_Outlying_Islands/Uruguay/Uzbekistan/Vanuatu/
Vatican_City/Venezuela/Vietnam/Virgin_Islands_British/
Virgin_Islands_US/Wallis_And_Futuna/Western_Sahara/
Western_Samoa/Yemen_Arab_Republic/Yugoslavia/Zaire/Zambia/
Zimbabwe

- The following attributes are for shipped archives:
  - Id (Mandatory), user needs to provide Workspace Id
  - ArchiveIncludeAuditInformation (Optional), user can provide true/false. If this attribute is set to true, audit information will be included in the generated archive
  - ArchiveViewpointUser (Mandatory), email address of the user who is part of the exchange that will be used as viewpoint user for generating the archive
  - ArchiveType (Mandatory), should be set to "Shipped_USB" for ordering shipping archives
  - ArchiveFormat (Mandatory), can set to either ZIP/VHD

- ArchiveRecipientEmail (Mandatory), email address of the user who is part of the exchange that will receive notifications for archive generation
- ArchiveShippedUSBQuantity (Mandatory), allowed values are from 1-99
- ArchiveShipping_Company (Mandatory), company name to be added to shipping label
- ArchiveShipping_FirstName (Mandatory), first name to be added to shipping label
- ArchiveShipping_LastName (Mandatory), last name to be added to shipping label
- ArchiveShipping_AddressLine1 (Mandatory), address first line to be added to shipping label
- ArchiveShipping_AddressLine2 (Optional), address second line to be added to shipping label
- ArchiveShipping_City (Mandatory), city to be added to shipping label
- ArchiveShipping_State (Mandatory), state to be added to shipping label
- ArchiveShipping_PostalCode (Mandatory), postal/zip code to be added to shipping label
- ArchiveShipping_Country (Mandatory), country to be added to shipping label (please refer acceptable country values from ArchivePrimaryDownloadLocation above)
- ArchiveShipping_Phone (Mandatory), phone number to be added to shipping label

The following is a sample XML file for a downloadable archive:

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

        <Identity UserId="testUser@intralinks.com"/>

        <batch Id="1000">

              <Job Id="10" OperationType="Archive">

                  <Workspace Id="123456" OrderArchive="true"
ArchiveIncludeAuditInformation="true"
ArchiveViewpointUser="testManager@intralinks.com"
ArchiveType="Downloadable" ArchiveFormat="ZIP"
ArchiveRecipientEmail="testUser@intralinks.com"
ArchivePrimaryDownloadLocation="United_States" />

              </Job>
```

```
        </batch>

</ExecuteData>
```

The following is a sample XML file for shippable archive:

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

        <Identity UserId="testUser@intralinks.com"/>

        <batch Id="1000">

             <Job Id="10" OperationType="Archive">

                <Workspace Id="123456" OrderArchive="true"
ArchiveIncludeAuditInformation="true"
ArchiveViewpointUser="testManager@intralinks.com"
ArchiveType="Shipped_USB" ArchiveFormat="ZIP"
ArchiveRecipientEmail="testUser@intralinks.com"
ArchiveShippedUSBQuantity="1" ArchiveShipping_Company="Test
Company" ArchiveShipping_FirstName="Test"
ArchiveShipping_LastName="Test"
ArchiveShipping_AddressLine1="Test Address1"
ArchiveShipping_AddressLine2="Test Address2"
ArchiveShipping_City="Test City" ArchiveShipping_State="Test
State" ArchiveShipping_PostalCode="11111"
ArchiveShipping_Country="United_States"
ArchiveShipping_Phone="1111122222"/>

             </Job>

        </batch>

</ExecuteData>
```

Return to Sample XML files index.

Working with folders, documents and reports

Create folders

This process will create two folders if “InvestorA CoInvestors VII B” and “LLC\Quarterly Reports” exist on the exchange. If they do not exist, they will be created.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="aUser@intralinks.com" />

  <batch Id="1001">
```

```
    <Job Id="1" OperationType="Create">

      <Workspace Id="761425" Name="Mega Investment Fund
VII,L.P.">

        <Folders>

          <Folder Name="InvestorA CoInvestors VII B
LLC\Quarterly Reports\2009" />

          <Folder Name="InvestorA CoInvestors VII B
LLC\Quarterly Reports\2010" />

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Add folders and documents to an exchange

- This process will add a document to a new folder. To add a document to an existing folder, you do not need to add Create=True, but you need to make sure that the folder name specified is a folder that already exists in the exchange.
- You need to specify a file extension (for example, .doc or .pdf) when adding a document.
- If you are creating a new document and a document already existing within the target folder with the same name, the new document will be added with the same name plus a suffix. (For example you add a document K1.pdf and a file with that name already exists within the folder. The new file will be added with the name K1[2].pdf.
- If long path support has been enabled on your operating system, you can upload documents from paths longer than 260 characters. Contact your system administrator for information on whether long paths are enabled on your system.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="100">
```

```
    <Job Id="2" OperationType="Create">

      <Workspace Id="647645">

        <Folders>

          <Folder Name="My Root Folder Test A\Folder Level 1\My
Sub Folder" Create="true">

            <Documents>

              <Document Name="EIS2.2Document.png"
LocalPath="C:\Documents and Settings\ILIAUser\My
Documents\EIS2.2Document.png" />

              <Document Name="EIS2.2InputSchema.png"
LocalPath="C:\Documents and Settings\ILIAUser\My
Documents\EIS2.2InputSchema.png" />

            </Documents>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Add document with a custom field to an exchange

The labels defined in the CustomFields must match labels in fields definitions in the exchange.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="100">

    <Job Id="2" OperationType="Create">

      <Workspace Id="647645">

        <Folders>
```

```
          <Folder Name="My Root Folder Test A\Folder Level 1\My
Sub Folder" Create="true">

            <Documents>

              <Document Name="EIS2.2Document.png"
LocalPath="C:\Documents and Settings\ILIAUser\My
Documents\EIS2.2Document.png">

                <CustomFields>

                  <CustomField Label="Field1" Value="Test" />

                  <CustomField Label="Field2" Value="12345" />

                </CustomFields>

              </Document>

            </Documents>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Add documents with notes

The “NoteRequired” flag means that the note will be displayed when opening the document form within a browser.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="aUser@YourOrganziation.com" />

  <batch Id="2010-05-21 12:28:23Z">

    <Job Id="2" OperationType="Create">

      <Workspace Id="111111">

        <Folders>
```

```
          <Folder Name="MyTopLevelFolder\RM">

            <Documents>

              <Document Name="anImportantDocument1.pdf"
LocalPath="C:\Users\MyUserName\Documents\Rating Project
Test\Temp files\cfile.pdf" NoteRequired="true">

               <Note>"This is where your note would go."</Note>

              </Document>

              <Document Name="anImportantDocument2.pdf"
LocalPath="C:\Users\MyUserName\Documents\Rating Project
Test\Temp files\afile.pdf" NoteRequired="true">

               <Note>"This is where your note would go."</Note>

              </Document>

            </Documents>

          </Folder>

          <Folder Name="MyTopLevelFolder\ABS">

            <Documents>

              <Document Name="anImportantDocument3.pdf"
LocalPath="C:\Users\MyUserName\Documents\Rating Project
Test\bfile.pdf" NoteRequired="true">

               <Note>"This is where your note would go." </Note>

              </Document>

            </Documents>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Update documents

You can make changes to the content and metadata of documents (for example, rename, sort order, replace attached file, etc.).

To select documents to update, use either the document ID or the full folder path in the exchange. You must specify the file extension (for example, .doc or .pdf) when updating documents.

This example shows how to move a document to a new folder, upload a new document version, and change the name of an existing document.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="MM/dd/yyyy">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Update">

      <Workspace Id="1265631">

        <Folders>

          <Folder Name="Folder Test 200">

            <Documents>

              <Document Name="Doc1.doc" TargetFolder="Folder
Test 300"/>

              <Document Name="Doc2.pdf"
LocalPath="C:\temp\Doc.pdf"/>

             <Document Name="Doc3.xls" NewName="New Name.xls"/>

            </Documents>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Update folder properties

You can update folder properties on a specified exchang. Updatable properties include:

- Folder name
- Folder location within the exchange
- Folder note
- Whether indexing is allowed
- Display order

This input file will create a new folder and set the folder note

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///Z:/ILIA/Schemas/v1.0/
CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1075181" Name="Test Workspace">

        <Folders>

          <Folder Name="Folder 200">

            <Note>"This is a test folder note."</Note>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

This input file will rename an existing folder and move the folder to a new location.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///Z:/ILIA/Schemas/v1.0/
CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Update">

      <Workspace Id="1075181" Name="Test Workspace">

        <Folders>

          <Folder Name="Folder 200" TargetFolder="Top Folder"
NewName="Sub Folder 200"/>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Set ownership of a new document

A tag indicating document ownership is supported for groups, users in group, and users, when adding, removing, downloading, and reporting on documents. This example will set the ownership of new documents.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1265631">

        <Folders>

          <Folder Name="Folder Test 100">

            <Documents>
```

```
            <Document Name="Doc1.doc" OwnerGroup="Test Grp 1"/
>

            <Document Name="Doc2.pdf" OwnerUser="testUser01@e-
trial.com"/>

             <Document Name="Doc3.pdf" OwnerUser="Joe Smith"/>

            <Document Name="Doc4.xls" OwnerGroup="Test Grp 1"
OwnerUser="testUser01@e-trial.com"/>

          </Documents>

        </Folder>

      </Folders>

    </Workspace>

  </Job>

 </batch>

</ExecuteData>
```

Return to Sample XML files index.

Change ownership of an existing document

A tag indicating document ownership is supported for groups, users in group, and users, when adding, removing, downloading, and reporting on documents. This example will change the ownership of existing documents.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

 <Identity UserId="ILIAUser@intralinks.com"/>

 <batch Id="1001">

  <Job Id="1" OperationType="Update">

   <Workspace Id="1265631">

    <Folders>

     <Folder Name="Folder Test 100">

      <Documents>

       <Document Name="Doc1.doc" OwnerGroup="Test Grp 1"/>
```

```
       <Document Name="Doc2.pdf" OwnerUser="testUser01@e-
trial.com"/>

       <Document Name="Doc3.xls" OwnerGroup="Test Grp 1"
OwnerUser="testUser01@e-trial.com"/>

      </Documents>

     </Folder>

    </Folders>

   </Workspace>

  </Job>

 </batch>

</ExecuteData>
```

Return to Sample XML files index.

Set sort order of folders and subfolders

If you want your exchange folders and/or subfolders to be organized in a specific order, you must define this order in the XML Input file. Here is an example of how to do this:

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///Z:/ILIA/Schemas/v1.0/
CSAInputXMLSchema.xsd">

  <Identity UserId="user@yourcompany.com"/>

  <batch Id="6R9379HI">

    <Job Id="adhoc" OperationType="Create">

      <Workspace Id="1397421" Name="2.8 most settings enabled">

        <Folders>

          <Folder Name="TopPerformers\2012" SortOrder="5">

            <Documents>

             <Document Name="First one" LocalPath="C:\Documents
and Settings\lrozin\My Documents\WSB\File Split\Saved Values
Set Entity.doc"/>
```

```
              <Document Name="Second one"
LocalPath="C:\Documents and Settings\lrozin\My
Documents\WSB\File Split\UC_FS_3 Create a New Distribution
.doc"/>

            </Documents>

          </Folder>

          <Folder Name="TopPerformers\2010" SortOrder="2">

            <Documents>

              <Document Name="Best one" LocalPath="C:\Documents
and Settings\lrozin\My Documents\WSB\File Split\Saved Values
Set Entity.doc"/>

              <Document Name="Second best one"
LocalPath="C:\Documents and Settings\lrozin\My
Documents\WSB\File Split\UC_FS_3 Create a New Distribution
.doc"/>

            </Documents>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Update the existing sort order of folders and subfolder

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">

  <Identity UserId="user@intralinks.com" />

  <batch Id="6R9379HI">

    <Job Id="adhoc" OperationType="Update">

      <Workspace Id="1243581" Name="ILIA test">
```

```
        <Folders>

          <Folder Name="Legal\2011 Settlements\APAC"
SortOrder="2" TargetFolder="Legal\2011 Settlements"></Folder>

          <Folder Name="Legal\2011 Settlements\Europe"
SortOrder="1" TargetFolder="Legal\2011 Settlements"></Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Set sort order for documents

This example adds documents and a folder to the exchange and assigns a sort order to them. It also tracks the folder and documents using PassThrough fields (these are returned in the Output XML file).

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="MM/dd/yyyy">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1265631">

        <Folders>

          <Folder Name="Folder Test 200">

            <PassThroughFields>

             <PassThroughField Name="ProjectID" Value="4990" />

            </PassThroughFields>

            <Documents>

              <Document Name="Doc2.pdf"
LocalPath="C:\temp\Doc.pdf" SortOrder="2">
```

```
                <PassThroughFields>

                  <PassThroughField Name="ArtifactID"
Value="44776625" />

                </PassThroughFields>

              </Document>

              <Document Name="Doc8.pdf"
LocalPath="C:\temp\Doc.pdf" SortOrder="8">

                <PassThroughFields>

                  <PassThroughField Name="ArtifactID"
Value="76882922" />

                </PassThroughFields>

              </Document>

            </Documents>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Update sort order for documents

This example shows how to to update the sort order for existing documents.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="MM/dd/yyyy">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Update">

      <Workspace Id="1265631">
```

```
        <Folders>
          <Folder Name="Folder Test 200">
            <Documents>
              <Document Name="Doc2.pdf" SortOrder="2" />
              <Document Name="Doc8.pdf" SortOrder="8" />
            </Documents>
          </Folder>
        </Folders>
      </Workspace>
    </Job>
  </batch>
</ExecuteData>
```

Return to Sample XML files index.

Filter a list of documents using the “Submitted By” or “Submitted On” fields

For the “Submitted By” and “Submitted On” filters to work, you must have Workflow enabled on the Intralinks exchange you are filtering.

Use this filter to report on and download documents that were submitted to an exchange via workflow.

You can provide either the email address or the first and last name of the user in the SubmittedBy tag.

This example shows how to download a list of documents that were submitted by user testuser01@e-trial.com.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

<Identity UserId="ILIAUser@intralinks.com"/>

 <batch Id="1001">

  <Job Id="1" OperationType="Download">
```

```
    <Workspace Id="881865">
     <DocumentCriteria Submitter="testuser01@e-trial.com"/>
    </Workspace>
   </Job>
 </batch>
</ExecuteData>
```

Return to Sample XML files index.

Filter a list of documents using the “Effective Date Range” field

Use this filter to to search for documents that fall within the specified dates. The Effect Date Range filter can be used for downloads,removals, and reporting.

Two attributes, EffectiveDateFrom and EffectiveDateTo will let you specify a range of dates to be matched. The document filtering mechanism in ILIA will use the two fields when comparing the exchange documents. All documents whose EffectiveDate falls within the specified range will be included in the document list.

This input file will download all documents whose EffectiveDate is between Jan 1st, 2011 and April 1st, 2011.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="MM/dd/yyyy" xsi:noNamespaceSchemaLocation="file:///
Z:/ILIA/Schemas/CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Download">

      <Workspace Id="651041">

        <DocumentCriteria EffectiveDateFrom="1/1/2011"
EffectiveDateTo="4/1/2011"/>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Removing folders, subfolders and documents

The chart below provides methods you can use for various removal scenarios. Several XML examples follow.

Removal scenarios

	Condition	Desired result	Input XML and remarks
1	Folder is empty	Remove folder	Remove command with folder name.
2	Folder has a document plus a subfolders.	Remove folder, documents, and subfolders.	Remove command with folder name. All documents and subfolders will be removed.
3	Folder has three documents	Remove documents	Remove command with document names. Documents are removed, folder remains.
4	Folder has three documents	Remove documents and folder	Remove command with folder name. Folder and documents are removed.
5	Folder has two documents	Remove one of the documents	Remove command with folder name and document name. Folder and other documents remain.
6	Folder with empty subfolder	Remove subfolder	Remove command with subfolder name. Subfolder is removed.
7	Folder and subfolder with document.	Remove subfolder and document within it.	Remove command with subfolder name Folder is left, subfolder and document are removed.
8	Folder has document plus subfolder. Subfolder has document.	Remove document in subfolder.	Remove command with subfolder name and document name. Document is removed but subfolder remains.
9	Folder has document plus subfolder. Subfolder has document.	Remove subfolder and its document.	Remove command with subfolder name. Subfolder and its document are removed.
10	Folder has document plus subfolder. Subfolder has two documents	Remove one document from subfolder	Remove command with subfolder name and document name. Document is removed from subfolder. Other document remains in subfolder.
11	Folder is empty	Remove folder	Remove command with folder ID. Folder is removed.
12	Folder has two documents and subfolder	Remove folder, documents and subfolder	Remove command with folder ID. Folder, documents, and subfolders are removed.
13	Folder has one document.	Remove folder and document.	Remove command with folder ID. Folder, and document are removed.
14	Folder has two documents	Remove folder and documens	Remove command with folder ID. Folder, and documents are removed.

	Condition	Desired result	Input XML and remarks
15	Folder has two documents	Remove documents, leave folder	Remove command with folder ID and set ContentsOnly=true. Documents are removed, folder remains.
16	Folder has two documents	Remove one document	Remove command with document ID. Document removed. Other document and folder remain.

Remove documents and folders

This example will remove documents and folders from an exchange.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="1234567">

        <Folders>

          <Folder Name=" Root Folder 1">

            <Documents>

              <Document Name="EIS2.2Document.png" />

            </Documents>

          </Folder>

          <Folder Name="Root Folder 2\Sub Folder">

          </Folder>

          <Folder Name="Root Folder 3" />

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Remove documents and leave folder

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="1265631">

        <DocumentCriteria>

          <FolderList Id="12345678" ContentsOnly="true"/>

        <FolderList Id="11122233" ContentsOnly="true"/>

        </DocumentCriteria>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Remove a list of documents and folders by ID

This example will remove a list of documents and folders by Id. Note that folder "55556660" will not be removed, but its contents (documents and subfolders) will be removed.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="1234567">
```

```
        <DocumentCriteria>
          <FolderList Id="55556660" ContentsOnly="true" />
          <FolderList Id="55556661" />
          <FolderList Id="55556662" />
          <FolderList Id="55556663" />
          <DocumentList Id="1234567890" />
          <DocumentList Id="1234567891" />
          <DocumentList Id="1234567892" />
          <DocumentList Id="1234567893" />
        </DocumentCriteria>
      </Workspace>
    </Job>
  </batch>
</ExecuteData>
```

Return to Sample XML files index.

Remove a list of documents by specified criteria

This example will remove PDF documents created by user jsmith@acme.com between January 1, 2010 and June 30, 2010.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="MM/dd/yyyy">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="1234567">

        <DocumentCriteria CreateDateFrom="01/01/2010"
CreateDateTo="06/30/2010" Creator="John Smith"
MIMEType="application/pdf" />

      </Workspace>
```

```
    </Job>
  </batch>
</ExecuteData>
```

Return to Sample XML files index.

Permissions

ILIA allows you to set permissions for all the documents in one or more folders. You can set or change permissions for all the documents in a folder—or for the documents in multiple folders—at the same time. This is referred to as bulk permissioning.

Intralinks exchanges can be administered to remember or not remember your permissioning settings for documents and folders. You cannot use ILIA to change the exchange setting that governs permissioning; this setting can be changed only by an Intralinks administrator.

- If your target exchange is set to remember permissioning selections for documents and subfolders, your selections will be applied to the documents and subfolders within the folder at the time of permissioning. When you set permissions in this way, documents are permissioned when they are added to the folder.
  - Permissions will be assigned automatically to documents and subfolders added to the folder later.
  - You can override these default permissions for individual documents or subfolders if you like.
  - If you move documents or subfolders from one permissioned folder to another, the permissions are changed automatically. (Any changes you made to the default permissions are remembered when the items are moved.)
  - You can apply permissions to both the folder and document. Both will be applied and document permissions will be marked as overrides to the folder permissions.
  - If permissions are already assigned to any folders/documents within the folder, you have the option to overwrite these permissions with the new ones or to merge the new permissions with the existing ones.
- If your target exchange is not set to remember permissioning selections for documents and subfolders, your selections are applied to whatever documents are located in the folder at the time of permissioning.
  - If you add more documents to a folder after you have set permissions, they will not be assigned the permissions.
  - If you move a folder that has been assigned permissions, it will retain these permissions.

- You can either assign permissions to each document that needs to be uploaded or permission the folder they're in, but not both.

You can use one method of permissioning or the other, but not both at the same time.

Set permissions for a folder

To set permissions for all documents in a folder:

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="100">

    <Job Id="2" OperationType="Create">

      <Workspace Id="647645">

        <Folders>

          <Folder Name="My Root Folder Test A">

            <Permissions>

              <Permission GroupName="EIS Group" Control="true"
Protection="Protect" />

            </Permissions>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Replace permissions for a folder

ILIA provides an option to append or replace existing permissions with newly assigned ones. This example will replace the current permissions on a folder. if the replace attribute is not mentioned, or mentioned as false, the new permissions will be merged with all the existing permissions.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

<Identity UserId="ILIA@intralinks.com"/>

 <batch Id="1001">

  <Job Id="1" OperationType="Update">

   <Workspace Id="1253631">

    <Folders>

     <Folder Name="Folder 01">

      <Permissions Replace="true">

       <Permission GroupName="P_3Ally" Control="true"/>

      </Permissions>

     </Folder>

    </Folders>

   </Workspace>

  </Job>

 </batch>

</ExecuteData>
```

Return to Sample XML files index.

Set permissions for a document

```
<?xml version="1.0" encoding="UTF-8" ?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Identity UserId="ILIAUser@test.com" />

<batch Id="1001">

<Job Id="1" OperationType="Create">

<Workspace Id="278891">
```

```
<Folders>

 <Folder Name="Folder Test">

  <Documents>

   <Document Name="Doc1.doc"
LocalPath="\\TestServer\ILIADocs\Doc1.doc">

    <Permissions>

     <Permission Protection="ProtectNoPrint" See="true"
GroupName="User Team"/>

     <Permission Protection="NoProtect" Control="true"
See="true" GroupName="Admin Team"/>

    </Permissions>

   </Document>

  </Documents>

 </Folder>

</Folders>

</Workspace>

</Job>

</batch>

</ExecuteData>
```

Return to Sample XML files index.

Replace permissions for a document

ILIA provides an option to append or replace existing permissions with newly assigned ones.This example will replace the current permissions assigned to a document.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

<Identity UserId="ILIA @intralinks.com"/>

 <batch Id="1001">
```

```
  <Job Id="1" OperationType="Update">
   <Workspace Id="1253631">
    <Folders>
     <Folder Name="Folder 01">
      <Documents>
       <Document Name="Doc2.pdf" LocalPath="C:\temp\Doc2.pdf">
        <CustomFields>
         <CustomField Label="Document Type" Value="Adverse Drug
Report">
          <CustomField Label="ADE Type" Value="Serious Adverse
Event"/>
         </CustomField>
        </CustomFields>
        <Permissions Replace="true">
         <Permission GroupName="Grp1" Control="true"
Protection="ProtectNoPrint"/>
        </Permissions>
       </Document>
      </Documents>
     </Folder>
    </Folders>
   </Workspace>
  </Job>
 </batch>
</ExecuteData>
```

Return to Sample XML files index.

Permission documents to the All Users group

The All Users Group is a special system group that aggregates all users currently on the exchange. If you permission a document to the All Users group, the permissions will be applied to all users on the exchange in addition to any users

who are added to the exchange later. Note that you can only apply “See Permissions.”

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">

  <Identity UserId="user@yourcompany.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1138881">

        <Folders>

          <Folder Name="Folder Test">

            <Documents>

              <Document Name="Variable length op2"
LocalPath="C:\2.8 Most settings enabled_1397421\Variable length
op2.pdf">

                <Permissions>

                 <Permission See="true" GroupName="All Users -
Current and Future" />

                </Permissions>

                <CustomFields>

                  <CustomField Label="Amount" Value="99" />

                </CustomFields>

              </Document>

            </Documents>

          </Folder>

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Groups and users

- The Global User Directory (GUD) is the master directory of all users and members of Intralinks exchanges.
- The profile information for each person in the GUD is controlled by that person and in most situations, only that person can make changes to their own profile information.
- When adding a person to an Intralinks exchange, you need to define five primary fields – First Name, Last Name, Email Address, Phone, Organization. You should also set the user's role in the exchange he/she is being added to. For new users, other optional profile information can be defined.
  - If a person is new to Intralinks, his/her initial Intralinks account will be created using this information.
  - For people who are already in the GUD, their existing profile information will remain unchanged.
  - The user will be added to the exchange with his/her assigned role. If no role is specified in the XML Input file for the user, a default role will be assigned following the setting in the ILIA Configuration Manager.
- There are a number of optional profile information fields that users can define for themselves.
- A User Difference report is included in the output file when you add people to an exchange.
  - It can be used to see the difference between what users have defined for the profile information fields versus what your internal systems have provided.

Synchronize users in an exchange

The synchronize operation is used to ensure that users on Intralinks exchanges match the users in your organization's back-end system. Documents, groups, reports, and folders are not synchronized.

The synchronization process is unidirectional; from your system of record to Intralinks exchanges. Users are added or removed, or existing user data is updated. User information includes:

- Email address
- First name
- Last name
- Organization
- Phone number

The following table explains the interaction between the synchronization process and ILIA configuration settings.

ILIA Configuration Settings	User Needs to be Added	User Needs to be Removed
Role Exclusion List is defined	User will be added	Protected user will not be removed
Domain Exclusion List is defined	User will be added	Protected user will not be removed
Remove user protection period defined	User will be added	Protected user will not be removed
Create Groups on	User will be added. If the group already exists, a new one will not be created. If the group is missing, it will be created in the exchange.	User will be removed
Create Groups off	User will be added. Group will not be created.	User will be removed
Protection period for addition of users is enabled	Removed user may or may not be added depending on the protection period.	User will be removed
Protection period for addition of users is disabled	Removed user will be added	User will be removed
Remove Empty Groups on	User will be added. Any preexisting empty groups will not be removed.	User will be removed. Empty group that the user belongs to will be removed as a result of user removal. Pre-existing empty groups will not be removed.
Remove Empty Groups off	User will be added. Groups will not be removed.	User will be removed. Empty group that the user belonged to will not be removed as a result of user removal. The synchronization operation will only affect users.

The following table explains the interaction between the synchronization process and synchronization-related XML file content.

ILIA XML Settings	User Needs to be Added	User Needs to be Removed
Only users are mentioned at the exchange level.	User will be added, unless the user was previously removed and the protection period for re-addition of removed users is set to that user.. Empty groups are not removed.	User removed, unless protected. Groups that become empty are not removed, unless the “Remove Empty Groups” is on. Then, the groups that become empty will be removed.
Only groups are mentioned		Groups will be reconciled together with group members and users will be removed from the exchange based on the Configuration Manager setting regardings users who remain without a group.
All the users are mentioned in the context of groups	Users will be added based on the Configuration Manager setting regarding re-addition of previously removed users. Only groups that match the defined criteria will be reconciled.	Users will be removed.
Some users are mentioned in the context of group and some are mentioned at the exchange level.	The operation will be rejected and an error message will be generated.	The operation will be rejected and an error message will be generated.

Synchronizing users in an exchange

This example will add/update three users in the exchange. All other users currently in the exchange will be removed.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Synchronize">

      <Workspace Id="777871" Name="Smoke_B2_WS1">

        <Users>

          <User EMailId="testuser1@e-trial.com" FirstName="Arc"
LastName="User1" Org="Test" Phone="555 123-4567"
Role="Manager+" />
```

```
         <User EMailId="testuser2@e-trial.com" FirstName="Arc"
LastName="User2" Org="Test" Phone="555 123-4321" Role="Manager"
/>

         <User EMailId="testuser3@e-trial.com" FirstName="Arc"
LastName="User3" Org="Test" Phone="555 123-1234"
Role="Reviewer" />

        </Users>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Synchronize groups and users in an exchange

This process will synchronize both groups and users to your system of record. Be aware that the rules specified in the ILIA Configuration Manager under Group Processing Options wil be applied, when applicable, during this process.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="swati_mahadevpur@mindtree.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Synchronize">

      <Workspace Id="996952">

        <Groups>

          <Group Name="Sc7" FTSEnable="false"
DefaultFolder="Folder1">

            <CustomFields>

              <CustomField Label="Allow this Group to run
advanced reports" Value="Yes" />

              <CustomField Label="Group Category" Value="IRB /
EC">

                <CustomField Label="Type of IRB / EC"
Value="Local" />
```

```
                <CustomField Label="Communication Method"
Value="Fax" />

              </CustomField>

            </CustomFields>

            <Note>Synchronizing existing group</Note>

          </Group>

          <Group Name="Sc34_1" FTSEnable="true"
DefaultFolder="Folder4">

            <CustomFields>

              <CustomField Label="Allow this Group to run
advanced reports" Value="Yes" />

              <CustomField Label="Group Category" Value="IRB /
EC">

                <CustomField Label="Type of IRB / EC"
Value="Local" />

                <CustomField Label="Communication Method"
Value="Fax" />

              </CustomField>

            </CustomFields>

            <Note>Synchronizing new group with users</Note>

            <Users>

             <User EMailId="User1@e-trial.com" FirstName="ILIA"
LastName="User1" Org="Test" Phone="555 123-1234" Role="Manager"
/>

             <User EMailId="User2@e-trial.com" FirstName="ILIA"
LastName="User2" Org="Test" Phone="555 123-1234"
Role="Manager+" />

             <User EMailId="User3@e-trial.com" FirstName="ILIA"
LastName="User3" Org="Test" Phone="555 123-1234"
Role="Manager+" />

            </Users>

          </Group>

          <Group Name="Sc34_3" Type="Collaboration"
FTSEnable="true" DefaultFolder="Folder4">
```

```
            <CustomFields>

              <CustomField Label="Allow this Group to run
advanced reports" Value="Yes" />

              <CustomField Label="Group Category" Value="IRB /
EC">

                <CustomField Label="Type of IRB / EC"
Value="Local" />

                <CustomField Label="Communication Method"
Value="Fax" />

              </CustomField>

            </CustomFields>

            <Note>Synchronizing new group with users</Note>

            <Users>

             <User EMailId="User7@e-trial.com" FirstName="ILIA"
LastName="User1" Org="Test" Phone="555 123-1234" Role="Manager"
/>

             <User EMailId="User8@e-trial.com" FirstName="ILIA"
LastName="User2" Org="Test" Phone="555 123-1234"
Role="Manager+" />

             <User EMailId="User9@e-trial.com" FirstName="ILIA"
LastName="User3" Org="Test" Phone="555 123-1234"
Role="Manager+" />

            </Users>

          </Group>

          <Group Name="Sc37_1" FTSEnable="true"
DefaultFolder="Folder4">

            <CustomFields>

              <CustomField Label="Allow this Group to run
advanced reports" Value="Yes" />

              <CustomField Label="Group Category" Value="IRB /
EC">

                <CustomField Label="Type of IRB / EC"
Value="Local" />

                <CustomField Label="Communication Method"
Value="Fax" />
```

```
            </CustomField>

          </CustomFields>

          <Note>Synchronizing new group with users</Note>

          <Users>

           <User EMailId="User1@e-trial.com" FirstName="ILIA"
LastName="User1" Org="Test" Phone="555 123-1234" Role="Manager"
City="Boston" State="MA" Zip="01234" Country="UNITED_STATES"
Mobile="155567777" Title="BOARD_MEMBER"
Industry="FINANCIAL_SERVICES" Address1="1 Main St" Address2="PO
Box 12345" TimeZone="GMT" Fax="1234" />

           <User EMailId="User2@e-trial.com" FirstName="ILIA"
LastName="User2" Org="Test" Phone="555 123-1234"
Role="Manager+" City="Boston" State="MA" Zip="01234"
Country="UNITED_STATES" Mobile="155567777" Title="BOARD_MEMBER"
Industry="FINANCIAL_SERVICES" Address1="1 Main St" Address2="PO
Box 12345" TimeZone="GMT" Fax="1234" />

           <User EMailId="User3@e-trial.com" FirstName="ILIA"
LastName="User3" Org="Test" Phone="555 123-1234"
Role="Manager+" City="Boston" State="MA" Zip="01234"
Country="UNITED_STATES" Mobile="155567777" Title="BOARD_MEMBER"
Industry="FINANCIAL_SERVICES" Address1="1 Main St" Address2="PO
Box 12345" TimeZone="GMT" Fax="1234" />

          </Users>

        </Group>

      </Groups>

    </Workspace>

  </Job>

</batch>

</ExecuteData>
```

Return to Sample XML files index.

Update user roles on an exchange

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />
```

```
  <batch Id="1001">

    <Job Id="1" OperationType="Update">

      <Workspace Id="178621">

        <Users>

          <User EMailId="ILIAUser@intralinks.com"
FirstName="ILIA" LastName="User" Org="Dev" Phone="4343526729"
Role="Manager" City="Boston" State="MA" />

          <User EMailId="tuser3@test.com" FirstName="Test"
LastName="User3" Org="Intralinks" Phone="21311234"
Role="Reviewer" Title="ASSOCIATE" City="Paris" Country="FRANCE"
/>

          <User EMailId="tuser4@test.com" FirstName="Test"
LastName="User4" Org="Dev" Phone="21311235" Role="Reviewer"
City="Boston" />

          <User EMailId="tuser88@test.com" FirstName="Test"
LastName="User88" Org="Test Org" Phone="5554443333"
Role="Reviewer" City="San Juan" Country="UNITED_STATES"
Address1="1 Main St." />

          <User EMailId="tuser89@test.com" FirstName="Test"
LastName="User89" Org="Intralinks" Phone="5555556666"
Role="Previewer" City="London" Country="UNITED_KINGDOM"
Address1="1 Elm St." />

        </Users>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Set member roles of new users

Group member roles are supported for identifying the role of a user in a group (for example, “Senior Auditor”). Group member roles are tags that are useful for managing or reporting on exchange users. Group member roles can be added, changed, or removed within the context of a group.

This example will create three new users and set their group member roles. Note that if users are specified at the exchange level, the member roles will be ignored.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

<Identity UserId="ILIAUser@intralinks.com"/>

<batch Id="1001">

 <Job Id="1" OperationType="Create">

  <Workspace Id="188831" Name="Test Workspace">

   <Groups>

    <Group Name="FL Team">

     <Users>

      <User EMailId="testuser01@e-trial.com" FirstName="Test"
LastName="User_01" Org="BSIL" Phone="7776665501"
Role="Reviewer+">

       <MemberRoles>

        <MemberRole Role="Sub Investigator"/>

       </MemberRoles>

      </User>

      <User EMailId="testuser02@e-trial.com" FirstName="Test"
LastName="User_02" Org="BSIL" Phone="7776665502"
Role="Manager+">

       <MemberRoles>

        <MemberRole Role="Contacts Manager"/>

        <MemberRole Role="Principal Investigator"/>

        <MemberRole Role="Req Docs Manager"/>

        <MemberRole Role="Site Coordinator"/>

        <MemberRole Role="Sub Investigator"/>

       </MemberRoles>

      </User>

      <User EMailId="testuser03@e-trial.com" FirstName="Test"
LastName="User_03" Org="BSIL" Phone="7776665503"
Role="Publisher">
```

```
        <MemberRoles>
         <MemberRole Role="Contacts Manager"/>
         <MemberRole Role="Req Docs Manager"/>
        </MemberRoles>
       </User>
      </Users>
     </Group>
    </Groups>
   </Workspace>
 </Job>
</batch>
</ExecuteData>
```

Return to Sample XML files index.

Set member roles of existing users

Group member roles are supported for identifying the role of a user in a group (for example, “Senior Auditor”). Group member roles are tags that are useful for managing or reporting on exchange users. Group member roles can be added, changed, or removed within the context of a group.

This example will change the member roles of existing users. Note that the second user in the sample will remove all the member user roles.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Update">

      <Workspace Id="188831" Name="Test Workspace">

        <Groups>

          <Group Name="FL Team">

            <Users>
```

```
              <User EMailId="testuser01@e-trial.com"
FirstName="Test" LastName="User_01" Org="BSIL"
Phone="7776665501" Role="Reviewer+">

                <MemberRoles>

                  <MemberRole Role="Principal Investigator"/>

                  <MemberRole Role="Sub Investigator"/>

                </MemberRoles>

              </User>

              <User EMailId="testuser02@e-trial.com"
FirstName="Test" LastName="User_02" Org="BSIL"
Phone="7776665502" Role="Manager+">

                <MemberRoles/>

              </User>

            </Users>

          </Group>

        </Groups>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Add groups and users to an exchange

This file creates two groups and adds a user to both groups.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1234567" Name="Test Workspace">
```

```
        <Groups>

          <Group Name="P_1">

           <Users>

            <User EMailId="test_1@intralinks.com"
FirstName="EIS_test1" LastName="testser_1" Phone="6171111111"
Org="IntraLinks" Role="Publisher" />

           </Users>

          </Group>

          <Group Name="P_2">

           <Users>

            <User EMailId="Test_2@intralinks.com"
FirstName="EIS_test_2" LastName="testuser_2" Phone="6171111111"
Org="IntraLinks" Role="Publisher" />

           </Users>

          </Group>

        </Groups>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Remove a user from an exchange

- Note that the Group element is not required if you are removing a user from an exchange. The user will be removed from *all* groups and the exchange.
- Include the Group element if you wish to remove the user from a specific group, but not from the exchange.
- Removal is based on the user's email address.
- The Workspace Name is not mandatory. ILIA uses the Workspace Id to identify an exchange.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="user@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="144631">

        <Users>

          <User EMailId="user.1243@gmail.com" FirstName="John"
LastName="Smith" Org="IntraLinks" Phone="14354444442"
Role="Publisher" City="" State="" />

        </Users>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Remove all group members from an exchange

If you want to remove all group members from an exchange without specifying the users, you can use the RemoveMembersFromExchange attribute for the targeted group and set it to “true”.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="user@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="1075181">

        <Groups>

          <Group Name="Grp 1" RemoveMembersFromExchange="true"/>

        </Groups>
```

```
      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Remove two users from two different groups

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="100">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="1234567" Name="ARC_Tags_Testing">

        <Groups>

          <Group Name="Group1">

            <Users>

              <User EMailId="joemanager@trial.com"
FirstName="Joe" LastName="Manager" Org="Acme"
Phone="5556667777" Role="Manager_Plus" />

            </Users>

          </Group>

          <Group Name="Group2">

            <Users>

              <User EMailId="joereviewer@trial.com"
FirstName="Joe" LastName="Reviewer" Org="Acme"
Phone="5556667778" Role="Reviewer" />

            </Users>

          </Group>

        </Groups>

      </Workspace>
```

```
    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Sending custom remove alerts for deleted users

You can send a customized note to users who have been removed from an exchange by the removal or synchronization operation. This note is included in the alert email and can be used to specify the reason for removal. Only one note can be created for each exchange.

In order for this feature to work, the exchange in question must be set to send alerts on removal. Send alert on removal can only be set by an Intralinks system administrator.

This example will send a custom remove alert to all users that are removed from an exchange. **Note:** Remove alerts will only be sent if the Remove Alert option is enabled on the exchange.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Synchronize">

      <Workspace Id="777871" Name="Smoke_B2_WS1">

        <Users>

          <User EMailId="testuser1@e-trial.com" FirstName="Arc"
LastName="User1" Org="Test" Phone="555 123-4567"
Role="Manager+" />

          <User EMailId="testuser2@e-trial.com" FirstName="Arc"
LastName="User2" Org="Test" Phone="555 123-4321" Role="Manager"
/>

          <User EMailId="testuser3@e-trial.com" FirstName="Arc"
LastName="User3" Org="Test" Phone="555 123-1234"
Role="Reviewer" />

        </Users>

        <RemoveUserNote>You have been removed by ILIA.</
RemoveUserNote>
```

```
      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Suppress the “Welcome to the Exchange” alert for new users

This example shows how prevent the "Welcome to the Exchange" alert from being sent to new users.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

<Identity UserId="ILIAUser@intralinks.com"/>

 <batch Id="1001">

  <Job Id="1" OperationType="Create">

   <Workspace Id="881865" AddUserProtectPeriod="10">

    <Users>

     <User EMailId="testuser01@e-trial.com" FirstName="Test"
LastName="User_01" Org="BSIL" Phone="7776665501"
Role="Reviewer+" SupressWelcomeAlert="true"/>

    </Users>

   </Workspace>

  </Job>

 </batch>

</ExecuteData>
```

Return to Sample XML files index.

Create multiple groups

This file creates two groups using one Group element and adds the same user to both. The group names are separated using the back slash (‘\’).

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1234567" Name="ARC_Tags_Testing">

        <Groups>

          <Group Name="Group3\Group4">

            <Users>

              <User EMailId="joereviewer@trial.com"
FirstName="Joe" LastName="Reviewer" Org="Acme"
Phone="5556667778" Role="Reviewer" />

            </Users>

          </Group>

        </Groups>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Remove groups from an exchange

ILIA allows you to remove empty and non-empty groups from an exchange.

- To remove groups, specify the group name you wish to remove in the XML input file.
  - ILIA will not attempt to remove groups after a download, create, update, or synchronize operation.
  - If your input file removes all *members* from a group, the group may or may not be removed depending on your setting for the **Remove groups if all members have been removed** option in the Configuration Manager.

This input file will remove a group from an exchange. Users who were members of the group will no longer be associated with the group but will remain on the exchange.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="1075181">

        <Groups>

          <Group Name="Grp 1"/>

        </Groups>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

This input file will remove a group and its members from an exchange.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="1075181">

        <Groups>

          <Group Name="Grp 1" RemoveMembersFromExchange="true"/>

        </Groups>

      </Workspace>
```

```
    </Job>

  </batch>

</ExecuteData>
```

This input file will remove all groups whose name begins with "Test" and groups whose Group Category is "Clinical Research Site".

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Remove">

      <Workspace Id="1075181">

        <GroupUserCriteria>

          <GroupList>

            <GroupFilter GroupName="Test"/>

            <CustomFields>

             <CustomField Label="Name" Value="Clinical Research
Site"/>

            </CustomFields>

          </GroupList>

        </GroupUserCriteria>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Change default folder for groups on create or update

You can set or change the default folder for specified groups when performing a Create or Update operation. On an update, the default folder can be changed or cleared.

This input file will create a new group and set the default folder.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1075181" Name="Test Workspace">

        <Groups>

          <Group Name="New Group" Type="Collaboration"
DefaultFolder="Folder 200"/>

          <Group Name="Test Group" DefaultFolder="Folder 300"/>

        </Groups>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Updating group metadata

The following metadata can be updated for groups on an exchange:

- Name
- Allowing full text search
- Note

This input file will create a new group and set the note and full text search capability.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1075181">

        <Groups>

          <Group Name="Grp 1" FTSEnable="true">

            <Note>This is a group note test.</Note>

          </Group>

        </Groups>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

This input file will rename an existing group.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Update">

      <Workspace Id="1075181">

        <Groups>

          <Group Name="Grp 1" NewName=”Grp 2”/>

        </Groups>

      </Workspace>

    </Job>
```

```
  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Custom fields

- You can create, populate, and publish custom fields for documents, groups and exchanges. Custom fields allow users to describe these elements in greater detail. They can be used to enter specific pieces of information, such as expiration dates, internal tracking numbers, and the like.
- The labels assigned to custom fields are predefined for each exchange. They cannot be changed using ILIA. However, the values associated with these labels can be set using ILIA (see below).
- Once a set of custom fields is defined, some or all of them can be assigned, as needed, to individual exchanges. These assignments can be turned on or off using ILIA (see below).
- Custom fields need to be published to be visible to users on an exchange. ILIA can be used to publish custom fields when creating new exchanges or for existing exchanges.
- Once published, custom field values can be changed using ILIA.

Assign values to exchange custom fields

The labels defined in the custom fields must match labels defined in the field definitions in the exchange.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAuser@intralinks.com" />

  <batch Id="100">

    <Job Id="2" OperationType="Create">

      <Workspace Id="647645">

        <CustomFields>

          <CustomField Label="Field1" Value="Test" />

          <CustomField Label="Field2" Value="12345" />

        </CustomFields>

      </Workspace>
```

```
    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Update values to exchange custom fields

This example shows how to change the value for an existing exchange-level custom field.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">

  <Identity UserId="user@intralinks.com" />

  <batch Id="6R9379HI">

    <Job Id="adhoc" OperationType="Update">

      <Workspace Id="1243581" Name="ILIA test">

        <CustomFields>

          <CustomField Label="Project" Value="Alpha KT-302" />

          <CustomField Label="Status" Value="Analysis" />

        </CustomFields>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Assign values to group custom fields

The labels defined in the custom fields must match labels defined in the field definitions in the exchange.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
 instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId=" ILIAUser @intralinks.com" />

  <batch Id="100">

    <Job Id="2" OperationType="Create">

      <Workspace Id="647645">

        <Groups>

          <Group Name="TestGroup">

            <CustomFields>

              <CustomField Label="Allow this group to run
advanced reports" Value="Yes">

               <CustomField Label="Group Category" Value="IRC /
EC" />

                 <CustomField Label="Type of IRC / EC"
Value="Central" />

                 <CustomField Label="Communication Method"
Value="eMail" />

              </CustomField>

            </CustomFields>

          </Group>

        </Groups>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Change custom fields of an existing document

This example moves a document and changes its effective date and changes the custom fields on a second document.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="MM/dd/yyyy" xsi:noNamespaceSchemaLocation="file:///
C:/IntraLinks/IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

<Identity UserId="ILIAUser@intralinks.com"/>

<batch Id="1001">

 <Job Id="1" OperationType="Update">

  <Workspace Id="1265631">

   <Folders>

    <Folder Name="Folder Test 100">

     <Documents>

      <Document Name="Doc1.doc" TargetFolder="Folder Test 300"
EffectiveDate="11/17/2010"/>

      <Document Name="Doc2.pdf">

       <CustomFields>

        <CustomField Label="Document Type" Value="Adverse Drug
Report">

         <CustomField Label="Adverse Event" Value="Adverse
Event"/>

        </CustomField>

       </CustomFields>

      </Document>

     </Documents>

    </Folder>

   </Folders>

  </Workspace>

 </Job>

</batch>

</ExecuteData>
```

Return to Sample XML files index.

Enable/disable custom fields when creating an exchange

You can enable/disable custom fields when you create a new exchange.The labels defined in the ConfigCustomFields must match labels defined in the field definitions in the exchange.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="100">

    <Job Id="2" OperationType="Create">

      <Workspace Name="TestCreate3" Template="105401" Host="IL
Testing" Description="This is a test exchange" Create="true">

        <ConfigCustomFields>

          <ConfigCustomField Label="Document Type"
Type="Document">

            <ConfigCustomField Label="Additional CCF #1"
Type="Document" Parent="Document Template" Enable="true" />

          </ConfigCustomField>

          <ConfigCustomField Label="Group Category"
Type="Group">

            <ConfigCustomField Label="Country/Region"
Type="Group" Parent="IRB / EC" Enable="false" />

          </ConfigCustomField>

        </ConfigCustomFields>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Update the availability of custom fields at the exchange and group levels and then publish them

This example shows how to update exchange-level and group-level custom fields, some enabled, some disabled, and how to publish the whole set.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">

  <Identity UserId="user@intralinks.com" />

  <batch Id="6R9379HI">

    <Job Id="adhoc" OperationType="Update">

      <Workspace Id="1243581" Name="ILIA test"
PublishFields="true">

        <ConfigCustomFields>

          <ConfigCustomField Label="Project" Type="Exchange"
Enable="true">

            <ConfigCustomField Label="Phase" Type="Exchange"
Enable="false" />

          </ConfigCustomField>

          <ConfigCustomField Label="Location" Type="Group"
Enable="true"></ConfigCustomField>

        </ConfigCustomFields>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Reports

Report User IDs

All the ILIA reports listed below generate a unique identifier for each user that can be entered in the XML Input file to find the user on Intralinks VDRPro when it is not feasible to use an email address.

- For example, if a user is permissioned to a document and then changes his email address, you can enter the user's unique identifier to find the user, without specifying an email address.

Access report

An Access report provides information about access to documents by users and groups, including alert information. Report criteria is specified by the user (for example, "all documents in Folder A").

You can use filter criteria to report only on targeted documents/users/groups.

One Access report will be created for each targeted document. Access reports will be stored inside the destination folder specified in the Input XML file in the following manner:

(Destination folder) / (Exchange ID) (Exchange name) / 'Access Reports' (Date)(Time) / (Document ID) (Document name).XML

Access reports for an exchange

This example will create Access reports for all the documents that exist on the exchange.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Download"
DownloadReportType="DocumentAccess"
DownloadReportPath="C:\Temp\DocumentAccessReports">

      <Workspace Id="881865"/>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Access report using a group filter

This example shows how to create an Access report for an exchange and include only the access data for the specified groups.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Download"
DownloadReportType="DocumentAccess"
DownloadReportPath="C:\Temp\DocumentAccessReports">

      <Workspace Id="881865">

        <GroupUserCriteria>

          <GroupList>

            <GroupFilter GroupName="group 1"/>

            <GroupFilter GroupName ="group 2"/>

            <GroupFilter GroupId="1234567"/>

          </GroupList>

        </GroupUserCriteria>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Access report using a group custom field filter

This example shows how to create an Access report for an exchange and include only the access data for groups with matching custom fields label/value pairs.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">
```

```
    <Job Id="1" OperationType="Download"
DownloadReportType="DocumentAccess"
DownloadReportPath="C:\Temp\DocumentAccessReports">

      <Workspace Id="881865">

        <GroupUserCriteria>

          <GroupList>

            <CustomFields>

              <CustomField Label="Group Category"
Value="Clinical Research Site">

                <CustomField Label="Type of Research Site"
Value="Research Center"/>

              </CustomField>

            </CustomFields>

          </GroupList>

        </GroupUserCriteria>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Access report using a user filter

This example show how to create an Access report for an exchange and include only the access data for the specified users.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">
```

```
    <Job Id="1" OperationType="Download"
DownloadReportType="DocumentAccess"
DownloadReportPath="C:\Temp\DocumentAccessReports">

      <Workspace Id="881865">

        <GroupUserCriteria>

          <UserList>

            <UserFilter UserName="Joe Smith"/>

            <UserFilter UserEMailId="testuser@e-trial.com"/>

            <UserFilter UserEMailId="user001@e-trial.com"/>

          </UserList>

        </GroupUserCriteria>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Access report using a folder name

This example shows how to create an access report for an exchange and include only the access data for documents within a folder specified by the folder name.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">

                  <Job Id="1" OperationType="Download"
            DownloadReportType="DocumentAccess"
            DownloadReportPath="C:\Temp\DocumentAccessReports">

                        <Workspace Id="881865">

                        <Folders>

                        <Folder Name="Reports\2022"/>
```

```
                        </Folders>
                    </Workspace>
                </Job>
            </batch>
    </ExecuteData>
```

Return to Sample XML files index.

Access report using a folder ID

This example shows how to create an access report for an exchange that only includes the access data for documents within the folder specified by the folder ID.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">

                  <Job Id="1" OperationType="Download"
            DownloadReportType="DocumentAccess"
            DownloadReportPath="C:\Temp\DocumentAccessReports">

                        <Workspace Id="881865">

                        <DocumentCriteria>

                                    <FolderList Id="123456789"/
                              >

                              </DocumentCriteria>
                        </Workspace>
                  </Job>
            </batch>
    </ExecuteData>
```

Access report using a document name filter

This example shows how to create an acess report for an exchange that only includes the access data for a document specified by name.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">
```

```
            <Job Id="1" OperationType="Download"
        DownloadReportType="DocumentAccess"
        DownloadReportPath="C:\Temp\DocumentAccessReports">

                    <Workspace Id="881865">

                    <Folders>

                    <Folder Name="Reports\2022">

                          <Documents>

                                        <Document
                                  Name="Doc1.pdf"/>

                                  </Documents>
                              </Folder>
                          </Folders>
                    </Workspace>
              </Job>
        </batch>
</ExecuteData>
```

Return to Sample XML files index.

Access report using a document created date range filter

This example shows how to create an access report for an exchange that only includes the access data for all documents that were created within the specified date range.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">

                  <Job Id="1" OperationType="Download"
            DownloadReportType="DocumentAccess"
            DownloadReportPath="C:\Temp\DocumentAccessReports">

                        <Workspace Id="881865">

                        <DocumentCriteria
                  CreateDateFrom="08.22.2022"
                  CreateDateTo="08.23.2022" />

                        </Workspace>
                  </Job>
            </batch>

</ExecuteData>
```

Return to Sample XML files index.

Access report using a document creator filter

This example shows how to create an access report for an exchange that only includes the access data for all documents created by a specified user.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">

                  <Job Id="1" OperationType="Download"
            DownloadReportType="DocumentAccess"
            DownloadReportPath="C:\Temp\DocumentAccessReports">

                        <Workspace Id="881865">

                        <DocumentCriteria Creator="John Smith"
                  />
                        </Workspace>
                  </Job>
            </batch>
   </ExecuteData>
```

Return to Sample XML files index.

Access report using a document EffectiveDate filter

This example shows how to create an access report for an exchange that only includes the access data for all documents with the specified effective date.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">

                  <Job Id="1" OperationType="Download"
            DownloadReportType="DocumentAccess"
            DownloadReportPath="C:\Temp\DocumentAccessReports">

                        <Workspace Id="881865">

                        <DocumentCriteria EffectiveDate="08/23/
                  2022" />
                        </Workspace>
                  </Job>
            </batch>
   </ExecuteData>
```

Return to Sample XML files index.

Access report using a document effective date range filter

This example shows how to create an access report for an exchange that only includes the access data for all documents to which its effective date falls within the specified range.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">

                  <Job Id="1" OperationType="Download"
            DownloadReportType="DocumentAccess"
            DownloadReportPath="C:\Temp\DocumentAccessReports">

                        <Workspace Id="881865">

                        <DocumentCriteria
                  EffectiveDateFrom="08.11.2022"
                  EffectiveDateTo="08.12.2022" />
                        </Workspace>
                  </Job>
            </batch>
   </ExecuteData>
```

Return to Sample XML files index.

Access report using a document extension filter

This example shows how to create an access report for an exchange that only includes the access data for all documents with a specific extension.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">

                  <Job Id="1" OperationType="Download"
            DownloadReportType="DocumentAccess"
            DownloadReportPath="C:\Temp\DocumentAccessReports">

                        <Workspace Id="881865">

                        <DocumentCriteria Extension="pdf" />
                        </Workspace>
                  </Job>
```

```
        </batch>
</ExecuteData>
```

Return to Sample XML files index.

Access report using a document modified date range filter

This example shows how to create an access report for an exchange that only includes the access data for all documents that were last modified within the specified date range.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">

                  <Job Id="1" OperationType="Download"
            DownloadReportType="DocumentAccess"
            DownloadReportPath="C:\Temp\DocumentAccessReports">

                        <Workspace Id="881865">

                        <DocumentCriteria
                  ModifiedDateFrom="08.23.2022"
                  ModifiedDateTo="08.23.2022" />
                        </Workspace>
                  </Job>
            </batch>
</ExecuteData>
```

Return to Sample XML files index.

Access report using a document custom field filter

This example shows how to create an access report for an exchange that only includes the access data for documents with matching custom field label/vlue pairs.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <Identity UserId="ILIAUser@intralinks.com"/>

            <batch Id="1001">

                  <Job Id="1" OperationType="Download"
            DownloadReportType="DocumentAccess"
            DownloadReportPath="C:\Temp\DocumentAccessReports">

                        <Workspace Id="881865">
```

```
                        <DocumentCriteria>
                                <CustomField Label="Status"
                            Value="Active" />
                        </DocumentCriteria>
                    </Workspace>
                </Job>
            </batch>
    </ExecuteData>
```

Return to Sample XML files index.

Access report using the last alert failure filter

This example shows how to create an access report for an exchange to include documents that have at least one alert failure for any user/group.

Add FailedAlertsOnly attribute to job in input file.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">

        <Identity UserId="ILIAuser@intralinks.com"/>

        <batch Id="1001">

                <Job Id="1" OperationType="Download"
DownloadReportType="DocumentAccess"
DownloadReportPath="C:\ILIA\Downloads\DocumentAccessReports"
FailedAlertsOnly="true">

                        <Workspace Id="881865" />

                </Job>

         </batch>

</ExecuteData>
```

Return to Sample XML files index.

Document-centric report

The Document-centric report is an XML file showing the contents of requested ILP exchanges as well as their hierarchies.

You can use filter criteria to report only on targeted documents.

The following information is returned:

Exchange-level details:

- Exchange Name

- Exchange ID
- Exchange host
- Exchange phase
- Organization that the exchange belongs to
- Exchange Custom Fields

Folder-level details:

- Full folder path
- Folder ID
- Folder index
- Sort order
- ILP folder permissions
  - Groups with See/Control permissions/rights to add documents
  - Implicit and explicit folder permissions per group
  - Implicit and explicit protections per group
  - Implicit and explicit document creation rights per group

Document level details:

- Document name
- Mime type
- Document ID
- Attachment name
- Document index
- Document sort order
- Document note
- Document metadata (size, last accessed, creator, effective date, owner etc.)
- Document custom fields

Document permissions details:

- Permissioned groups or users:
  - Group name
  - Group ID
  - Group type
  - Group permissions
  - Document protection (None, Protect, Protect and Prevent Print)

- Group custom Fields
- User first and last names
- User ID
- User permission
- User attributes (email address, phone number, organization, primary contact indication)

- Non-permissioned groups or users:
  - Group name
  - Group ID
  - Group type
  - Group custom Fields

Document-centric report for an exchange

This example will create a document-centric report for all the documents that exist on the exchange.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xsi:noNamespaceSchemaLocation="file:///C:/
IntraLinks/IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">

<Identity UserId="ILIA-UserId@intralinks.com"/>

   <batch Id="2010-11-15 13:44:14Z">

      <Job Id="1" OperationType="Download" DownloadReportType =
"DocumentCentric" DownloadReportPath="C:\IL-IA
Reports\report.xml" >

         <Workspace Id="831372">

         </Workspace>

      </Job>

   </batch>

</ExecuteData>
```

Document-centric report using filters

Document-centric repors can be generated using the same filters as described above for the access report. The only difference is that `DownloadReportType` and `DownloadReportPath` will be different for each filter.

Return to Sample XML files index.

User-centric report

The User-centric report provides filterable information about:

- Exchanges (names, ID, phase, etc.)
- Groups (names, custom fields, etc.)
- Users (names, address, phone numbers, title, etc.)
- Removed users (names, address, phone numbers, title, etc.)

The report will support the existing filter criteria to filter the report data by users and groups. The existing "DownloadAllExchanges" can be used to create reports for all exchanges the user has access to.

The DownloadReportPath attribute in the Workspace element can be used to set the top-level folder where the User-centric reports will be saved. One report is produced for each exchange that is found.

The report files will be saved in the following structure:

DownloadReportPath\User Centric Reports {datetime}\{Exchange Id Exchange Name}.xml

This input file will create a User-centric report for all users in an exchange.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xsi:noNamespaceSchemaLocation="file:///C:/
IntraLinks/IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">

IntraLinks.com"/>

    <batch Id="1001">

      <Job Id="1" OperationType="Download"
DownloadReportType="UserCentric"
DownloadReportPath="C:\Temp\UserCentricReports">

        <Workspace Id="1075181">

        </Workspace>

     </Job>

    </batch>

</ExecuteData>
```

This input file will create a User-centric report for a specific user in an exchange.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ExecuteData xsi:noNamespaceSchemaLocation="file:///C:/
IntraLinks/IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">

  <Identity UserId="ILIAUser@intralinks.com"/>

    <batch Id="1001">

      <Job Id="1" OperationType="Download"
DownloadReportType="UserCentric"
DownloadReportPath="C:\Temp\UserCentricReports">

        <Workspace Id="1075181">

          <GroupUserCriteria>

            <UserList>

              <UserFilter UserEMailId="testuser@intralinks.com"/
>

            </UserList>

          </GroupUserCriteria>

        </Workspace>

      </Job>

    </batch>

</ExecuteData></ExecuteData>
```

This input file will create a User-centric report for all exchanges the user has access to.

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xsi:noNamespaceSchemaLocation="file:///C:/
IntraLinks/IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Download"
DownloadReportType="UserCentric"
DownloadReportPath="C:\Temp\UserCentricReports"
DownloadAllWorkspaces="true"/>

  </batch>
```

```
</ExecuteData>
```

Return to Sample XML files index.

User Comparison report

The User Comparison report compares user profile information in an Intralinks exchange to user profile information stored on a back-end system. It reports all differences in profile details (name, phone number, etc.) and also indicates whether the user:

- could be found.
- was removed from the exchange.
- was assigned a new email address.
- was on the alias list (users who had more than one profile and these profiles were merged by the system administrator).

When a user is entered for comparison, a pass-through field can be included for each user, for identification purposes.

- Example: John Smith is included in the XML Input file with 50 other individuals, and has an email address of jsmith@corp.com as his main email address on Intralinks, and has also been identified by a client back-end system based on an ID of EmpId787783. This ID can be returned for the user in the actual report next to the his name and email address, as a way to find the user in the report. To do this, include a pass-through field of EmpId787783 in the Input XML file for that user.

Running the User Comparison report

- You must be assigned an InterLinks role of Publisher Plus or above to run this report.
- The XML Input file must contain:
  - Operation type = Download
  - Report type = User Comparison
  - Report folder path (location to store the report)
  - One or more exchange names or IDs
  - A report target folder path
  - At least one email address
- The report can be requested in either an XML format (default), or in an CSV format.
- Only ILP exchanges are supported.
- ILIA will generate one report file per exchange.

XML Format

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="dd/MM/yyyy">

  <Identity UserId="pjohnson@intralinks.com" />

  <batch Id="6R9379HI">

    <Job Id="adhoc" OperationType="Download"
DownloadReportType="UserComparison"
DownloadReportPath="C:\Users\pjohnson\Documents\">

      <Workspace Id="316031">

        <Users>

          <User EMailId="mroberts@yourcompany.com"
FirstName="Mark" LastName="Roberts" Phone="444" Country="CHILE"
/>

          <User EMailId="jjones@yourcompany.com"
FirstName="James" LastName="Jones" Phone="55555"
Country="CHILE" />

        </Users>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

CSV Format

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="dd/MM/yyyy">

  <Identity UserId="lrodgers@yourcompany.com" />

  <batch Id="6R9379HI">

    <Job Id="adhoc" OperationType="Download"
DownloadReportType="UserComparison"
DownloadReportPath="C:\Users\jsmith\Documents\"
DownloadReportFormat="CSV">
```

```
      <Workspace Id="316031">

        <Users>

         <User EMailId="jsmith@yourcompany.com" FirstName="Liv"
LastName="Smith" Phone="444" Country="CHILE" />

          <User EMailId="jjones@yourcompany.com"
FirstName="James" LastName="Jones" Phone="55555"
Country="CHILE" />

        </Users>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Rollback files

Rollback files allow the Intralinks Integration Adapter (ILIA) to remove documents and folders created by a previous batch job. The rollback file is a standard input.xml file without the workspace elements. The OperationType will be set to Rollback and the RollbackFile attribute must specify the name (full path) of the output file created by the previous batch job.

When ILIA processes the rollback file it will search for all successfully added documents and folders for each workspace and remove them from the workspace.

Note that permissions applied to a folder cannot be rolled back.

Rollback input XML file

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Rollback" RollbackFile="c:/temp/
csa/reports/ILIAUser@intralinks.com 1.10.2010 10.06.12 AM.xml"
/>

  </batch>
```

```
</ExecuteData>
```

Return to Sample XML files index.

Download files

- You can download files from a designated exchange or all exchanges.
  - You cannot download protected files.
- You can download files that have been deleted from an exchang.
  - Exchanges can be set up to keep track of all files that have been deleted (a copy of each file will be kept in a “deleted” folder).
- A full download downloads all files, even those that may already exist in the storage location.
- An incremental download will compare the requested download with what already exists in the storage location and download those items that have not already been downloaded.
- A timebound download downloads documents that were added, documents that were modified or both new and modified documents within a specified time frame.
- XML file defines:
  - Source exchange
  - Download filter criteria – the folders and files to be included
  - Optional - Include deleted, include previous versions, include placeholders
- You can select specific files to be downloaded using filters, such as:
  - File name (Title = “Invest*”)
  - Custom Field values such as:

    Label = “Owner” Value = “Joe Smith”

    Creation date range – from + to

    Creator name

    Modified date range (from and to)

    Modified by name

    Document owner
- Files will be downloaded to the location you specified in the Configuration Manager and a report will be generated.
- A folder will be created for each exchange using the exchange name and ID.
- Files will be saved with unique names. Files with the same name will be assigned a unique suffix.

- A folder will be created for each exchange using the exchange name and ID. For example, if the selected destination directory is: *C:\My Documents*, ILIA will store the files in *C:\My Documents\Workspace ABC_123456 w*here “Workspace ABC” is the exchange name, and 123456 is the ID of the exchange.
- Add truncations rules here.

- Files are downloaded as a flat file (no folder hierarchy) but two extra files are created to help reconstruct the original file structure on the exchange.
  - XML file with meta-data and file structure
  - HTML file to view structure locally

Download all files from an exchange

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Download"

      <Workspace Id="1234567" />

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Download all exchange files using a file title filter

This input file will download all files that begin with "Test".

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Download">

      <Workspace Id="131591" DownloadType="Full">

        <DocumentCriteria Name="Test*" />
```

```
      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Download all files from a folder

This input file will download all files from folder "Test Folder".

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Download">

      <Workspace Id="131591" DownloadType="Full">

        <Folders>

          <Folder Name="Test Folder" />

        </Folders>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Download files from a folder using a custom field filter

This input file will download all files from folder that have a custom field named "Status" and Value is "Active".

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
  <Identity UserId="ILIAUser@intralinks.com" />
  <batch Id="1001">
    <Job Id="1" OperationType="Download">
      <Workspace Id="131591" DownloadType="Full">
        <Folders>
          <Folder Name="Test Folder" />
        </Folders>
        <DocumentCriteria>
          <CustomField Label="Status" Value="Active" />
        </DocumentCriteria>
      </Workspace>
    </Job>
  </batch>
</ExecuteData>
```

Return to Sample XML files index.

Download all exchange files using a custom field filter

This input file will download all exchange files that have a custom field named "Exchange Use Type" and a value of "Clinical Study".

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Download">

      <Workspace Id="131591" DownloadType="Full">

        <DocumentCriteria >

         <CustomField Label="Exchange Use Type" Value="Clinical
Study" />

        </DocumentCriteria>
```

```
      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Download all files from all exchanges

This input file will download all files in all exchanges that the users has permissions to access.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Download"
DownloadAllWorkspaces="true" />

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Download files added to an exchange within a specified time range

This input file will download files that were added to the specified exchange during the specified time range.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat= "dd/MM/yyyy" >

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Download" DownloadType
="TimeBound" DocumentType ="Create" FromDate="08/31/2015"
ToDate ="08/31/2015" >
```

```
      <Workspace Id="4566301" />

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Download files that were modified in an exchange within a specified time range

This input file will download files that were modified in the specified exchange during the specified time range.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat= "dd/MM/yyyy" >

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Download" DownloadType
="TimeBound" DocumentType ="Update" FromDate="08/31/2015"
ToDate ="08/31/2015" >

      <Workspace Id="4566301" />

    </Job>

</batch>

</ExecuteData>
```

Return to Sample XML files index.

Download both files that were added to and modified in an exchange within a specified time range

This input file will download both files that were added to the specified exchange and files that were modified in the exchange during the specified time range.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat= "dd/MM/yyyy" >

  <Identity UserId="ILIAUser@intralinks.com" />
```

```
  <batch Id="1001">

    <Job Id="1" OperationType="Download" DownloadType
="TimeBound" DocumentType ="Both" FromDate="08/31/2015" ToDate
="08/31/2015" >

      <Workspace Id="4566301" />

    </Job>

</batch>

</ExecuteData>
```

Return to Sample XML files index.

Miscellaneous

Use PassThrough fields

When performing an action using the Input XML file, optional attributes can be included for each:

- user
- group
- exchange
- folder
- document

These optional attributes are passed through without processing to the Output XML files for reporting purposes and they can be used for tracking purposes. An external system that is reading the Output files will be able to use the optional fields to identify any of those users, groups, exchanges, folders, or documents that were marked with PassThrough fields in the XML Input file.

PassThrough fields cannot be defined as part of a report generation request.

This example shows how to set pass-through fields for exchanges and users in the XML Input File.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="881865">
```

```
        <PassThroughFields>

          <PassThroughField Name="dealTrackingNumber"
Value="1234567890"/>

          <PassThroughField Name="dealRID" Value="555"/>

          <PassThroughField Name="dealName" Value="Test Name"/>

        </PassThroughFields>

        <Users>

          <User EMailId="testuser01@e-trial.com"
FirstName="Test" LastName="User_01" Org="BSIL"
Phone="7776665501" Role="Reviewer+">

            <PassThroughFields>

             <PassThroughField Name="contactRid" Value="12345"/
>

            </PassThroughFields>

          </User>

        </Users>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

This example shows how to set PassThrough fields for users and groups in the XML Input File.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">

  <Identity UserId="user@intralinks.com" />

  <batch Id="6R9379HI">

    <Job Id="adhoc" OperationType="Create">
```

```
      <Workspace Id="265591" Name="Test exchange123">

        <PassThroughFields>

          <PassThroughField Name="dealTrackingNumber"
Value="#48Y9F1O" />

          <PassThroughField Name="dealRID" Value="#48Y9F1O" />

          <PassThroughField Name="dealName" Value="WHITE BIRCH
PAPER COMPANY DIP" />

          <PassThroughField Name="dealCUSIP" Value="" />

        </PassThroughFields>

        <Groups>

          <Group Name="AlphaGroup">

            <PassThroughFields>

             <PassThroughField Name="SiteID" Value="78887277" /
>

            </PassThroughFields>

            <Users>

              <User EMailId="ajones@company.com"
Role="Reviewer">

                <PassThroughFields>

                  <PassThroughField Name="contactRID"
Value="4B8ZLBJM" />

                </PassThroughFields>

              </User>

              <User EMailId="Lmatalon@company.com">

                <PassThroughFields>

                  <PassThroughField Name="contactRID"
Value="6C8ZLBJP" />

                </PassThroughFields>

              </User>

              <User EMailId="NewMember@company2.com">

                <PassThroughFields>
```

```
                    <PassThroughField Name="contactRID"
Value="6B8ZLBJK" />

                  </PassThroughFields>

                </User>

                <User EMailId="Jsmith@company2.com"
FirstName="John" LastName="Smith" Org="JMP" Phone="555-123-
1234" Role="Reviewer">

                  <PassThroughFields>

                    <PassThroughField Name="contactRID"
Value="7T8ZLBJK" />

                  </PassThroughFields>

                </User>

               <User EMailId="user1@email.com" FirstName="user1"
LastName="last1" Org="CS" Phone="212-123-1234" Role="Manager">

                  <PassThroughFields>

                    <PassThroughField Name="contactRID"
Value="8B8Z33JY" />

                  </PassThroughFields>

                </User>

              </Users>

            </Group>

          </Groups>

        </Workspace>

      </Job>

    </batch>

</ExecuteData>
```

Return to Sample XML files index.

This example shows how to set PassThrough fields for folders and documents in the XML Input File.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
DateFormat="MM/dd/yyyy">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1265631">

        <Folders>

          <Folder Name="Folder Test 200">

            <PassThroughFields>

             <PassThroughField Name="ProjectID" Value="4990" />

            </PassThroughFields>

            <Documents>

              <Document Name="Doc2.pdf"
LocalPath="C:\temp\Doc.pdf" SortOrder="2">

                <PassThroughFields>

                  <PassThroughField Name="ArtifactID"
Value="44776625" />

                </PassThroughFields>

              </Document>

              <Document Name="Doc8.pdf"
LocalPath="C:\temp\Doc.pdf" SortOrder="8">

                <PassThroughFields>

                  <PassThroughField Name="ArtifactID"
Value="76882922" />

                </PassThroughFields>

              </Document>

            </Documents>

          </Folder>

        </Folders>
```

```
      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Set the user protection period in the Input XML file

ILIA provides an option to specify a protection period during which users who have been removed from an exchange will not be re-added. This will prevent a client's internal system of record from automatically replacing users who have been intentionally removed using the web user interface. This value can also be set in the Configuration Manager.

This example shows how to set the protection period in effect when adding new users to an exchange. The value specified in the Input XML file overrides the value entered in the Configuration Manager. This example will set the period to 10 days for this input file.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="881865" AddUserProtectPeriod="10">

        <Users>

          <User EMailId="testuser01@e-trial.com"
FirstName="Test" LastName="User_01" Org="BSIL"
Phone="7776665501" Role="Reviewer+"/>

        </Users>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Override email notification

Email notification is set using the ILIA Configuration Manager but you can override the recipient(s) of the email for a given job using the procedure below. You must use the Configuration Manager to change the reason(s) for notification (on error, on completion, etc.).

```
<?xml version="1.0" encoding="UTF-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
EMailNotification="ILIAUser1@intralinks.com,ILIAUser2@intralink
s.com,ILIAUser3@intralinks.com"
xsi:noNamespaceSchemaLocation="file:///C:/IntraLinks/
IntraLinks%20Integration%20Adapter%20Service/
CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser1@intralinks.com"/>

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="278891">

        <Users>

          <User EMailId="ILIA_TestUser_005@e-trial.com"
FirstName="ILIA" LastName="TestUser_005" Phone="6171181115"
Org="IntraLinks"/>

          <User EMailId="ILIA_TestUser_006@e-trial.com"
FirstName="ILIA" LastName="TestUser_006" Phone="6171181116"
Org="IntraLinks"/>

        </Users>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Change the name of the Output XML file

You can use the Input XML file to change the name of the Output XML file, instead of using the ILIA Configuration Manager. This is helpful if you have more

than one person submitting Input files. A setting here will override the setting in the Configuration Manager.

You can define the following naming conventions in the Configuration Manager:

Valid Output File Naming

The output filename is defined using any of the following options:

- ILIA User ID + date/time stamp
- Input filename + batch ID + date/time stamp
- Customizable name + date/time stamp

Invalid Output File Naming

When a serious error occurs and the Input file cannot be processed, there are two options for naming the output file:

- Input filename + date/time stamp
- Customizable name + date/time stamp

Output XML File location

The XML Output file is stored in the Output Directory you specified in the Configuration Manager. By default, the adapter will use the 'Output' folder under the installation folder.

The file below will cause the name of the Output XML file to start with “myOutputFileName”. The rules for changing the Output file name using the Input XML file are:

- ILIA will accept an alpha-numeric string as the requested Output file name.
- ILIA will use this string at the beginning of the Output file name and will append to it the Batch ID from the Input file and a date/time stamp.
  - The new name will override the chosen Output file name option from the Configuration settings.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
OutputFilename="myOutputFileName"
xsi:noNamespaceSchemaLocation="file:///Z:/ILIA/Schemas/v1.0/
CSAInputXMLSchema.xsd">

  <Identity UserId="ILIAUser@intralinks.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="16589" Name="Smoke_B2_WS1">
```

```
        <Users>

         <User EMailId="testuser1@e-trial.com" FirstName="Arc"
LastName="User1" Org="Test" Phone="555 123-4567"
Role="Manager+" />

         <User EMailId="testuser2@e-trial.com" FirstName="Arc"
LastName="User2" Org="Test" Phone="555 123-4321" Role="Manager"
/>

         <User EMailId="testuser3@e-trial.com" FirstName="Arc"
LastName="User3" Org="Test" Phone="555 123-1234"
Role="Reviewer" />

        </Users>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Question and Answer functionality

Some Intralinks exchanges support Question and Answer (Q&A) functionality. If the Q&A setting is marked (on), specified users will be able to ask questions and coordinators in your organization will be able to answer the questions or delegate them to subject matter experts.

Add user as a question submitter

If Q&A is enabled on an exchange, the following procedure enables a user to submit questions.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="user@yourcompany.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Create">

      <Workspace Id="1053545"  >

        <Groups>

          <Group Name="BuyerGroup1" Type="Buyer">
```

```
            <Users>

              <User EMailId="testuser_1@e-trial.com"
FirstName="User1" LastName="EIS-218" Org="EIS218_Org"
Phone="996662299" QuestionSubmitter="true" Role="Reviewer"  />

              <User EMailId="testuser_2@e-trial.com"
FirstName="User1" LastName="EIS-218" Org="EIS218_Org"
Phone="996662299" QuestionSubmitter="true"  Role="Previewer"  /
>

            </Users>

          </Group>

        </Groups>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.

Revoke a user's right as a question submitter

The following procedure will revoke a user's right to submit questions.

```
<?xml version="1.0" encoding="utf-8"?>

<ExecuteData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Identity UserId="user@yourcompany.com" />

  <batch Id="1001">

    <Job Id="1" OperationType="Update">

      <Workspace Id="1053545"  >

        <Groups>

          <Group Name="BuyerGroup3" Type="Buyer">

            <Users>

              <User EMailId="testuser_11@e-trial.com"
FirstName="User1" LastName="EIS-218" Org="EIS218_Org"
Phone="996662299" QuestionSubmitter="false" Role="Reviewer"  />
```

```
              <User EMailId="testuser_21@e-trial.com"
FirstName="User1" LastName="EIS-218" Org="EIS218_Org"
Phone="996662299" QuestionSubmitter="false"  Role="Previewer"
/>

            </Users>

          </Group>

        </Groups>

      </Workspace>

    </Job>

  </batch>

</ExecuteData>
```

Return to Sample XML files index.