

# Join Account and Multiple Activations

---

This document contains information to answer some commonly asked questions about Join Account and Multiple Activations. It is still a work-in-progress paper, your contribution, feedback and questions are greatly appreciated.

**Note:** When we refer to Adobe ID, note that Vendor ID account has the same behavior. Unless these two types of accounts are clearly different under a circumstance, the account referred to as “Adobe ID” account also applies to a Vendor ID account.

The use of the word “device” also includes applications, unless differences are clearly stated when necessary.

If your device or application supports Vendor ID activation, you are required to support join account for passing Adobe’s certification. The latest certification forms are labeled Version 3.

# FAQ on Join Accounts (Account Synchronization)

---

**Q:** What is the main purpose of joining Adobe ID accounts?

To allow devices activated with one Adobe ID (or Vendor ID) to be able to read books previously fulfilled with another Adobe ID (or Vendor ID). The goal is to provide a convenient method of synchronizing multiple reader accounts for the same user, particularly if the user has purchased books from multiple Vendors who support Vendor ID accounts.

**Q:** How many accounts can be joined in one group?

There is no known limitation on the number of accounts that can be joined as a group.

**Q:** Can an account be joined with multiple “groups”?

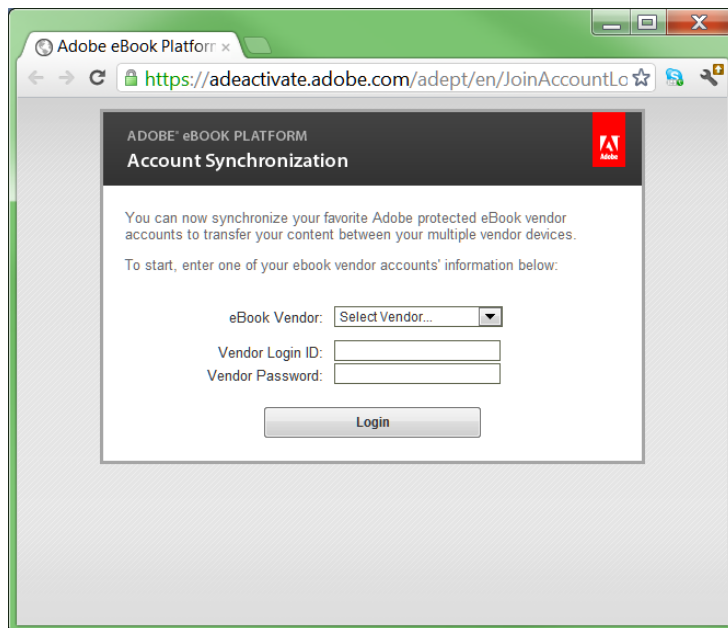
If an account is joined with another account, that account cannot be joined with a different group of accounts. For instance, let’s say we have 4 accounts, A, B, C and D. If A and B are joined, C and D are joined, you can’t later join A and C together.

**Q:** Can I “unjoin” accounts?

Currently you cannot “unjoin” accounts once they are joined.

**Q:** How do I find out which accounts are joined together?

You can go to this page: <https://adeactivate.adobe.com/adept/en/JoinAccountLoginForm.jsp> to see which accounts are joined together. See a sample screenshot:



**Q:** How many devices can be activated with joined accounts?

The number of devices is limited to 6 for each Adobe ID account. After two or more accounts are joined, the group as a whole will have the limit of 6, not multiples of 6 devices.

To be more precise, the limits are: 6 standalone devices and 6 mobile and tethered devices (mobile and tethered devices share the same pool).

See FAQ on Activations from our knowledgebase:

[http://kb.datalogics.com/articles/FAQ/FAQ-on-Activations-in-Adobe-RMSDK-and-Adobe-Digital-Editions?retURL=%2Fapex%2FknowledgeProduct%3Fc%3DAdobe\\_Content\\_Server%26k%3D%26lang%3Den\\_US&popup=false&lang=en\\_US](http://kb.datalogics.com/articles/FAQ/FAQ-on-Activations-in-Adobe-RMSDK-and-Adobe-Digital-Editions?retURL=%2Fapex%2FknowledgeProduct%3Fc%3DAdobe_Content_Server%26k%3D%26lang%3Den_US&popup=false&lang=en_US)

**Q: What is the active account when a device is activated with multiple accounts vs. when a device is activated with joint accounts?**

See the answer to question about the active account under multiple activations. For joint account, you are still logged in with one account which is the only login account you have used (say user A). If you tried to open a book from another account (say user B). If the A and B are joined already, the book B should open just like if you are opening a book fulfilled under user A. If A and B are not joined yet, the device will prompt the user with a web page to join two accounts (only a device or app that has internet connection can initiate joining process). After the user finishes the joining step, goes back to the reader app, the book from user B should be allowed to open at this point.

**Q: Why is the advantage of supporting join account and why is it required?**

Joint account allows a more seamless integration of a reader's multiple accounts. There may be many reasons that a reader has multiple accounts, either from Adobe or from multiple vendors. By giving the reader the ability to read these books on one device and by allowing readers to join these accounts together overall, helps prompting a more interoperable reading environment.

Join account is designed to provide such flexibility. Although some vendors may prefer to have their ebooks delivery system closed, i.e., providing the use of their Vendor ID account solely on a reader application. Although such use is allowed and Adobe provides the Vendor ID services to facilitate such use cases, Adobe still strongly believe in providing the user the option to work in more open system.

**Q: What are the differences between Join Accounts (JA) and Multiple Activations (MA)?**

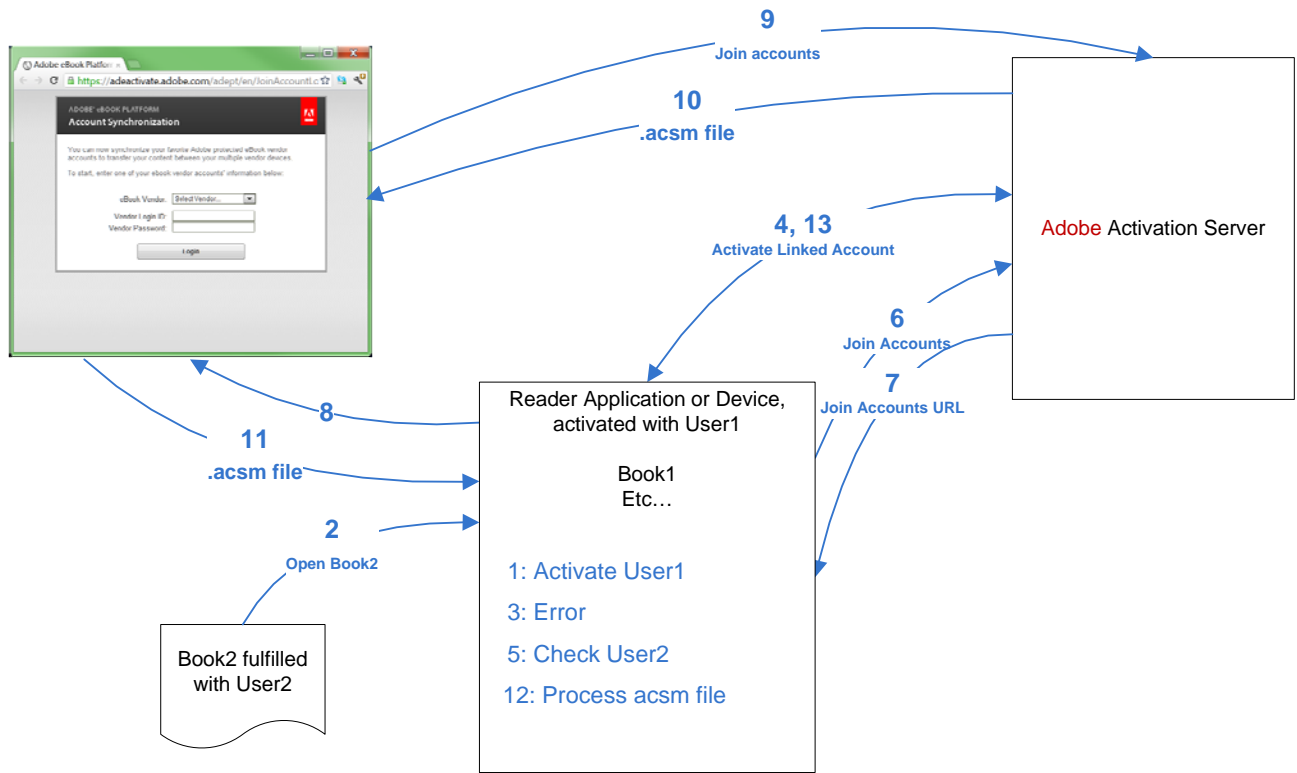
More details about Multiple Activations is contains in the 2<sup>nd</sup> half of this document.

There are a couple of main differences between JA and MA. Join Account allows a reader to do a single sign-on, i.e., the reader only needs to enter one credential, the reader application can fetch the joined accounts which subsequently enables the reader to read books fulfilled with any of these accounts.

The other difference between JA and MA, is that once accounts are joined, the reader can sign on to a different device/application with a single account, without having to re-enter different account credential multiple times.

If your device and application supports Vendor ID, for the above reasons, it is required to support Join Account, but support of multiple activations is optional.

# Join Accounts Workflow

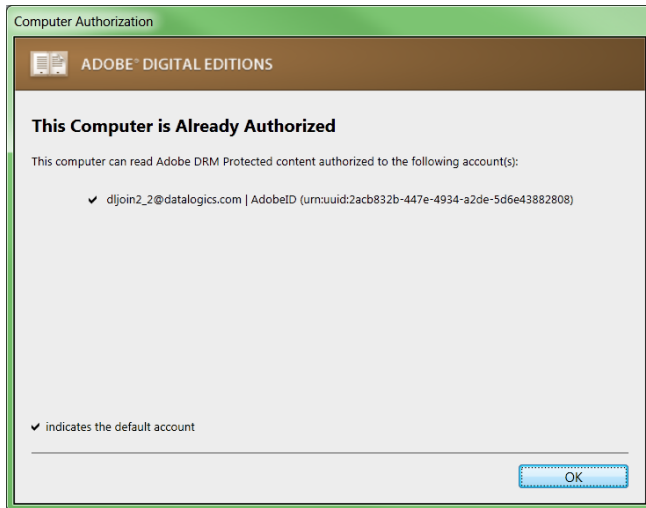


1. Activate reader with account User1 (Adobe ID or Vendor ID account), download and read books.
2. Side load Book2 that was previously fulfilled with Account User2 (Adobe ID or Vendor ID from the same or a different Vendor).
3. This will result in E\_ADEPT\_CORE\_USER\_NOT\_ACTIVATED error which should be trapped by the reader application. The ID for User2 is stored with this message.
4. The application starts an Activate Linked Accounts workflow.
5. The application checks if the User2 is activated after the Linked Accounts activation.
6. If the User2 is not activated still, the application starts a Join Accounts workflow.
7. Server returns a URL for initiating the Join Accounts from the user.
8. The application starts the browser with the given URL.
9. User enters data for User2 and User1 and User2 accounts are joined.
10. Server responds with an acsm format token, either as success or failure.
11. Browser brought the user back to the reader application by loading the .acsm file.
12. The reader application processes the acsm file, determine if the join accounts workflow completed successfully.
13. The application starts an Activate Linked Accounts workflow again, Step 4.
14. Now the User2 is activated and the reader can proceed opening Book2.
15. If the reader application always requests the linked accounts in the future activation, both Book1 and Book2 will be able to open since now the reader application is activated with the joined accounts.

# Join Accounts Workflow in Adobe Digital Editions Preview

Currently released ADE 1.7 does not support join accounts. ADE Preview (Version 1.8) can be downloaded from Adobe Labs (labs.adobe.com). This new version of ADE can recognize credentials from joined accounts, but does not support the workflow that initiates the join account (as of 2/24/2012) so you cannot join accounts with ADE.

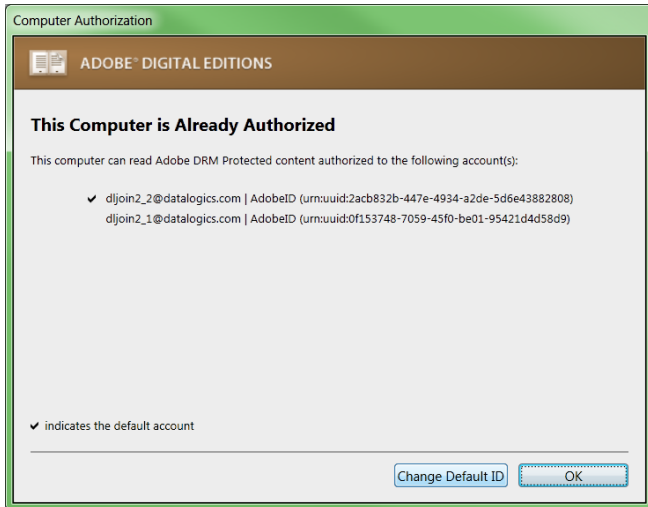
Step 1. Open ADE Preview, activate with one user account, it shows only one account:



Step 2. Since accounts are joined already, at this point, if you try to open a book fulfilled with the 2<sup>nd</sup> account, ADE will automatically authorize with this second account to open the new book.

If you have books from the 2<sup>nd</sup> user already added in ADE Preview, it will also open automatically when you perform step 1.

Step 3. If you open the dialog "Authorization Information" in ADE Preview, you will see both accounts listed, with the 1<sup>st</sup> one as the default.



# Notes for developing join account workflow

---

## Join Accounts Implementation

Join accounts implementation is covered in section 8.7.3 in the RMSDK User manual, and additional API information in html pages of RMSDK API Specification.

The User Manual lists three parameters for `initJoinAccountsWorkflow` which does not quite match the API listed in the specification. The names of the parameters in the specification match more closely to the meaning of the parameters. See more detailed explanation from the specification.

```
virtual unsigned int dpdrm::DRMProcessor::initJoinAccountsWorkflow
( const dp::String & user,
  const dp::String & operatorURL,
  const dp::String & title );
```

Initializes account join workflow.

This workflow is typically initiated when an imported ("side-loaded") resource failed to open because it is licensed to a different user. Error message (`E_ADEPT_CORE_USER_NOT_ACTIVATED`) will contain user id to which the resource belongs and `operatorURL` of the operator who issued the license. Account join workflow purpose is to initiate contact to the authentication provider of that new userid and establish a link between the current DRM user id and this new user id ("join accounts"). Parameter `user` defines DRM account which should be joined to the current active account. Parameter `operatorURL`, if non-null value is passed, is taken from the license of the resource and `title`, if not null, is the title of the resource.

Note that the parameter `title` comes from the `dc:title` metadata field from the book. It is not clear whether this serves any real purpose or not since the key to identify a book is usually by its resource ID.

### 8.7.3 JOINING ACCOUNTS

You can also get the URL that allows a user to join a new account to the current account:

```
unsigned int dpdrm::DRMProcessor::initJoinAccountsWorkflow(
unsigned int workflows,
const dp::String & userID,
const dp::String & newUserID )
```

This contacts the activation server and authenticates with the credentials for the `userID`, which must be present on the device for the `DRMProcessor`, and negotiate a session and resulting URL on the activation server. Next, the `launchBrowser` callback for the `DRMProcessorClient` is called with the negotiated URL, and a web browser should be launched going to the URL by the RMSDK implementer. The user can enter the required credentials to authenticate for the `newUserID`, or choose to cancel. The activation server generates a `.acsm` file containing either a success or failure message which, when opened, will be reported to the `DRMProcessorClient` using either `workflowsDone` or `reportWorkflowError` as appropriate.

Even if successful, this workflow will not activate the device to the account indicated by `newUserID`. Instead it is the responsibility of the RMSDK implementer to start an `Activate Linked Accounts` workflow upon successful completion of the `Join Accounts` workflow.

After calling this API, you will need to trigger the call to start the workflow. Here is a sample request:

```
<?xml version="1.0"?><adept:joinRequest xmlns:adept="http://ns.adobe.com/adept">
<adept:user>urn:uuid:bd53b885-8d84-4582-a586-b5d27cf6194f</adept:user>
```

```
<adept:newUser>urn:uuid:0f153748-7059-45f0-be01-95421d4d58d9</adept:newUser>
<adept:operatorURL>http://acswin.datalogics.com:8090/fulfillment</adept:operatorURL>
<adept:title>dljoin2_2@datalogics.com</adept:title>
<adept:clientOS>IPHONE 3.x.x</adept:clientOS>
<adept:clientLocale>en</adept:clientLocale>
<adept:clientVersion>DLReader 1.6.1</adept:clientVersion>
<adept:encryptedSessionID>Inq/xQtvUXDkhmPaCaJUpwME49Wd8mi9+1M8nSQG+1z3eoPcQep4IyBJ9dhz
gK7xxqLdFIT3ta2yjkpB2/t7o9UfImQ557ETcbH62F4gFyc97xm1/uryQyd3WHcY+KQaiohMi9yFLqcCOBWQAr
pOo+afrqrHQdP8LmuWSPCRxI8=</adept:encryptedSessionID>
<adept:expiration>2012-03-09T19:43:54Z</adept:expiration>
<adept:signature>Ysqqib4JHR6ChOMGGYe8Bz0WehlU2pbyP9Kk9F/j3zQd/m7g9SHvgjBBmLzfnKxdCBxx8
IcQbSvuaxl0Mx6BBH1cX/Sk/PdJ3fRaZMobMmt5baK+PHvXQTNEvJzCPoP/syVXVdg8Kd3zZ3YC3oeduu2lEgR
TrR6F5pcoKoJwytE=</adept:signature></adept:joinRequest>
```

A sample response:

```
<joinResponse xmlns="http://ns.adobe.com/adept">
  <browse>https://adeactivate.adobe.com/adept/en/JoinAccountsForm.jsp?sid=89d201a6d93afecdf7f2</browse>
</joinResponse>
```

A few words on join account web form

The returned browser link for joining accounts is a temporary link. The link expires in a short time and is unique for each response. You may see an error stating that “Account synchronization session expired” during debugging if you wait too long to consume the link.

#### 8.7.4 OPENING A BOOK

To take advantage of the Activating Linked Accounts and Joining Accounts workflows, the RMSDK listen for an `E_ADEPT_CORE_USER_NOT_ACTIVATED` error, which is reported to the `DPDocumentClient::reportDocumentError` callback. For this error, the second token (space separated) is the `userID` for the license in the book. To respond to this error, the RMSDK implementer should first check whether a network connection is available. If a network connection is not available, the error should be reported to the user. If a network connection is available, an ‘Activate Linked Accounts’ workflow should be started to see whether the user has already joined accounts. After successful completion, `DRMProcessor::getActivations` can be used to determine whether the needed `userID` has been activated for the device. If the `userID` has not been activated for the device, the implementer should, if the device has a browser that is available, start a ‘Join Accounts’ workflow and, after successful completion, then immediately start an ‘Activate Linked Accounts’ workflow.

If the `DRMProcessor::initWorkflow` for the application does include `DW_GET_CREDENTIAL_LIST` then you can read books from the Joined IDs. `DW_GET_CREDENTIALS_LIST` populates all the records associated with signed in user. It is safe to use for non-joined accounts. In other word, you can keep this flag on at all times. What this does is go out and get the accounts that are joined to the account you are activating with and recreate the credentials for those accounts in the activation.xml.



DW\_GET\_CREDENTIAL\_LIST - This workflow is initiated by by calling DRMProcessor::initWorkflow method. It may be combined with DW\_ACTIVATE flag.

## Join accounts test with book2png

Content of JoinAccounts.txt:

```
createDRMProcessor "primary" 0
DRMProcessor_initJoinAccountsWorkflow I1 "" "http://foo" "title"
DRMProcessor_startWorkflows I1
process 0
```

```
cmd> book2png -command-file JoinAccounts.txt
```

When you run JoinAccounts.txt with book2png you will be given a URL, then you copy and paste the URL to a browser, you will see the web form in the browser.

Content of getCredentialList.txt:

```
createDRMProcessor "primary" 0
createDRMWorkflowFlags I0 "DW_GET_CREDENTIAL_LIST DW_ACTIVATE"
DRMProcessor_initWorkflows I1 I0 ""
DRMProcessor_startWorkflows I0
process 0
```

```
cmd> book2png -command-file getCredentialList.txt
```

getCredentialList.txt is used to get the activation records for all the joined accounts to the device.

# FAQ on Multiple Activations

---

Q: Does multiple activations mean that one can activate a RMDSK reader app with multiple accounts?

Yes.

Q: If a device is activated with multiple accounts, a book already fulfilled by any of those accounts can be read?

Yes.

Q: If a device is activated with multiple accounts, under which account will a new book fulfilled? The first or the last account used to activate the device or the reader can choose?

DRMProcessor has setUser() method, which should be called in order to use a particular user account for DRM workflows. If this is not called (by default it is not), RMSDK uses the default user, which should match the first valid activation.

Q: If a device supports multiple activations, why would anyone want to use joint account?

With joint account you only need to sign in once (single sign-on). Another benefit is for older accounts, say you have a Vendor ID created with a bookstore that is no longer in business, if the two accounts are joined, you will be able to read the books purchased from the now defunct bookstore with your currently activate Adobe ID account or a different Vendor ID account.

Q: From a vendor's point of view, such as a bookstore, who wants to implement their own devices or applications that support Vendor ID, are there any reasons that they would prefer supporting joint account or multiple activations? What are the business incentives in choosing one over the other?

Consumers will benefit if a device can work with all other vendors. This is all built upon the design philosophy of promoting an open eBook system.

Q: How does the device deactivate if multiple accounts are logged in? Is the device required to ask user which account to logout?

All the accounts will be logged out upon deactivating. There is no method to deactivate a particular user account.

# Multiple Activations Workflow in Adobe Digital Editions Preview

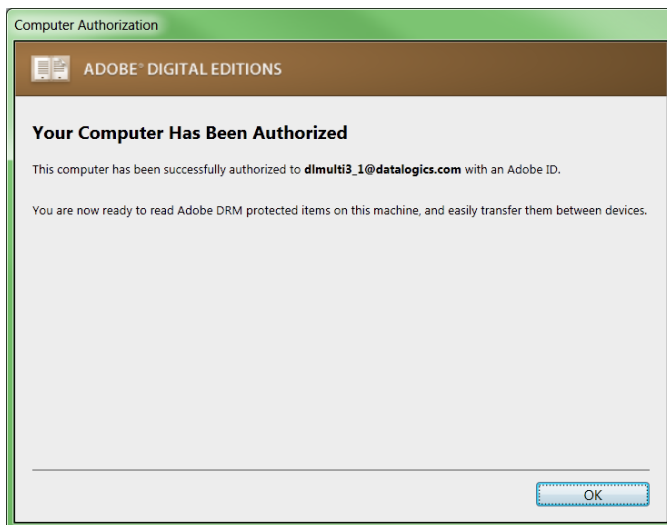
---

Currently released ADE 1.7 does not support multiple activations. ADE Preview that can be downloaded from Adobe Labs ([labs.adobe.com](http://labs.adobe.com)) has support for multiple activations. See the steps below for details.

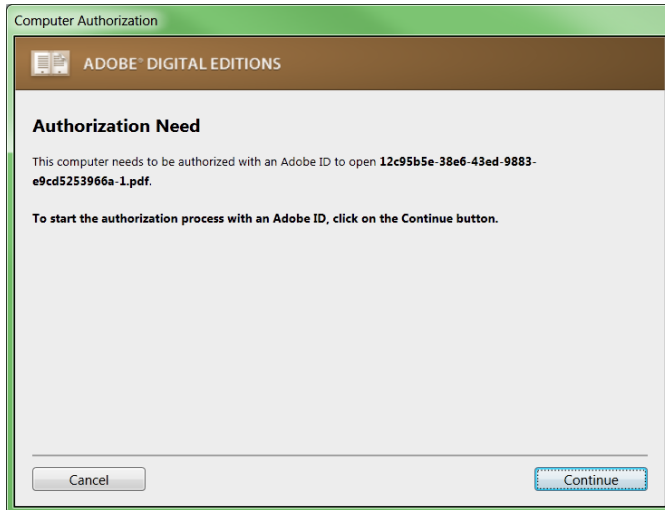
Step 1. Open ADE Preview, activate with one user account:



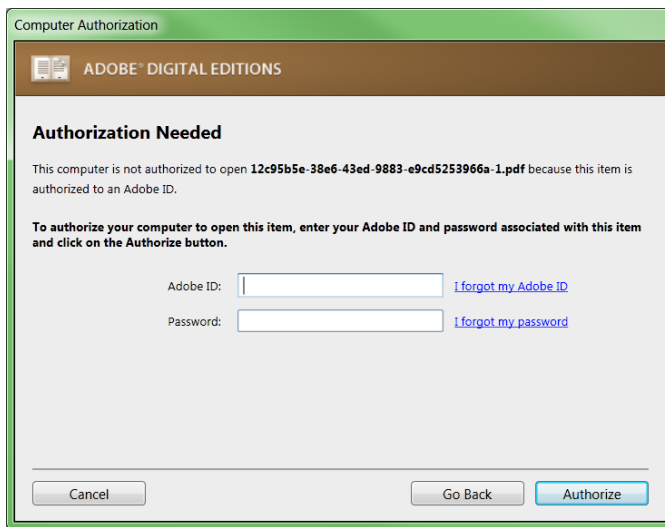
Step 2. ADE Preview shows the user info:



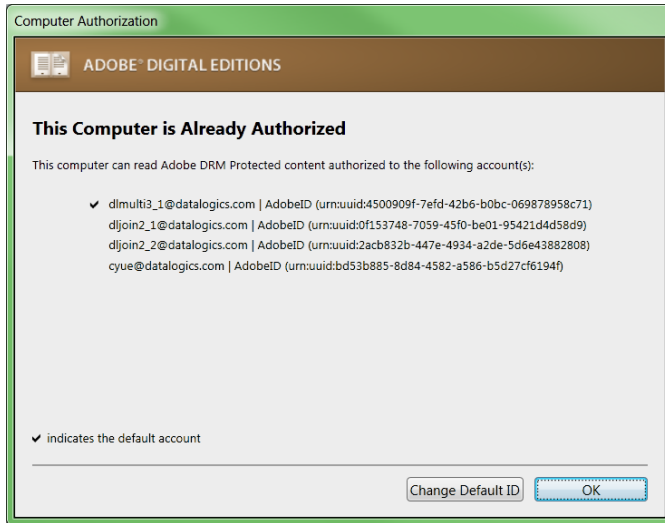
Step 3. Open a book that is fulfilled with a different Adobe ID account, ADE Preview will prompt for the message that the book needs to be authorized:



Step 4: Hit Continue, you will be taken to the authorization dialog:



Step 5: At this point, I entered one of my accounts that were joined with some other accounts, the reader is now activated with all these accounts:



Note that ADE Preview sets the first user as the default. This can be changed by using the “Change Default ID” button.

## Notes for developing multiple activations

---

If you only want to support one activation, then before you start an Activation workflow, you must delete the current activations (call `dpdev::Device::setActivation` record with a null `dp::Data` object).

If you want to support multiple activations, while this is possible, you do need to deal with the concept of an Active user. To set the active user for a particular fulfillment you should use `dpdrm::DRMProcessor::setUser` prior to calling `initWorkflow`. The application should provide a UI workflow to allow readers to set the “default” user to fulfill books with.

Note a couple deficiencies in the activation implementation:

- There is no supported way to delete a single activation record.

- There is no way to reorder/change the first (default) activation record.

If `DRMProcessor` has `setUser()` is not called to set the default user, RMSDK uses the default user, which should match the first valid activation in version 9.2. Version 9.1 may have behave differently. However, the important thing to note here is to make sure that you always call `setUser` to set the default user explicitly instead of relying on the RMSDK to find the default user.