



SSF Tools: Access Request Attachments User Guide

Document Revision History

Revision Date	Written/Edited By	Comments
May 2017	Paul Wheeler	Initial release with SSDv4.

© Copyright 2017 SailPoint Technologies, Inc., All Rights Reserved.

SailPoint Technologies, Inc. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SailPoint Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Restricted Rights Legend. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of SailPoint Technologies. The information contained in this document is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Regulatory/Export Compliance. The export and reexport of this software is controlled for export purposes by the U.S. Government. By accepting this software and/or documentation, licensee agrees to comply with all U.S. and foreign export laws and regulations as they relate to software and related documentation. Licensee will not export or reexport outside the United States software or documentation, whether directly or indirectly, to any Prohibited Party and will not cause, approve or otherwise intentionally facilitate others in so doing. A Prohibited Party includes: a party in a U.S. embargoed country or country the United States has named as a supporter of international terrorism; a party involved in proliferation; a party identified by the U.S. Government as a Denied Party; a party named on the U.S. Government's Entities List; a party prohibited from participation in export or reexport transactions by a U.S. Government General Order; a party listed by the U.S. Government's Office of Foreign Assets Control as ineligible to participate in transactions subject to U.S. jurisdiction; or any party that licensee knows or has reason to know has violated or plans to violate U.S. or foreign export laws or regulations. Licensee shall ensure that each of its software users complies with U.S. and foreign export laws and regulations as they relate to software and related documentation.

Trademark Notices. Copyright © 2017 SailPoint Technologies, Inc. All rights reserved. SailPoint, the SailPoint logo, SailPoint IdentityIQ, and SailPoint Identity Analyzer are trademarks of SailPoint Technologies, Inc. and may not be used without the prior express written permission of SailPoint Technologies, Inc. All other trademarks shown herein are owned by the respective companies or persons indicated.

Table of Contents

Introduction.....	4
Prerequisites and Limitations	4
Components	5
Deployment	6
Using Access Request Attachments.....	7
Access Requests.....	7
Approvals.....	8
Appendix: Modifying Existing Provisioning Workflows to use Attachments.....	10

Introduction

Out of the box, IdentityIQ does not provide the ability to attach documents to access requests. This is a commonly-requested feature, as customers sometimes want requesters to justify their need for a requested role or entitlement or to provide additional information. The Access Request Attachments solution addresses this requirement by permitting the requester to upload files as attachments associated with the request so that they can be viewed by the approver to aid in making an approval decision or as background information.

The solution requires a shared storage area for attachments, and a web server pointing to this location which allows viewing of the files. In some cases this could use storage on one of the IdentityIQ application servers, with the application server itself providing the web server functionality, or it could be a dedicated shared storage area and web server, which is the preferred option.

Prerequisites and Limitations

The Access Request Attachments solution has the following prerequisites and limitations:

- A shared filesystem must be available that can be accessed by all the IdentityIQ servers, so the service account used by IdentityIQ must have read/write/delete access to this area, but end users should not have any filesystem access to it.
- A web server must be available for viewing/downloading the file attachments, and this must point to the shared file storage so that files stored there can be accessed using a URL. If no other web server is available, an application server used by IdentityIQ may be used, with the shared storage located inside the folder structure served by the application server.
- This solution should not be thought of as a secure Content Management solution. Anyone who knows the correct URL will be able to access files via the web server, although URLs will be very difficult to guess, as the original filename will be prefixed with a 13-character numeric string (a Java timestamp representation). The web server should therefore not allow a root view of all the files available in the configured attachment folder.
- Periodic file housekeeping, such as removal of files pertaining to old requests, will be a manual process. While the solution provides the ability to limit the maximum size of an attachment and the maximum number of attachments per request, the monitoring of available disk space will need external tools. Deletion of files will prevent those attachments from being viewable in IdentityIQ.
- The solution requires replacement of the default LCM Provisioning and Provisioning Approval Subprocess workflows with customized versions. Versions that will work with 7.0 and 7.1 are provided and will be deployed when building for these versions, but older versions will require modifications to the default workflows. Details of how to do this are provided in the Appendix.
- The solution uses old-style Ext-JS forms, not IdentityIQ 7.x responsive forms. It is possible that future versions of IdentityIQ may remove support for these, in which case the solution will no longer work in its current form.

Components

The following components are used in the Access Request Attachments solution:

File	Location in SSD	Description
FileUploadServlet.java	src/sailpoint/services/standard/lcmattachments	Java servlet class which stores files on the file system and creates/updates a list of "fileParameters" on the WorkflowCase object
LCMAttachments_File_Attachment_Parameters.xml	config/SSF_Tools/LCMAttachments/Source/Custom	Tokenized Custom object storing the settings for the Access Request Attachments solution
LCMAttachments_LCMProvisioningWorkflowWithAttachments.xml	config/SSF_Tools/LCMAttachments/Source/Workflow/7.x	Modified version of the LCM Provisioning workflow for IdentityIQ 7.x
LCMAttachments_ProvisioningApprovalSubprocessWithAttachments.xml	config/SSF_Tools/LCMAttachments/Source/Workflow/7.x	Modified version of the Provisioning Approval Subprocess workflow for IdentityIQ 7.x
LCMAttachments_ApproveAndProvisionSubprocessWithAttachments.xml	config/SSF_Tools/LCMAttachments/Source/Workflow/7.x	Modified version of the Approve and Provision Subprocess workflow for IdentityIQ 7.x
LCMAttachments_Configuration_SystemConfiguration_merge.xml	config/SSF_Tools/LCMAttachments/Source/Configuration/Sample	Merge file to set the access request workflow to the Access Request Attachments version of LCM Provisioning
lcmattachments.target.properties	config/SSF_Tools/LCMAttachments	target.properties file containing the tokens and values used in the Custom object
FileUploader.js	web/scripts/sailpoint	The Javascript file used to provide the uploader functionality in the UI
lcmFileUploadRenderer.xhtml	web/workitem	Renderer file to provide the uploader form to the requester
lcmFileUploadView.xhtml	web/workitem	Renderer file to provide the uploader in the forms
lcmFileUploadWorkItemRenderer.xhtml	web/workitem	Renderer file to provide the attachment view in the approval form
build.custom.FileUpload-Modify-WEB-XML.xml	scripts	Custom build script to modify web.xml in order to map the servlet to a URL

Deployment

To deploy the Access Request Attachments solution, follow the instructions below. If you are using the SSD Deployer tool, steps 1 and 2 will be completed for you if you selected the option to deploy Access Request Attachments. Refer to the SSD Deployer User Guide for more information.

1. In the SSB build.properties file, set the 'deployLCMAttachments' and 'runCustomScripts' properties to 'true'.
2. Copy the tokens from config/SSF_Tools/LCMAttachments/lcmattachments.target.properties into the target.properties file for your environment.
3. Modify the target.properties for your environment:

%%SP_LCM_ATTACHMENTS_BASE_PATH%%

The path for the shared storage where attachments will be stored.

%%SP_LCM_ATTACHMENTS_BASE_URL%%

The URL path for the web server that points to the shared file system location defined above.

%%SP_LCM_ATTACHMENTS_MAX_FILES%%

The maximum number of files that may be uploaded for an individual request.

%%SP_LCM_ATTACHMENTS_MAX_SIZE_IN_MB%%

The maximum permitted size in MB of a single attachment.

%%SP_LCM_ATTACHMENTS_PERMITTED_EXTENSIONS%%

A comma-separated list of file extensions that are permitted for upload.

4. If you are deploying an IdentityIQ version older than 7.0, make a copy of the OOTB LCM Provisioning workflow and the provisioning subprocess, modify them according to the instructions in the Appendix and add to the build.
5. Set the workflow used by IdentityIQ for access requests to be the new "LCM Provisioning with Attachments" workflow. A sample merge file for the System Configuration is provided to do this in the build:

config/SSF_Tools/LCMAttachments/Source/Configuration/Sample/LCMAttachments_Configuration_SystemConfiguration_merge.xml

This can be moved up one level out of the Samples folder to enable it to be deployed with the build.

6. Run and deploy the build (refer to the Services Standard Build User Guide for more information).

Using Access Request Attachments

Access Requests

Once the solution is deployed and the “LCM Provisioning with Attachments” workflow is configured as the access request workflow, when an access request is submitted a screen will be presented to upload attachments.

Attach files to support Access Request 0000000138 for Alice Ford

File attachments

Description:

File:

Select File

Browse

Add File

Reset

Description	File name	View

Complete

A file can be selected using the “Browse” button and a description must be entered. On clicking the “Add File” button, the file will be uploaded to the shared location and will appear in the list of uploaded files.

Attach files to support Access Request 0000000138 for Alice Ford

File attachments

Description:

Details of available access levels

File:

Select File

Browse

Add File

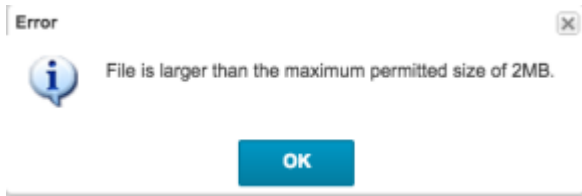
Reset

Description	File name	View
Access requirement	AccessRequirements-AliceFord.docx	View
Details of available access levels	PaymentSystemAccessMatrix.xlsx	View

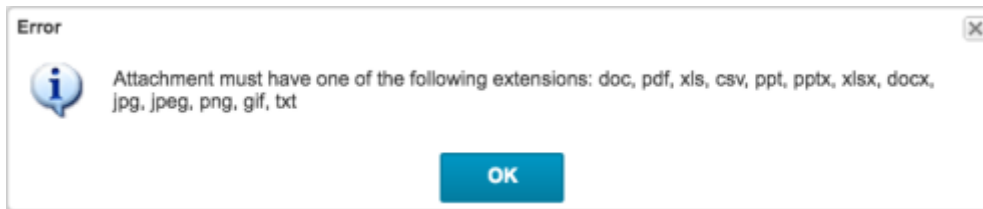
Complete

The uploaded files can be viewed by clicking in the “View” link or removed from the list (and the shared storage) by clicking the red ‘no entry’ icon.

If the requester attempts to upload a file larger than the configured maximum they will be presented with a message and the file will not be uploaded:



Similarly, if the file extension of an attachment is not in the configured list of permitted extensions they will see a message like this:



The request must be completed by clicking the “Complete” button. It is not mandatory to add any attachments, but the Complete button must still be clicked to move the request to the approval stage.

Approvals

When an access request has been submitted and approvals are enabled in the workflow, the configured approver(s) will receive a work item which includes a list of attachments and the ability to view each one:

Account Update Approval

Summary

Work Item ID 140

Access Request ID [1](#)

Requester The Administrator

Owner Amanda Ross

Description Owner Approval - Account Changes for User: Alice Ford

Created May 19, 2017 12:44:58 PM

Priority Normal

History None

Send Comment to Requester

None

[Add Comment](#)

File attachments

Description	FileName	View
Access requirement	AccessRequirements-AliceFord.docx	View
Details of available access levels	PaymentSystemAccessMatrix.xlsx	View



Details

[View Details for Alice Ford](#)

Approval Items pending for Alice Ford

Legend: ● Approve ● Reject

Search: Filter by Decision ▼

Decision	Application	Operation	Attribute	Value(s)	Completion Comments
<input type="checkbox"/> 	IdentityIQ	Add	Role	Accounts Payable Approver	

Page 1 of 1

Displaying 1 - 1 of 1

-- Select Bulk Action -- ▼

Complete

Save

Forward

Cancel

The approver may use the information in the attachments to help with making an approval decision or for background information.

Appendix: Modifying Existing Provisioning Workflows to use Attachments

The solution includes workflows suitable for IdentityIQ 7.0 and 7.1 which are based on the out of the box “LCM Provisioning” and “Provisioning Approval Subprocess” workflows. For previous versions of IdentityIQ and for existing provisioning workflows that have already been customized it will be necessary to make some small changes to add this functionality.

1. Make a copy of your “LCM Provisioning” workflow, or the workflow that you are using for access requests, and name it “LCM Provisioning with Attachments”.

2. Add a new “File Upload Form” step to this workflow, identical to this:

```
<Step name="File Upload Form" posX="79" posY="90">
  <Approval mode="serial" name="File Upload Form" owner="ref:launcher" renderer="lcmFileUploadRenderer.xhtml"
send="identityDisplayName">
  <Arg name="launcher" value="ref:launcher"/>
  <Arg name="workItemDescription" value="script:return &quot;Upload file attachment for &quot; + identityDisplayName;"/>
  <Arg name="workItemRequester" value="$(launcher)"/>
  <Arg name="workItemType" value="Generic"/>
</Approval>
<Transition to="Initialize"/>
</Step>
```

3. Modify the existing “Start” step in this workflow so that it transitions to the “File Upload Form” step:

```
<Step icon="Start" name="Start" posX="25" posY="10">
  <Transition to="File Upload Form"/>
</Step>
```

4. In the step that calls the “Provisioning Approval Subprocess” workflow, change the name of the workflow subprocess that is being called to “Provisioning Approval Subprocess with Attachments”:

```
<WorkflowRef>
  <Reference class="sailpoint.object.Workflow" name="Provisioning Approval Subprocess with Attachments"/>
</WorkflowRef>
```

5. If your “LCM Provisioning with Attachments” workflow has the “Approve and Provision” and “Approve and Provision Split” steps, modify them in the workflow so that they call the “Approve and Provision Subprocess with Attachments” workflow instead of “Approve and Provision Subprocess”:

```
<WorkflowRef>
  <Reference class="Workflow" name="Approve and Provision Subprocess with Attachments"/>
</WorkflowRef>
```

6. Make a copy of your “Provisioning Approval Subprocess” workflow, and name it “Provisioning Approval Subprocess with Attachments”. In this workflow, change the renderer in the Approval step to “lcmFileUploadWorkItemRenderer.xhtml”:

```
<Step icon="Approval" name="Approval" posX="158" posY="22">
```

```
<Approval mode="ref:approvalMode" owner="call:buildCommonApprovals" renderer="lcmFileUploadWorkItemRenderer.xhtml"
send="identityDisplayName,identityName,approvalSet,flow,policyViolations,identityRequestId">
```

7. In the same Approval step, set the “workItemType” argument to “Generic”:

```
<Arg name="workItemType" value="Generic"/>
```

8. If your version of IdentityIQ has the “Approve and Provision Subprocess” workflow, make a copy of it and name it “Approve and Provision Subprocess with Attachments”. In this workflow, modify the “Approve” step so that it calls the “Provisioning Approval Subprocess with Attachments” workflow instead of “Provisioning Approval Subprocess”:

```
<WorkflowRef>
  <Reference class="sailpoint.object.Workflow" name="Provisioning Approval Subprocess with Attachments"/>
</WorkflowRef>
```