

1. Article Drafts Home	2
1.1 Adobe Reader PDF form help	3
1.1.1 Unsupported Browsers	5
1.1.2 Viewing PDF documents in Chrome	6
1.1.3 Viewing PDF documents in Firefox	8
1.1.4 Viewing PDF documents in IE 8 and IE 9	10
1.1.5 Viewing PDF documents in IE 10	13
1.1.6 Viewing PDF documents in Safari	15
1.1.7 Viewing PDF documents in Windows 8	17
1.2 AI Creating a New User in Transaction Manager	18
1.3 Ben Warner	19
1.3.1 2015-02-11 AvokaDoc Structure Review Meeting	20
1.3.2 File List - Test	22
1.3.3 Groovy Guide	23
1.3.4 How do we manage drafts and publishing of new articles?	24
1.3.5 Product Requirement - Standard	25
1.3.6 Troubleshooting Article - Standard	26
1.4 Charts Test	27
1.5 Collaboration Jobs - Rolling Back To a Previous Step (State)	28
1.6 Composer Performace issue with Pepper Flash	37
1.7 Decision log	38
1.8 Disclaimer	39
1.9 Email Notification Subscriptions Guide	40
1.10 File lists	43
1.11 Form Details in Composer	44
1.12 Form Schema Design and Composer	46
1.13 Form Test Mode in Transaction Manager	48
1.14 Groovy Scripting in Transact Manager	50
1.15 How-to articles	51
1.16 Keeping track of Composer Forms in TM	52
1.17 Managing Notifications	54
1.18 Meeting notes	55
1.19 Multi-Step Groovy Delivery Process	56
1.20 Overview of Search Functionality in Space Pages	59
1.21 Product requirements	60
1.22 Reset forgotten password for Avoka support portal	61
1.23 Resubmitting a Previously Submitted Transaction	64
1.24 Save Challenge Behaviour	68
1.25 test	70
1.26 TField and Locations	71
1.27 TM - Security Audit Logs (user-auth.log)	72
1.28 Transaction Delivery	73
1.29 Transaction Reporting Feed	76
1.30 Transact Submission Flow and Timings	77
1.31 Troubleshooting articles	78
1.32 Wet Signatures in FTX Forms - DO NOT USE	79

Avoka Transact

Article Drafts Home



Welcome to your new space!

Confluence spaces are great for sharing content and news with your team. This is your home page. Right now it shows recent space activity, but you can customize this page in anyway you like.

Complete these tasks to get started

- ✓ **Edit this home page** - Click *Edit* in the top right of this screen to customize your Space home page
- ✓ **Create your first page** - Click the *Create* button in the header to get started
- ✓ **Brand your Space** - Click *Configure Sidebar* in the left panel to update space details and logo
- ✓ **Set permissions** - Click *Space Tools* in the left sidebar to update permissions and give others access

Recent space activity



Christopher Eagar

[Wet Signatures in FTX Forms - DO NOT USE](#) updated Apr 19, 2018 • [view change](#)



Anonymous

[Groovy Guide](#) updated Aug 11, 2017 • [view change](#)



Joshua Miller

[Charts Test](#) updated Nov 24, 2016 • [view change](#)



Anonymous

[Groovy Guide](#) created Oct 21, 2016



Stuart Sharpless

[TM - Security Audit Logs \(user-auth.log\)](#) updated Oct 18, 2016 • [view change](#)

Space contributors

- [Christopher Eagar](#) (1743 days ago)
- [Joshua Miller](#) (2254 days ago)
- [Stuart Sharpless](#) (2291 days ago)
- [Anton Wakimin](#) (2509 days ago)
- [Howard Treisman](#) (2572 days ago)
- ...

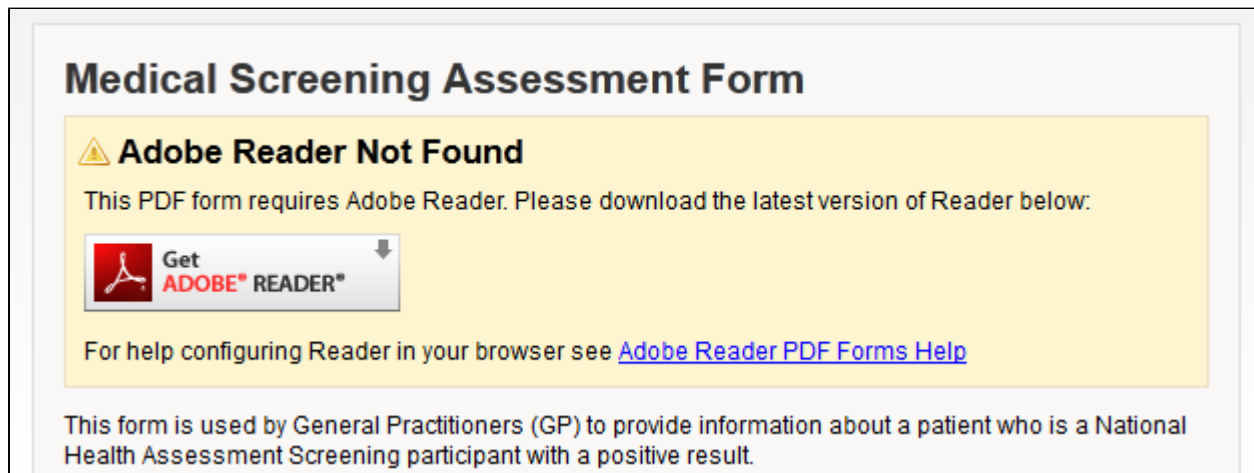
Adobe Reader PDF form help

This page provides up to date information on how to get PDF Forms and Adobe Reader working on your computer.

Please note if you are using an Apple iPad, Android or Windows RT device you will need to use a PC or Mac computer to complete your form. Currently XFA PDF forms are not supported on these tablet devices.

Adobe Reader Not Found

This message is displayed when your browser could not find the Adobe Reader plug-in. This may be due to Adobe Reader not being installed on your computer or the plug-in is not configured in your browser.



To install Adobe Reader please download the free installer from Adobe at <http://get.adobe.com/reader/>. We always recommend you use the latest version of Adobe Reader so you have the latest security updates have the best user experience. It is recommended that you don't use any versions before Reader 9 as these older versions are not supported by Adobe, and will not be updated to address the latest security threats.

If you have Adobe Reader installed by its not available in your browser please check the following browser settings.

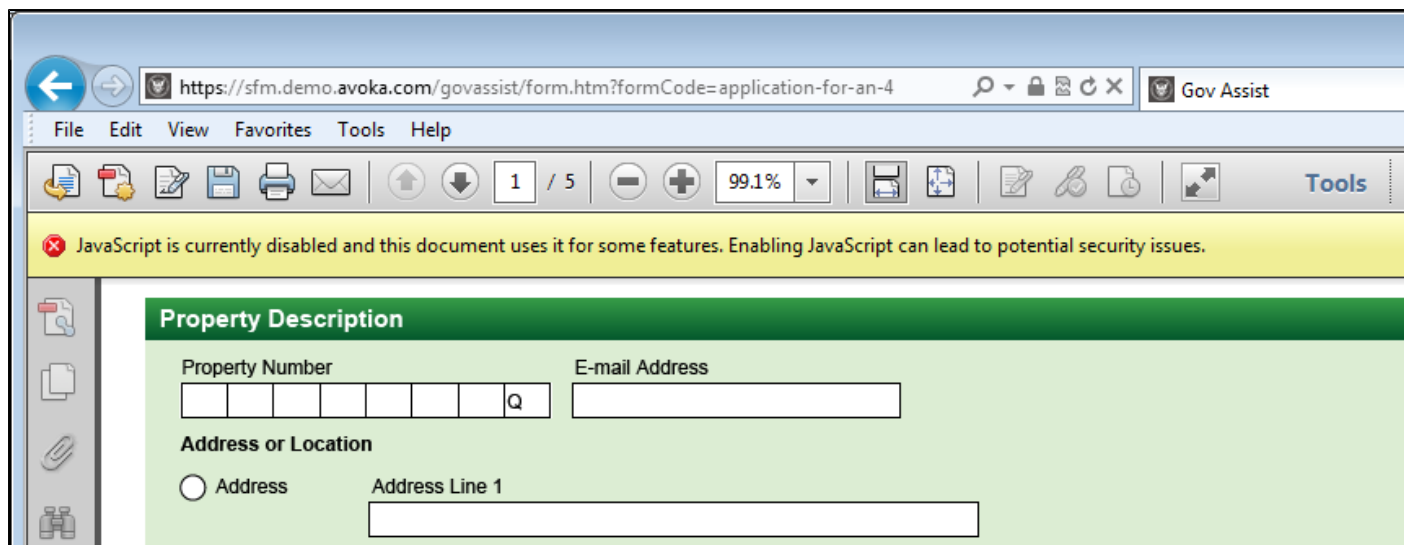
Adobe Reader Update Required

If this message is displayed the PDF form requires a more up to date version of Adobe Reader. Please install the latest version of Adobe Reader from:

<http://get.adobe.com/reader/>

Adobe Reader JavaScript is Disabled

If you open a PDF form and the following message ribbon is displayed then your Adobe Reader is configured to disable JavaScript.



Most PDF forms will not work without JavaScript and you will need to enable it to complete your form. To enable JavaScript:

1. Click on the Options ribbon menu and select the item 'Enable JavaScript for this document always'
2. In the 'Add to Privileged Location' click on the OK button

For more information on particular browsers please refer to

- [Unsupported Browsers](#)
- [Microsoft Windows 8](#)
- [Google Chrome](#)
- [Mozilla Firefox](#)
- [Internet Explorer 10](#)
- [Internet Explorer 8 & 9](#)
- [Safari](#)

Related articles

- [Adobe Reader PDF form help](#)
- [Viewing PDF documents in IE 10](#)
- [Viewing PDF documents in Firefox](#)
- [Viewing PDF documents in Windows 8](#)
- [Viewing PDF documents in Safari](#)

Unsupported Browsers

A number of browser now no longer support Adobe Reader plugins and as such don't support in browser pdf forms. The browsers are listed below:

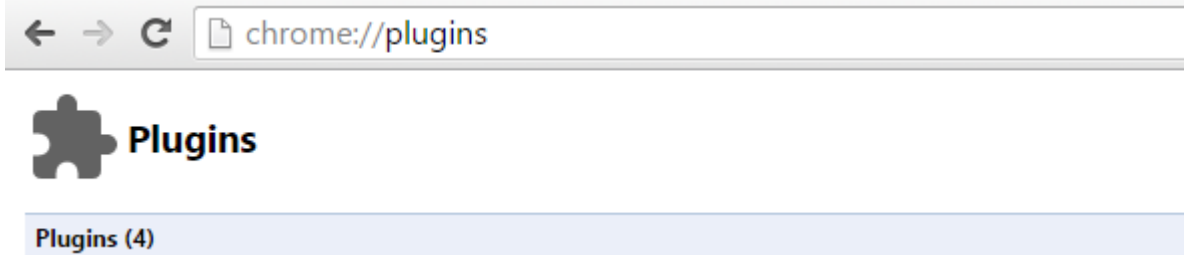
Google Chrome


Google chrome now no longer supports NPAPI(Netscape Plugin API), which Adobe Reader utilized to run from within the browser. To use PDF forms now if you can try one of the following solutions:

- Use a different browser. (eg Firefox, Internet Explorer)
- Disable reader, chrome's default PDF viewer. This will force chrome to open the PDF in the default application for the PC which can be set Adobe Reader.

To disable chrome plugins

1. Open Chrome and in the address bar type `chrome://plugins`



2. No click the details button  , on the right hand side of the page.
3. Scroll down to Chrome PDF Viewer and click disable at the bottom of the plugin information.

Chrome PDF Viewer (2 files)

Name: Chrome PDF Viewer
Version:
Location: `chrome-extension://mhjfbmdgcfjbbpaeojofohoefgihjai/`
Type: BROWSER PLUGIN
[Disable](#)

MIME types:	MIME type	Description	File extensions
	<code>application/pdf</code>		<code>.pdf</code>

Name: Chrome PDF Viewer
Description: Portable Document Format
Version:
Location: `internal-pdf-viewer`
Type: PPAPI (out-of-process)
[Disable](#)

MIME types:	MIME type	Description	File extensions
	<code>application/x-google-chrome-pdf</code>	Portable Document Format	<code>.pdf</code>

[Disable](#) Always allowed to run

4. PDFs will now open to the default App for your machine.

Microsoft Edge

The new Edge browser will not have any support for ActiveX plugins, which is what Adobe Reader uses in its browser plugins. Edge will open PDFs in the default PDF viewer app for your PC. If you still have trouble with PDFs in Edge it is recommended that you use another browser (i.e. Firefox).

Viewing PDF documents in Chrome

This page provides up to date information on how to get PDF Forms and Adobe Reader working on your computer.



By default, Chrome uses a built-in PDF viewer to open PDFs. Google Chrome has a built-in PDF viewer enabled by default. When you click a PDF link on the web, the file opens directly in your browser window.

The following controls are available in the lower-right area of the browser window:



If you've disabled the built-in PDF viewer, Chrome will use Adobe Acrobat or Adobe Reader to display PDFs.

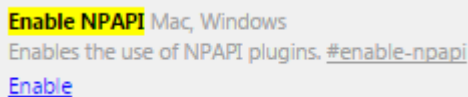


Important Note:

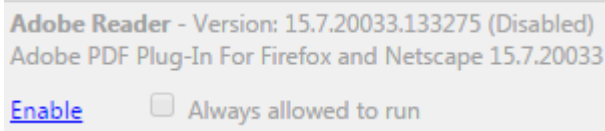
Adobe Reader runs inside Chrome as a NPAPI (Netscape Plugin API) plugin. Google is phasing out support for NPAPI plugins from the Chrome browser. Since April 2015 (Chrome 42) NPAPI plugins have been disabled by default. It is possible to re-enable them via an override however from September 2015 (Chrome 45) this override will be removed and NPAPI support will be permanently removed from Chrome.

To re-enable NPAPI (Note: This is only a temporarily solution until September 2015)

1. In the address bar enter: <chrome://flags/#enable-npapi>
2. Look for the **Enable NPAPI** entry.



3. Click **Enable**.
4. Restart Chrome and then enter in the address bar: <chrome://plugins/>
5. Look for the **Adobe Reader** entry:



6. Click **Enable**. (Note this will make Adobe Reader the default PDF viewer inside Chrome and consequently disable the Chrome PDF Viewer.)

Alternatively, you can set any of the plugin Enterprise policies (e.g. EnabledPlugins, PluginsAllowedForUrls) which will also temporarily re-enable NPAPI support.

After NPAPI support is removed from Chrome, if you need to use Adobe Reader inside a browser, you will need to use another browser until a replacement plugin becomes available for Chrome.

If you've enabled both NPAPI support and either the Adobe Acrobat or Adobe Reader plugin but are still experiencing problems please follow the following steps.

First check if the issue is specific to Chrome. Try opening PDFs in another browser on your computer, such as Firefox. If you experience similar problems viewing PDFs in another browser, the issue is likely not related to Chrome.

Please visit the [Adobe support site](#) for additional help.

If you are still having issues viewing PDF's and it is specific to Chrome, try the following solutions in the order that they're listed:

1. Check for updates to chrome, Adobe Reader and Acrobat

See instructions on [checking for Chrome updates](#). To check for Adobe Reader or Acrobat software updates, visit <http://www.adobe.com/downloads/>.

2. Check your Adobe Reader and Acrobat settings

Follow the instructions below to confirm that either Adobe Reader or Adobe Acrobat is configured to display PDFs in the browser.



Only one of the programs should be set to preview PDFs in the browser, not both.

1. In either Adobe Acrobat or Reader, go to **Edit > Preferences**.
2. Click **Internet** on the left.
3. Select the "Preview PDF in browser" checkbox at the top (may also be called "Display PDF in browser"). If the checkbox is already selected, you may have to deselect it, click **OK**, then access **Preferences** again to reselect it.
4. Click **OK**.
5. Follow the same steps in the other program, but make sure that the "Preview PDF in browser" checkbox is not selected.

3. Repair your Adobe Reader or Adobe Acrobat installation

In Acrobat, go to **Help > Repair Acrobat Installation**. Similarly, in Adobe Reader, go to **Help > Repair Adobe Reader Installation**.

4. Uninstall either Adobe Reader or Adobe Acrobat installation

If you have both Adobe Reader and Adobe Acrobat installed, they may be conflicting with one another. Try uninstalling one of the programs.

Windows instructions:

- a. Use the "Add or Remove Programs" function in your Windows Control Panel to remove either Adobe Reader and Acrobat from your computer.
- b. Restart your computer after you've completed uninstalling the program.

Mac instructions:

- a. Drag either Adobe Reader or Acrobat from the Applications folder to the Trash.
- b. Remove AdobePDFViewer.plugin from the Library > Internet Plug-Ins folder.

5. Uninstall and reinstall Adobe Reader or Acrobat

1. Follow the steps from the section above to uninstall all versions of these programs.
2. Visit <http://www.adobe.com/downloads/> to reinstall the latest version of Adobe Acrobat or Reader.

Related articles

- [Adobe Reader PDF form help](#)
- [Viewing PDF documents in IE 10](#)
- [Viewing PDF documents in Firefox](#)
- [Viewing PDF documents in Windows 8](#)
- [Viewing PDF documents in Safari](#)

Viewing PDF documents in Firefox

This page provides up to date information on how to get PDF Forms and Adobe Reader working on your computer.



Using the built-in PDF viewer

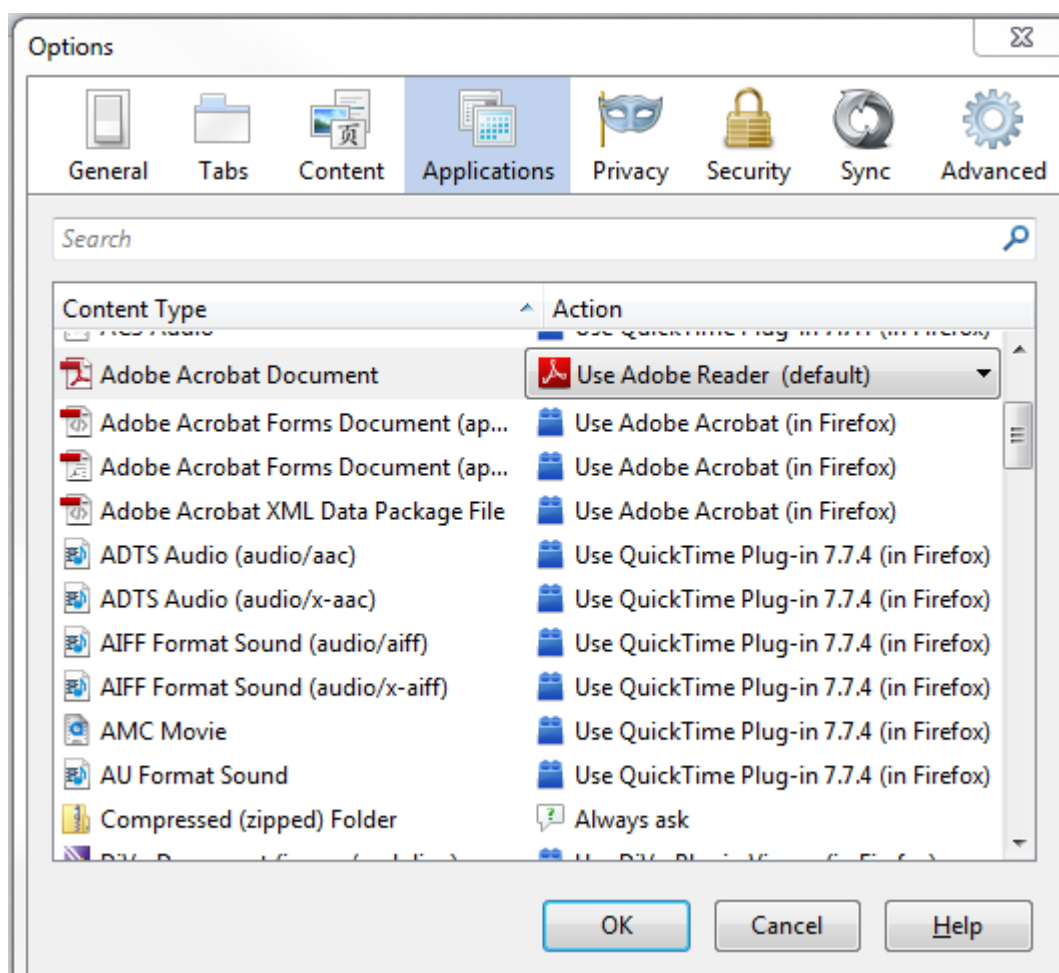
Firefox now includes a built-in PDF viewer that allows you to view almost all PDF files found on the web without a plugin. The built-in PDF viewer is enabled by default.

Using a PDF reader plugin

To check what your Firefox browser is set to you can go to the Tools Menu > Options and click on the Applications tab

To change from using the built-in PDF viewer to your PDF reader plugin:

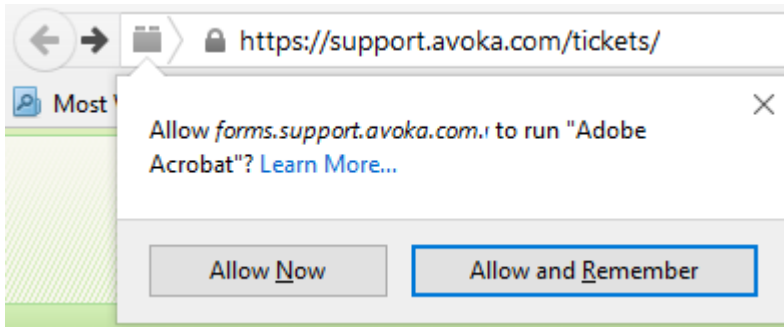
1. At the top of the Firefox window, click on the Firefox button and then select Options
2. Select the Applications panel.
3. Find **Portable Document Format (PDF)** in the list and click on it to select it.
4. Click on the drop-down arrow in the **Action** column for the above entry and select **Use PDF reader name (in Firefox)**.



Note: The Firefox Software Update feature does not update installed plugins. See the [Adobe Security bulletins and advisories page](#) for information about security and stability issues affecting Adobe Reader. To check if you have the latest versions of all your plugins, go to our [Plugin Check page](#).

If plugin is not working

Firefox can sometimes block plugins from working on certain webpages. If the PDF is not rendering check the plugin is allowed, click the the plugin icon in the address bar and click allow adobe reader.



Testing the Adobe Reader Plugin

Visit the [Mozilla Privacy Policy PDF](#), which is in PDF format, to test whether you are able to open PDF files.

If PDF files are downloaded instead of displayed, try these solutions

Reset download actions

1. Open your profile folder:

At the top of the Firefox window, click on the Firefox button, go over to the Help menu and select Troubleshooting Information. The Troubleshooting Information tab will open.

2. Under the **Application Basics** section, click on Show Folder. A window with your profile files will open.

Note: If you are unable to open or use Firefox, follow the instructions in [Finding your profile without opening Firefox](#).

Related articles

- [Adobe Reader PDF form help](#)
- [Viewing PDF documents in IE 10](#)
- [Viewing PDF documents in Firefox](#)
- [Viewing PDF documents in Windows 8](#)
- [Viewing PDF documents in Safari](#)

Viewing PDF documents in IE 8 and IE 9

This page provides up to date information on how to get PDF Forms and Adobe Reader working on your computer.



Internet Explorer is supported for use with Acrobat and Adobe Reader version 7.0.8 and higher. For versions of Acrobat earlier than 7.0.8, most functionality will work, but there may be issues with forms workflows.

If you run into any sort of issue, then you should upgrade to the latest version of Acrobat Reader (and if possible your browser version).

Internet Explorer 8 makes it more visible to the user which add-ons are enabled and disabled. This is done from the tools menu by selecting Manage Add Ons.

Using an Add-on to view PDF documents within an IE8 browser window

When you install Acrobat or Adobe Reader on a system that includes a browser, you automatically configure the browser to open PDF files within the browser window. Acrobat and Adobe Reader install ActiveX plug-in files that allow you to use Internet Explorer or AOL to open PDF files.

If either application is installed, but PDF files don't open in the browser window, then work through the following steps to configure the browser:

1. Make sure that the system meets the requirements for the Acrobat product you use.

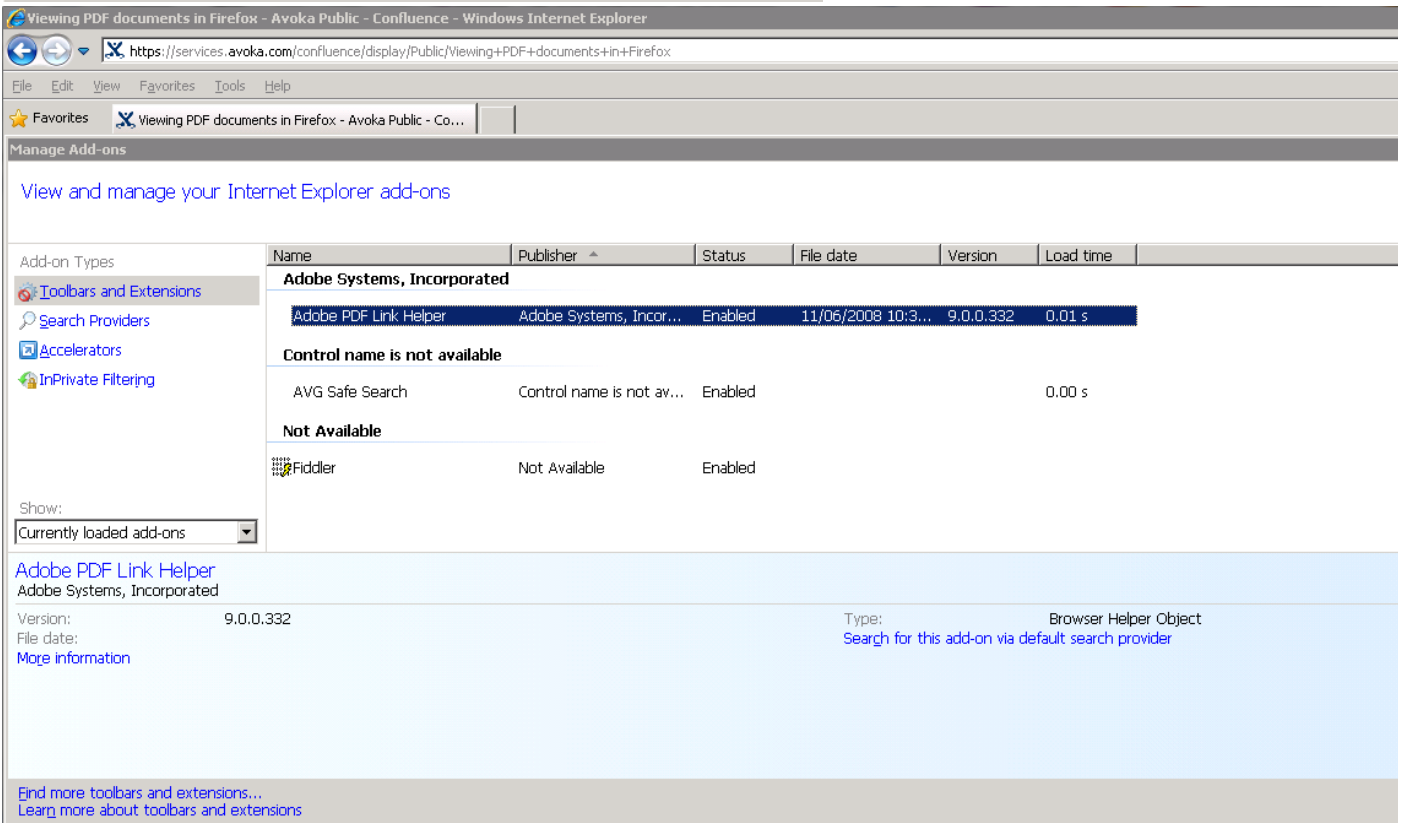
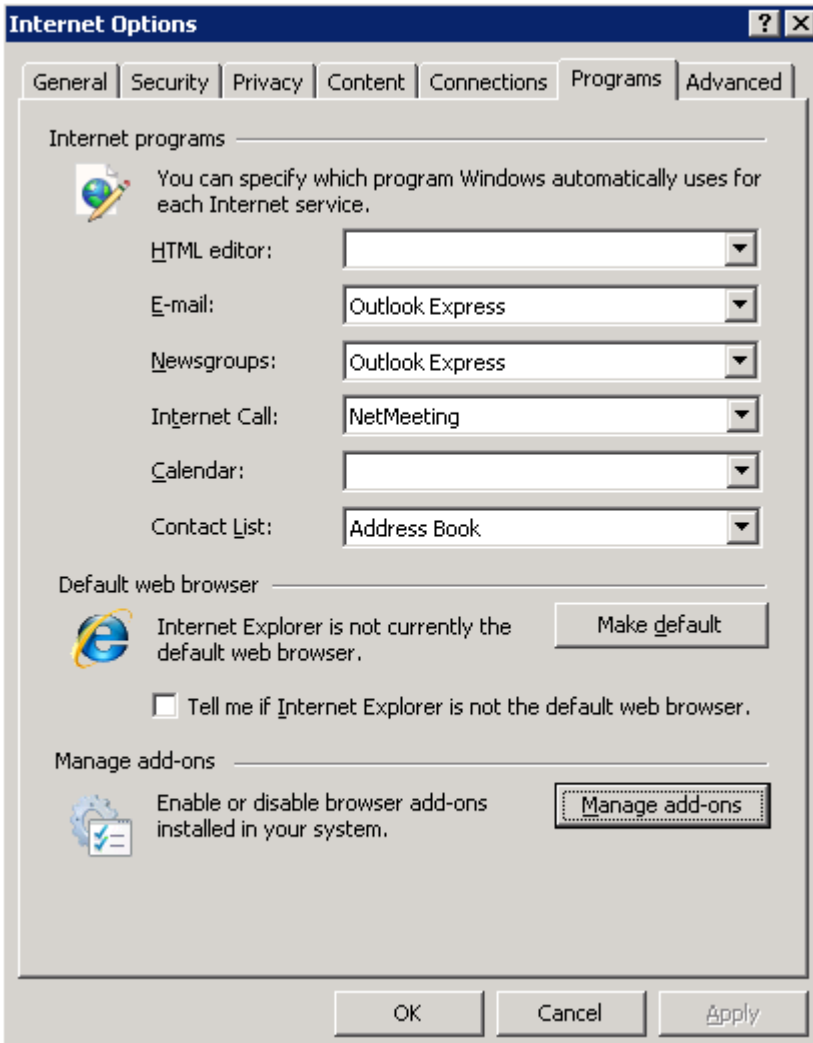
Acrobat may not run correctly if the system doesn't meet the following requirements:

- Intel Pentium III processor or equivalent
- Microsoft Windows 2000 with Service Pack 4 or Windows XP Professional, Home Edition, or Tablet PC Edition with Service Pack 2
- Microsoft Internet Explorer 6.0
- 256MB of RAM (512MB recommended)
- 860MB of available hard-disk space; cache for optional installation files (recommended) requires an additional 460MB of available hard-disk space.
- 1,024x768 screen resolution

2. Check your IE add-on

Internet Explorer 8 makes it more visible to the user which add-ons are enabled and disabled. This is done from the tools menu by selecting Manage Add Ons.

1. Open Internet Explorer by clicking the Start button . In the search box, type Internet Explorer, and then, in the list of results, click Internet Explorer.
2. Click the Tools button , and then click Manage add-ons.
3. Under Add-on Types, click Toolbars and Extensions.
4. Under Show, you can select one of the following views of your add-ons:
 - To display a complete list of the add-ons on your computer, click All add-ons.
 - To display only those add-ons that were needed for the current webpage or a recently viewed webpage, click Currently loaded add-ons.
 - To display add-ons that were preapproved by Microsoft, your computer manufacturer, or a service provider, click Run without permission.
5. When you're finished, click Close.



3. Make sure that the ActiveX folder contains files required for web browsing.

To open PDF files in a browser window, the ActiveX folder must contain the following files: AcroPDF.dll, AcroIEHelper.dll, and pdfshell.dll. The ActiveX controls should be located in: Program Files\Common Files\Adobe\Acrobat\ActiveX. If any of these files are missing, then repair Acrobat or Adobe Reader by choosing Help > Repair Acrobat Installation, or reinstalling the application.

4. Use a supported version of the browser.

Make sure that you use a version of your web browser that Acrobat and Adobe Reader support.

Supported browsers for Adobe Acrobat 8, Adobe Reader 8 are Internet Explorer 6.0, 6.0 SP1, Internet Explorer 7.0, Firefox 1.5 or later, Mozilla 1.7, and AOL 9.

5. How to turn on and off add-ons

1. Open Internet Explorer by clicking the Start button . In the search box, type Internet Explorer, and then, in the list of results, click Internet Explorer.
2. Click the Tools button , and then click Manage add-ons.
3. Under Show, click All add-ons.
4. Click an add-on, and then do one of the following:
 - To make the add-on available for use in the browser, click Enable.
 - To turn the add-on off so it can't be used in the browser, click Disable.
5. Repeat step 4 for each add-on that you want to turn on or off. When you're finished, click Close.

Note: Before you turn off a browser add-on, keep in mind that some webpages, or Internet Explorer itself, might not display properly if an add-on is turned off.

We recommend that you only turn off an add-on if it repeatedly causes Internet Explorer to close.

6. To delete Active X controls

You can only delete ActiveX controls that you've downloaded and installed. You can't delete ActiveX controls that were preinstalled or add-ons of any kind, but you can turn them off.

1. Open Internet Explorer by clicking the Start button . In the search box, type Internet Explorer, and then, in the list of results, click Internet Explorer.
2. Click the Tools button , and then click Manage add-ons.
3. To display all ActiveX controls, under Show, click Downloaded controls.
4. Click the ActiveX control you want to delete, and then click More information.
5. In the More Information dialog box, click Remove. If you're prompted for an administrator password or confirmation, type the password or provide confirmation.
6. Repeat steps 4 and 5 for every control you want to delete. When you're finished, click Close.

For more information please refer to the Windows website <http://windows.microsoft.com/is-IS/windows7/How-to-manage-add-ons-in-Internet-Explorer-9>

Related articles

- [Adobe Reader PDF form help](#)
- [Viewing PDF documents in IE 10](#)
- [Viewing PDF documents in Firefox](#)
- [Viewing PDF documents in Windows 8](#)
- [Viewing PDF documents in Safari](#)

Viewing PDF documents in IE 10


This page provides up to date information on how to get PDF Forms and Adobe Reader working on your computer.

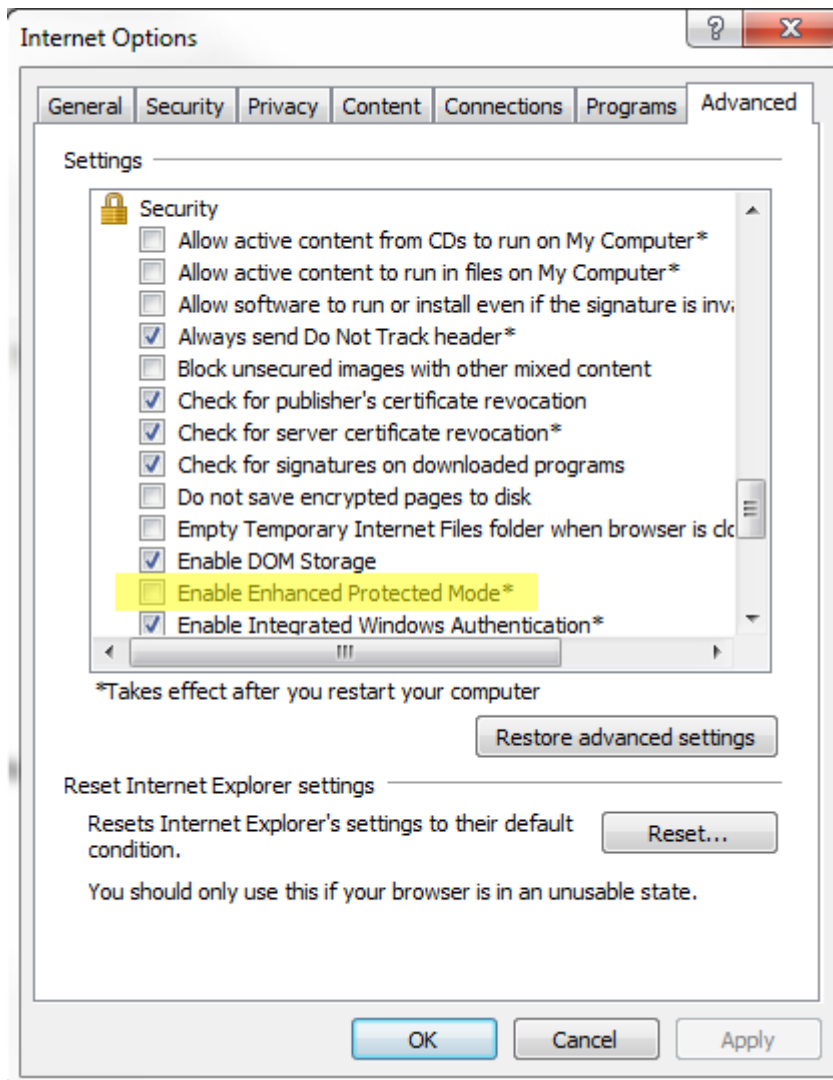


Using the built-in PDF viewer

Internet Explorer includes a built-in PDF viewer that allows you to view almost all PDF files found on the web without a plugin. If your PDF's are opening in a native application such as Acrobat Reader check the following settings in IE.

Check your Browser Settings

1. At the top of the IE window, go to the Tools Menu  and select Internet Options
2. Click on the Advanced Tab
3. Scroll to the Security settings
4. Uncheck the box for Enable Enhanced Protected
5. Restart your computer and PDFs should now open in your IE browser.



Still having Issues? Try this steps

Method 1: Disable Add-Ons

1. Click the 'Tools' button, and then click 'Manage add-ons'.
2. Disable the add-ons by clicking them one at a time to highlight them and then click on disable button.
3. Re-enable the add-ons one by one and check with which Add-on you get this error message.
4. Disable the Add-on causing the problem.

Method 2: Reset security settings for Internet Explorer

If you configure security settings to be too restrictive, you may prevent Internet Explorer from displaying certain Web sites. To determine whether an issue is caused by overly restrictive security settings, revert to default security settings. To do this, follow these steps:

1. Open **Internet Explorer**.
2. Click **Tools**, and then click **Internet Options**.
3. Click the **Security tab**.
4. Click **Reset all zones to default level**, and then click **OK**.

Method 3: Run the Internet explorer performance troubleshooter and check if it helps.

1. Press the **Windows + W** key on the keyboard.
2. Type **troubleshooter** in the **search** box and then press **enter**.
3. Select **Troubleshooting**. Click on **view all** and select **Internet explorer performance**.
4. Click **next** and follow the on-screen instructions.

Method 4: Reset Internet Explorer 10

1. Drag the mouse to the bottom right corner and click on search.
2. Click on **'Settings'** and type **'Internet Options'**.
3. Click on **Internet Options** and click on **'Advanced tab'**
4. Click **'Reset'** and click **'OK'**.

Note: Please note that running Reset Internet Explorer Settings will reset all user-defined settings including those set by installed extensions, toolbars and other add-ons to IE Defaults. This includes all Security, Privacy and Zone settings. Also this will clear browsing history, delete all temporary Internet Files, cookies, form data and especially all stored passwords.

Method 5: uninstall the Adobe Acrobat Reader and install the latest version.

Follow the link below to download the latest version:

<http://get.adobe.com/reader/otherversions/>

Also, if you have multiple Adobe Readers installed, uninstall them all then install the latest version.

Adobe doesn't support multiple versions of Reader or Acrobat on the same computer. Multiple versions simultaneously installed can lead to software conflicts and errors. Uninstall all versions of Reader or Acrobat and then reinstall only one version of the product.

Related articles

- [Adobe Reader PDF form help](#)
- [Viewing PDF documents in IE 10](#)
- [Viewing PDF documents in Firefox](#)
- [Viewing PDF documents in Windows 8](#)
- [Viewing PDF documents in Safari](#)

Viewing PDF documents in Safari

This page provides up to date information on how to get PDF Forms and Adobe Reader working on your computer.



The Adobe Reader and Acrobat stand-alone applications are unaffected and Safari 5.1 renders PDF documents natively. However, the Adobe Reader and Acrobat browser plug-ins before 10.1.3 don't function as expected in LiveCycle and Acrobat workflows that require either plug-in to render PDF documents in Safari 5.1.

Please note Adobe Reader for XFA PDF forms is not available on iPad or iPhone devices.

System requirements

Acrobat X and Adobe Reader X support viewing PDF files inside the Safari browser on Mac OS X. For more information about system requirements, click these links for your product: [Acrobat X Pro](#), [Acrobat X Standard](#), [Adobe Reader X](#).

Adobe PDFViewer for Mac OS X requires the following system requirements:

- Safari 4 for Mac OS X 10.5.8 and Mac OS X 10.6.4
- Safari 5 for Mac OS 10.6.4
- Acrobat and Reader via AdobePDFViewer Safari plug-in are 32/64-bit (64-bit available in Safari 10.6)

Note: If you don't have Acrobat or Reader, or haven't installed the PDFViewer plug-in, then Safari shows PDF files using the Mac OS X native PDF support. Support for many Adobe PDF workflows (such as the Collaboration feature or many Forms and Security features) is unavailable.

Configuring Acrobat to display PDF files in Safari

The AdobePDFViewer plug-in is used to display PDF files in Safari using Acrobat and Reader. This plug-in is installed as part of the Acrobat X or Reader X installation.

The location of this plug-in on a Macintosh is:
/Library/Internet Plug-ins/AdobePDFViewer.plugin

Troubleshooting issues in Safari

Options include the following:

- Continue to use Safari 5.0.x and Mac OS X 10.6.
- Upgrade to Adobe Reader or Acrobat version 10.1.3 or later.
- Save the PDF file from Safari to the computer, and then open it directly in Adobe Reader or Acrobat.

For more information on troubleshooting viewing PDFs within Safari please go to

<http://helpx.adobe.com/acrobat/kb/troubleshoot-safari-plugin-acrobat-x.html>

Known Issues

Reader's Preferences > Internet > Display PDF in Browser is always dimmed

- To disable Safari integration, delete the AdobePDFViewer.plugin from /Library/Internet Plug-ins
- To reenable Safari integration, rerun the installer over the current installation. Start the Acrobat/Reader X installer from the original media, and then follow the onscreen prompts to complete the installation.

Uninstalling an older version of Acrobat or Reader after Acrobat X or Reader X has been installed removes the AdobePDFViewer plug-in

If Acrobat X or Reader X is installed alongside an older version of Acrobat or Reader, removing the older version removes the AdobePDFViewer plug-in.

To reenable Safari integration, rerun the installer over the current installation. Start the Acrobat/Reader X installer from the original media, and then follow the onscreen prompts to complete the installation.

Older versions of the AdobePDFViewer plugin can overwrite newer versions of the plug-in

Having multiple versions of Acrobat or Reader installed on the same system can cause plug-in conflicts. For example, suppose you have both Acrobat 9 and Acrobat X on your system and Acrobat 9 gets updated. In this case, the Acrobat 9 version of the AdobePDFViewer plug-in can overwrite the Acrobat X version of the plug-in. To correct the issue, follow these steps:

- Remove the previous version of Acrobat or Reader.
- Reinstall Acrobat X or Reader X from the original media.

Related articles

- [Adobe Reader PDF form help](#)
- [Viewing PDF documents in IE 10](#)
- [Viewing PDF documents in Firefox](#)
- [Viewing PDF documents in Windows 8](#)
- [Viewing PDF documents in Safari](#)

Viewing PDF documents in Windows 8

This page provides up to date information on how to get PDF Forms and Adobe Reader working on your computer.



To make PDF forms work in Windows 8 you need to make Adobe Reader the default PDF Reader, as the Reader provided by Microsoft Windows does not support PDF forms.

Once you have Adobe Reader installed and configure please access forms with Internet Explorer in the Windows Desktop UI mode. To switch to Desktop mode press the keys 'Windows + D'.

The Adobe Reader plugin is not compatible with Internet Explorer running in start screen or Metro UI mode.

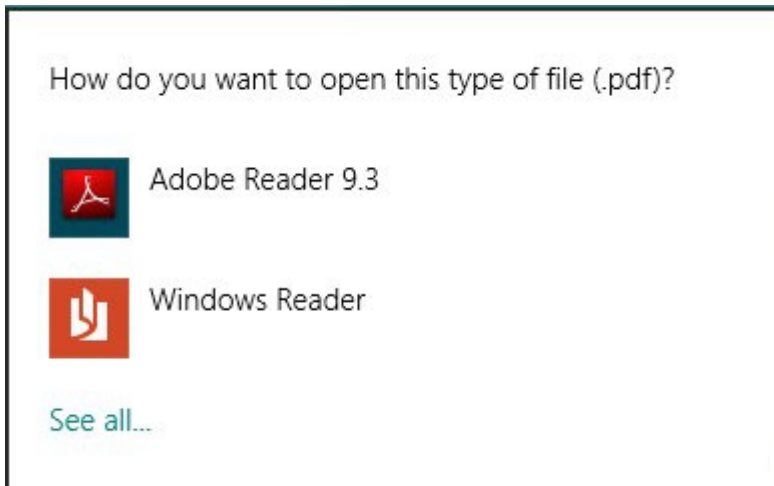
To make Adobe Reader the default PDF Reader please follow the instructions below:

Using Acrobat Reader to be the Default PDF viewer

Windows 8 doesn't associate Acrobat Reader as default application for PDF files after installing it.

If you want to open all your PDF files using Acrobat Reader by default, here's what you need to do:

1. Right click on any PDFfile and click "Properties".
2. Click on "Change" button against the "Open With" option.
3. You will get a new window which shows valid applications associated with PDF files. Just click on Adobe Acrobat.



Once you have Adobe Reader installed and configure please access forms with Internet Explorer in the Windows Desktop UI mode. To switch to Desktop mode press the keys 'Windows + D'.

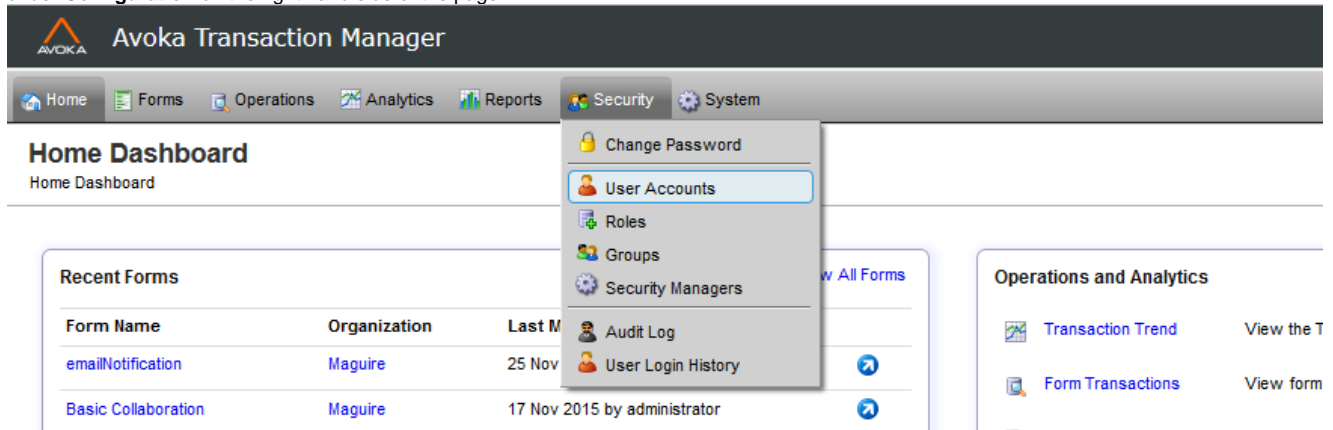
The Adobe Reader plugin is not compatible with Internet Explorer running in start screen or Metro UI mode.

Related articles

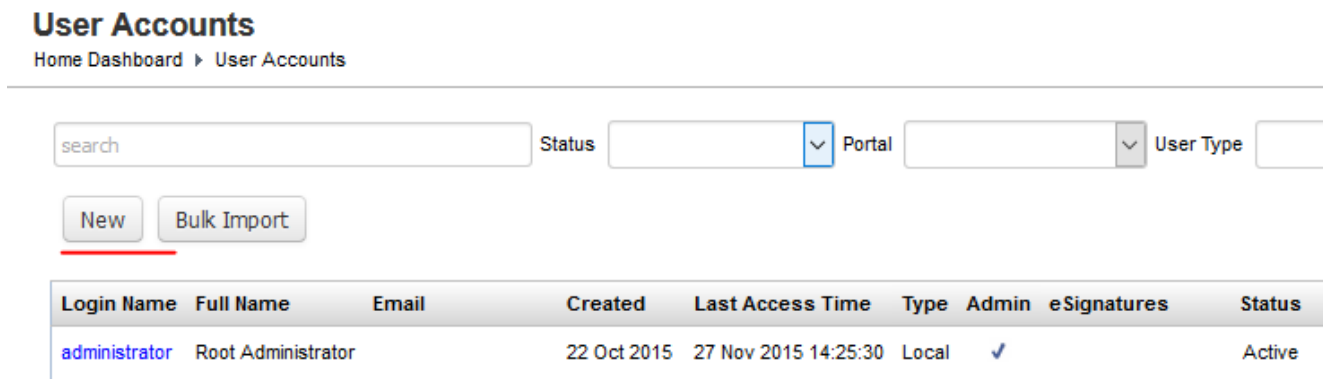
- [Adobe Reader PDF form help](#)
- [Viewing PDF documents in IE 10](#)
- [Viewing PDF documents in Firefox](#)
- [Viewing PDF documents in Windows 8](#)
- [Viewing PDF documents in Safari](#)

AI Creating a New User in Transaction Manager

1. From the home dashboard select the **Security** menu and click **User Accounts**. The user accounts link can also be access from the dashboard under **Configuration** on the right hand side of the page.



2. From the User Accounts page click the New button, under the search fields.



3. Fill in the required information for the new user and click **Save** at the bottom of the form. Note: your password will need to be entered for verification.

The screenshot shows the 'User' creation form. It has several tabs: 'User', 'Organizations', 'Spaces', 'Roles', 'Groups', 'User Profiles', and 'Login History'. The 'User' tab is active. The form is divided into 'User Details' and 'Verify Your Password' sections.

User Details

- Login Name*
- User Type* Local
- Password
- Confirm Password
- Change Password After Login
- Given Name(s)*
- Family Name*
- Email*
- Mobile
- Phone
- Spaces*

Verify Your Password

- Your Password*

Buttons: Save, Close

4. Now that the new user is created it can be assigned to roles, organisations, groups and spaces. Navigate through the tabs to view each property. Note: your password will need to be enter for verification to make changes to the user details.

Ben Warner

Task report

Looking good, no incomplete tasks.

2015-02-11 AvokaDoc Structure Review Meeting

Date

11 Feb 2015

Attendees

- [Unknown User \(bwarner\)](#)
- [John Clagnan](#)
- [Howard Treisman](#)
- [Larry Bunton](#)
- John T

Goals

- Discuss usage of AvokaDoc for Customer Content Management


Discussion items

Item	Discussion	Actions	Results
Granularity of Spaces	General agreement that a higher degree of granularity will be better to manage content than fewer larger spaces.	Unknown User (bwarner) to propose opening list of spaces All others to review and provide feedback	Proposed list of spaces: <ul style="list-style-type: none"> • Transact Integrations • The Avoka Salesforce App • Transact Analytics • Transact Collaboration Jobs • Composer Form Development • Maguire Forms for Composer • Maguire Forms for Maestro • Transact Portals
Types of Spaces	General agreement that Knowledge Base spaces and Documentation spaces are required, but that Team spaces not yet required. New group 'avoka' created for all internal avoka staff. All new spaces to have same permissions where confluence users (customers) only have view access and the ability to add comments. Users in the avoka group have full access.	Unknown User (bwarner) to modify permissions on space templates accordingly.	Team space disabled until such a time as it is considered relevant.
Types of Articles	Which article types will we use?		Disabled the following article templates: <ul style="list-style-type: none"> • Product Requirements • Decision • Meeting notes (like this one) • Task Report • Retrospective Still active: <ul style="list-style-type: none"> • How to guide • Trouble shooting • File list • blank page
User Identity	How do we know which Company a client user is from? There is no company field. Suggestion to use 'Department' to store the Company.	Unknown User (bwarner) to check with Anton if we can add Company field to user profile	
Groups	avoka group for internal staff. Possible future requirement for 'partner' group		Avoka group created. No further groups required at this stage
Process for Publishing	Either use drafts space and move content (has issues with hard coded values in links & macros) or author in place with restricted access before publishing (preferred). Issue where manual step required to restrict access to an article when the author creates it.	Unknown User (bwarner) to check with Anton if we can automatically set the restriction to avoka group on new articles.	
Authentication and User Management	Many systems that client users require authentication to, each with different identity management. Ideally we would have a registry of users maintained separately for single sign-on purposes across Zendesk, Confluence, and possibly Maestro. Atlassian Crowd a possible tool: https://www.atlassian.com/software/crowd/overview	Unknown User (bwarner) to check with Anton if we can setup single sign-on or single identity management.	

Templates	Modification to standard templates to suit purpose	Unknown User (bwarner) to make modifications	
-----------	----------------------------------------------------	----------------------------------------------	--

File List - Test

Test Files

File	Modified 
GIF File spinner.gif	Feb 11, 2015 by bwarnar
GIF File wait.gif	Feb 11, 2015 by bwarnar
PNG File spinner.png	Feb 11, 2015 by bwarnar

Drag and drop to upload or [browse for files](#)

 [Download All](#)

Groovy Guide

Groovy is a powerful scripting language which runs on the Java Virtual Machine. This gives you access to all the facilities provided by Transaction Manager and Java, while providing a scripting language without the steep learning curve of Java.

You can think of Groovy as JavaScript for the server. It is a dynamic language, there are no compile and deploy steps. Just write it and run it.

Avoka Transact provides customized Groovy runtime with security constraints for running in an application server context. For Groovy version information, see 'Groovy Lang' under [3rd Party Java Libraries](#).

Simple Example

The simple hello world example below defines a variable called name, and assigns it the value 'World'. Then on the next line, this variable \$name is included in the Hello text and this line is printed using the println function.

```
def name = 'World'
println "Hello $name!"
```

And as expected the result is:

```
Hello World!
```

Learning Groovy

If you are familiar with programming in a modern language then you will be able to quickly learn Groovy. It has a C like syntax in common with C#, JavaScript, and Java, but has scripting features so you don't have to write lots of code.

This Groovy Guide will provide you with a quick and gentle introduction to the language, but will give you the skills you need to develop Groovy Service scripts.

As you are working through the Groovy Guide examples, please use the [Groovy Script Console](#) under the System menu. This provides you a play ground to try out examples and prototype Groovy scripts.

Script Type Checking

Where possible configure your services to use the Groovy script Type Checked option. When using the option the script will be parsed before executing to see whether there are any script type errors. These compilation style errors messages are generally more informative than the standard Groovy dynamic execution errors.

The example below has the Type Checked option and is showing the cause of the error and its line number.

Without type checking the corresponding error is illustrated below. Note this error includes no line number and you would need to search your script to find the corresponding missing method.

```
Thu Apr 17 11:41:54 EST 2014
2014-04-17
groovy.lang.MissingMethodException: No signature of method: java.util.Date.formatter() is applicable for
argument types: (java.lang.String) values: [yyyy-MM-dd'T'HH:mm:ss]
Possible solutions: format(java.lang.String), format(java.lang.String, java.util.TimeZone)
    at org.codehaus.groovy.runtime.ScriptBytecodeAdapter.unwrap(ScriptBytecodeAdapter.java:55)
    at org.codehaus.groovy.runtime.callsite.PojoMetaClassSite.call(PojoMetaClassSite.java:46)
    at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCall(CallSiteArray.java:45)
    at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:108)
    at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:116)
    at Script1.run(Script1.groovy:9)
    at groovy.lang.GroovyShell.evaluate(GroovyShell.java:518)
    at groovy.lang.GroovyShell.evaluate(GroovyShell.java:556)
    at groovy.lang.GroovyShell.evaluate(GroovyShell.java:527)
```

Please note when using the Groovy Type Checked option you may need to write your scripts in a more statically declared Java like fashion to enable the Groovy runtime to figure out what you are trying to achieve.

To configure Groovy Type Checked on your service please configure the service definitions groovyTypeChecked Service Parameter to be true. By default this option is set to off, to ensure backward compatibility with older scripts.

Next Steps

The next topic discusses [Groovy Declarations](#).

Once you have completed this Groovy Guide and you want to learn more please see the [Groovy Language Documentation](#).

How do we manage drafts and publishing of new articles?

Status	IN PROGRESS
Stakeholders	John Clagnan
Outcome	
Due date	11 Feb 2015
Owner	Unknown User (bwarner)

Background

Action items

- Meet to discuss publishing articles [Unknown User \(bwarner\)](#) [John Clagnan](#)

Product Requirement - Standard

Target release	
Epic	
Document status	DRAFT
Document owner	Unknown User (bwarner)
Designer	
Developers	
QA	

Goals

Background and strategic fit

Assumptions

Requirements

#	Title	User Story	Importance	Notes
1				

User interaction and design

Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome

Not Doing

Troubleshooting Article - Standard

Problem


Solution


Related articles

- [Adobe Reader PDF form help](#)
- [Viewing PDF documents in IE 10](#)
- [Viewing PDF documents in Firefox](#)
- [Viewing PDF documents in Windows 8](#)
- [Viewing PDF documents in Safari](#)

Charts Test

[Test Lnk](#)

 Unknown macro: 'iframe'

 Unknown macro: 'sql'

Collaboration Jobs - Rolling Back To a Previous Step (State)

This article deals with how to roll back a Collaboration Job with the use of a Groovy Script.

It requires someone (preferably from Avoka) that is competent with

- [Transact Collaboration Jobs](#)
- Creating groovy scripts for Transact
- TM Administration Console
- TM Groovy Console.

This article discusses the:

- Background as why we would want to roll back a job.
- Details of an actual issue where we had to roll back.
- *Job Entity* State before and after roll back.

It also describes a step by step example with working Groovy Scripts that runs on TM 4.2+

Background

The diagram below shows a simple review and approval collaboration job.

It typically consists of the following steps:

- **Start Step:** (Applicant Start)
An end user initiates a collaboration job via a form submission.
The user has spent time and effort filling in the form.
It may be difficult to get the user to fill in the form again.
- **Internal Review Step(s):** (Applicant Review)
A reviewer will read the form, select a route such as Approve or Reject, and optionally enter some instructions or reason for their decision.
- **Collaboration Step(s):** (Not shown on Diagram)
A participant is assigned a task (or tasks) and may have to spend time filling in one or more sections of the form.

In particular when a Job has been started in **Production**, we wish to avoid:

- getting the initiator to refill and submit the form
- losing all the work that has already been entered as the job progresses through review or collaboration steps.

Rolling back a job to a previous Step or State may mean that work does not have to be completed again.

Actual Issue

[AVT-3235](#)

There was a multi-step approval job that stopped, because a bug in the form caused the form to fail due to the 'set a route name'.

- The Review Step was a simple User Task that contained User Task Action, Task Wait Action and had 2 routes **Approve** or **Reject**.
- The Task form submission sent a blank route name where Job was expecting **Approve** or **Reject**.
- As the route name was blank it could not route the job to a new step.
- The Action completed, Step completed and because it could not route to another step the Job Completed.

Here is the definition of the step from the Job Definition JSON

Step Definition

```
{
  "name": "Review",
  "type": "",
  "actions": [
    {
      "name": "Assign Review",
      "type": "Job Task Assign",
      "properties": [
        { "name": "Task Assign User", "value": "anne" },
        { "name": "Task Form Code", "value": "$func.startFormCode()" },
        { "name": "Task Message", "value": "Please review the ${submission.formName} by ${formDataMap.firstName} ${formDataMap.lastName}." },
        { "name": "Task Review Previous Step", "value": "true" },
        { "name": "Task Subject", "value": "Review ${submission.formName} by ${submission.contactEmailAddress}." },
        { "name": "Task Send Email", "value": "true" },
        { "name": "Task Assign Portal", "value": "Web Plug-in" },
        { "name": "Task Type", "value": "Review" }
      ]
    },
    {
      "name": "Review Wait",
      "type": "Job Task Wait"
    }
  ],
  "routes": [
    { "name": "Approve", "nextStep": "Application Approved" },
    { "name": "Reject", "nextStep": "Application Rejected" }
  ]
}
```

Summary of the Status for the Job and Submission Entities:

- Job :
 - Status: **Completed**
 - Steps:
 - JobStep (Review)
 - Status: **Completed**
 - Actions:
 - JobAction (Assign Review - Job Task Assign)
 - Status: **Completed**
 - Associated **Submission** Status: **Submitted**
 - Job Action (Review Wait - Job Task Wait)
 - Status: **Completed**

The result of the Collaboration Job State is that the Job will not be processed further.

Steps To Rectify

We needed to:

- modify the route name in form submission.
- roll the job back to the point where the last the step can be reprocessed.

I will describe one of the strategies for fixing this issue:

- fix and update the form template
- change the state of the submission from **Submitted** to **Saved**.
form status = **Saved**,
delivery status = **Not Ready**
We may want to remove the original receipt (set to null)
- change the state of the *Job* and *JobStep* entities to **In Progress**
- change *JobAction* state to be the same as they were before the task was submitted.
To do this refer to the diagram below, taken from the [Job Basics](#) - **Wait Action Processing** section.
- User:
Open saved Task, updates route name to Approve
Click Submit: which then kicks off the Job and completes the Job on the **Application Approved** step

Example

It is hard to repeat the example that was presented above. I have created a scenario using the **1 Step Review Job** that comes as a Job Controller Service Template in Transaction Manager 4.2.

Job Definition - 1 Step Review

```
{
  "jobDetails": {
    "name": "1 Step Review Job",
    "version": "4.1.0"
  },
  "steps": [
    {
      "name": "Application Start",
      "type": "start",
      "actions": [
        {
          "name": "Accept Quote",
          "type": "Job Form Start",
          "properties": [
            { "name": "Process Message Send Email", "value": "true" },
            { "name": "Process Message Text", "value": "Thank you ${formDataMap.firstName} ${formDataMap.lastName} your ${submission.formName} is being processed." }
          ]
        }
      ],
      "routes": [
        { "name": "Default", "nextStep": "Application Review" }
      ]
    },
    {
      "name": "Application Review",
      "type": "",
      "actions": [
        {
          "name": "Assign Review",
          "type": "Job Task Assign",
          "properties": [
            { "name": "Task Assign User", "value": "anne" },
            { "name": "Task Form Code", "value": "$func.startFormCode()" },
            { "name": "Task Message", "value": "Please review the ${submission.formName} by ${formDataMap.firstName} ${formDataMap.lastName}." },
            { "name": "Task Review Previous Step", "value": "true" },
            { "name": "Task Subject", "value": "Review ${submission.formName} by ${submission.contactEmailAddress}." },
            { "name": "Task Send Email", "value": "true" },
            { "name": "Task Assign Portal", "value": "Web Plug-in" },
            { "name": "Task Type", "value": "Review" }
          ]
        }
      ],
      "routes": [
        { "name": "Review Wait", "type": "Job Task Wait" }
      ]
    },
    {
      "name": "Application Delivery",
      "type": "",
      "actions": [
        {
          "name": "Application Delivery",
          "type": "Job Delivery"
        },
        {
          "name": "Application Delivery Wait",
          "type": "Job Delivery Wait"
        }
      ],
      "routes": [
        { "name": "Default", "nextStep": "Application Completed" }
      ]
    }
  ],
  {

```

```

"name": "Application Completed",
"type": "endpoint",
"actions": [
  {
    "name": "Process Message",
    "type": "Job Process Message",
    "properties": [
      { "name": "Process Message Send Email", "value": "true" },
      { "name": "Process Message Submission", "value": "$func.startSubmission()" },
      { "name": "Process Message Text", "value": "Thank you ${formDataMap.firstName} ${formDataMap.
lastName} your ${submission.formName} has been Approved." }
    ]
  }
],
},
{
  "name": "Application Rejected",
  "type": "endpoint",
  "actions": [
    {
      "name": "Process Message",
      "type": "Job Process Message",
      "properties": [
        { "name": "Process Message Send Email", "value": "true" },
        { "name": "Process Message Submission", "value": "$func.startSubmission()" },
        { "name": "Process Message Text", "value": "Sorry ${formDataMap.firstName} ${formDataMap.lastName}
your ${submission.formName} has been declined." }
      ]
    }
  ]
}
]
}
}

```

1. A form is submitted and a reviewer Anne routes it to **Reject**.
 2. The job is routed to the *Application Rejected* endpoint.
 3. Anne has made a mistake and actually wants to **Approve** the transaction.
-
1.
 - a. We will use a groovy script to roll back the Job state to the *Application Review* step.
 - b. Anne will then open the form, select **Approve** and submit.
 - c. The Job will move to *Application Approved*.

The following table shows the AS IS and TO BE states of the relevant Job Entities.

	Job	JobStep	JobAction	Linked Submission	JobAction
		Application Review	Assign Review	Assign Review	Review Wait
			Job Task Assign		Job Task Wait
AS IS	Completed	Completed	Completed	Submitted	Completed
TO BE	In Progress *	In Progress *	Assigned **	Saved	Pending **

* ref [Job Basics](#) See Job Status and Step Status sections

Pre Requisites

1. Setup a new TM user "anne". Give her access to the Maguire Portal and Maguire Org.
2. Create a new Job Controller from 1 Step User Template. Edit the Job Definition Assign **Task Assign User** to "anne" as follows

```

"name": "Application Review",
"type": "",
"actions": [
  {
    "name": "Assign Review",
    "type": "Job Task Assign",
    "properties": [
      { "name": "Task Assign User", "value": "anne" },
      { "name": "Task Form Code", "value": "$func.startFormCode()" },
      { "name": "Task Message", "value": "Please review the ${submission.formName} by ${formDataMap.firstName} ${formDataMap.lastName}." },
      { "name": "Task Review Previous Step", "value": "true" },
      { "name": "Task Subject", "value": "Review ${submission.formName} by ${submission.contactEmailAddress}." },
      { "name": "Task Send Email", "value": "true" },
      { "name": "Task Assign Portal", "value": "Web Plug-in" },
      { "name": "Task Type", "value": "Review" }
    ]
  }
],

```

3. Edit Form Version Services - **Job Example 1**. Set Job Controller to be our job controller above

Steps to Replicate

1. Create Job Controller from Template. Assign Task Assign User to "anne"

```

{ "name": "Task Assign User", "value": "anne" },

```

2. Reviewer (Anne) finds the task

3. Completes by Selecting Reject - Note the Test Field A

4. Check Receipt for the submission

5. In TM Operations -> Collaboration Jobs we can get details for the Job Submission Entities.

6. Find the JobKey

7. Find the Job Action Key for the Task Assign and Task Wait

8. Groovy Script to run in Groovy Console

When running a new instance the groovy script below needs to have the following substituted

- JobKey (Step 6)

- Job Action Key Task Assign and Task Wait substituted. (Step 7)

ModifyJobState.groovy

```
import com.avoka.fc.core.dao.DaoFactory
import com.avoka.fc.core.entity.Job
import com.avoka.fc.core.entity.JobAction
import com.avoka.fc.core.entity.JobStep
import com.avoka.fc.core.entity.Submission
import com.avoka.fc.core.service.ServiceFactory
import com.avoka.fc.core.service.job.impl.JobActionUtils

// Pause the Schedule Jobs - Collaboration Jobs Before running script
final STEP_NAME_REVIEW = 'Application Review'
final STEP_NAME_REJECTED = 'Application Rejected'

// Code to change Job Status to inprogress
def job = DaoFactory.getJobDao().getJobForKey("06d54ee8b06732a12e00f56e7f1cf6ac")
job.status = Job.STATUS_IN_PROGRESS
def jobStepReview = JobActionUtils.getLastStep(job, STEP_NAME_REVIEW)
job.setCurrentStep(jobStepReview)

// Code to set Job action status for the Task Assign
def jobActionTaskAssign = DaoFactory.getJobActionDao().getJobActionForKey("665962a1dd92e9bbe21b3bcb283003e8")
jobActionTaskAssign.status = JobAction.STATUS_ASSIGNED

// Code to set the status to saved.
Submission submission = jobActionTaskAssign.submission
submission.formStatus = Submission.STATUS_Saved
submission.deliveryStatus = Submission.STATUS_NotReady
// Reset the receipt data to null in groovy
submission.receiptStatus = null
```

```
submission.submissionData.receiptData=null

// Code to reinstate JobStep
// def jobStepReview = jobActionTaskAssign.getJobStep() // Alternative way of getting the JobStep
jobStepReview.status = JobStep.STATUS_IN_PROGRESS
def jobActionWait = DaoFactory.getJobActionDao().getJobActionForKey("bac62ab48f539f94e13e839ad1a8bald")
jobActionWait.status = JobAction.STATUS_PENDING

// We need to unlink the 'Application Rejected' Endpoint step from the job. Note we are prevented from delete it
in Groovy.
def jobStepReject = JobActionUtils.getLastStep(job, STEP_NAME_REJECTED)
def jobActionRejectMessage = jobStepReject.getJobActions().get(0);

// there isn't a way to do this in Groovy that can do this "JobControllerService.delete"+Object(jobStepReject);"

// CommitChanges
ServiceFactory.getSubmissionService().commitChanges()
```

 **Important**

Remember to check the groovy console output

9. Check the Job State. It should be similar to this

10. Anne Finds the Task again and then can fill it out selects

11. Check Receipt has regenerated using the Test field and Route.

12. The New status

13. Action List.

Note

It still has the **Application Rejected** step which I haven't been able to get rid of. However this acts as an Audit Log entry so it is ok to leave it.

Related articles

- [Transaction Delivery](#)
- [Transaction Reporting Feed](#)
- [TField and Locations](#)
- [Collaboration Jobs - Rolling Back To a Previous Step \(State\)](#)
- [test](#)

Composer Performance issue with Pepper Flash

The Composer Editor uses Adobe Flash for its user interface. Google Chrome has its own version of Flash, known as Pepper Flash. (Technically it's not a Google implementation of Flash, it's a different way of embedding Adobe's Flash inside Chrome). We have seen some fairly severe performance problems running Composer in Chrome using Pepper Flash.

If you (or your customers) experience these problems, then you can try one of the following solutions:

- Use a different browser. (eg Firefox)
- Disable Pepper Flash, and use the Adobe default. Instructions below:
<https://helpx.adobe.com/flash-player/kb/enable-flash-player-google-chrome.html>

Decision log

[Create decision](#)

Decision	Status	Stakeholders	Outcome	Due date	Owner
How do we manage drafts and publishing of new articles?	IN PROGRESS	John Clagnan		11 Feb 2015	Unknown User (bwarner)

Disclaimer

Copyright (c) 2016 Avoka Technologies Pty. Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

In terms of this disclaimer, the term Software includes snippets of configuration included in KB articles.

Email Notification Subscriptions Guide

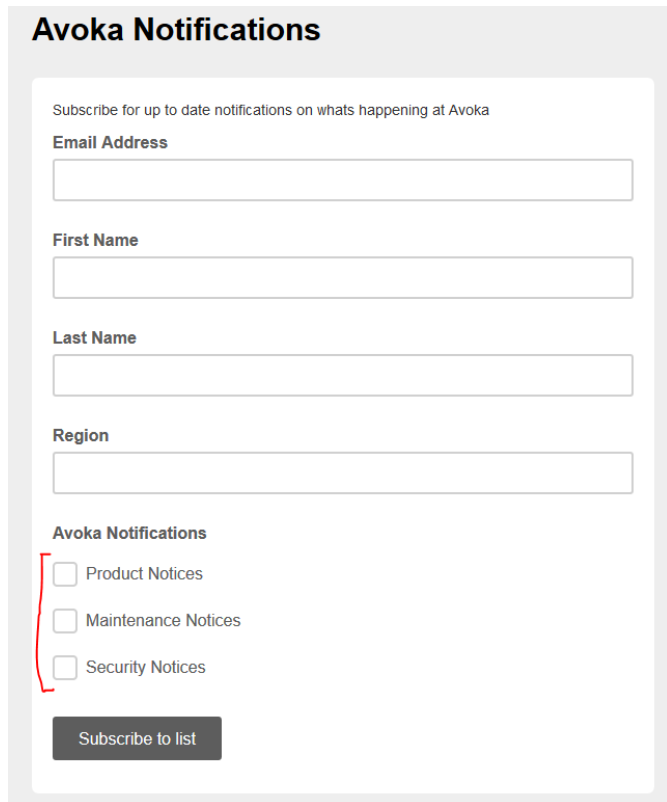
Avoka makes regular notifications about new or updated products and services, maintenance schedules for services and important security notices. It is possible to manage who receives these notifications by subscribing or unsubscribing from the mailing groups used.

Subscribing to Avoka Email Notifications

The link below will subscribe a user to Avoka's Email notifications.

[Subscribe Here](#)

Here you can decide which notifications you want to receive by selecting the check boxes for each of the relevant groups.



Avoka Notifications

Subscribe for up to date notifications on whats happening at Avoka

Email Address

First Name

Last Name

Region

Avoka Notifications

- Product Notices
- Maintenance Notices
- Security Notices

Managing your Subscriptions

From the subscribe page if a user enters their email address the system will detect the existing Account and offer to update the details.

Subscribe for up to date notifications on whats happening at Avoka

There are errors below

Email Address

jmillier@avoka.com

jmillier@avoka.com is already subscribed to list Avoka Notifications. [Click here to update your profile.](#)

First Name

Josh

Last Name

Miller

Region

AUS

Avoka Notifications

- Product Notices
- Maintenance Notices
- Security Notices

Subscribe to list

This will send an email to your registered email address.

Email Sent

For security, we've sent an email to your inbox that contains a link to update your preferences.

Avoka Technologies
Level 2, 1A Rialto Lane
Manly, NSW 2095
Australia

[Add us to your address book](#)

The link in the email direct the user to an update details page. The user can then subscribe to or unsubscribe from any of the mailing groups they desire.

Update your preferences

Email Address

j*****@a****.com

First Name

Josh

Last Name

Miller

Region

AUS

Avoka Notifications

- Product Notices
- Maintenance Notices
- Security Notices

Update Profile

or [Unsubscribe](#)

Profile Updated

Your profile information has been updated. For your records, here is a copy of the information you submitted to us...

Email Address: j*****@a****.com

First Name: Josh

Last Name: Miller

Region: AUS

Avoka Notifications: Product Notices, Maintenance Notices, Security Notices

« return to our website

File lists

[Create file list](#)

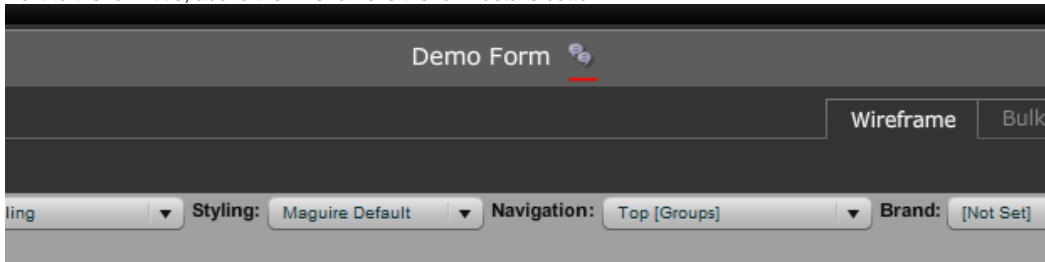
Title	Creator	Modified
File List - Test	Unknown User (bwarner)	Feb 11, 2015

Form Details in Composer

This guide will provide instructions for retrieving details for a form in composer.

Step-by-step guide

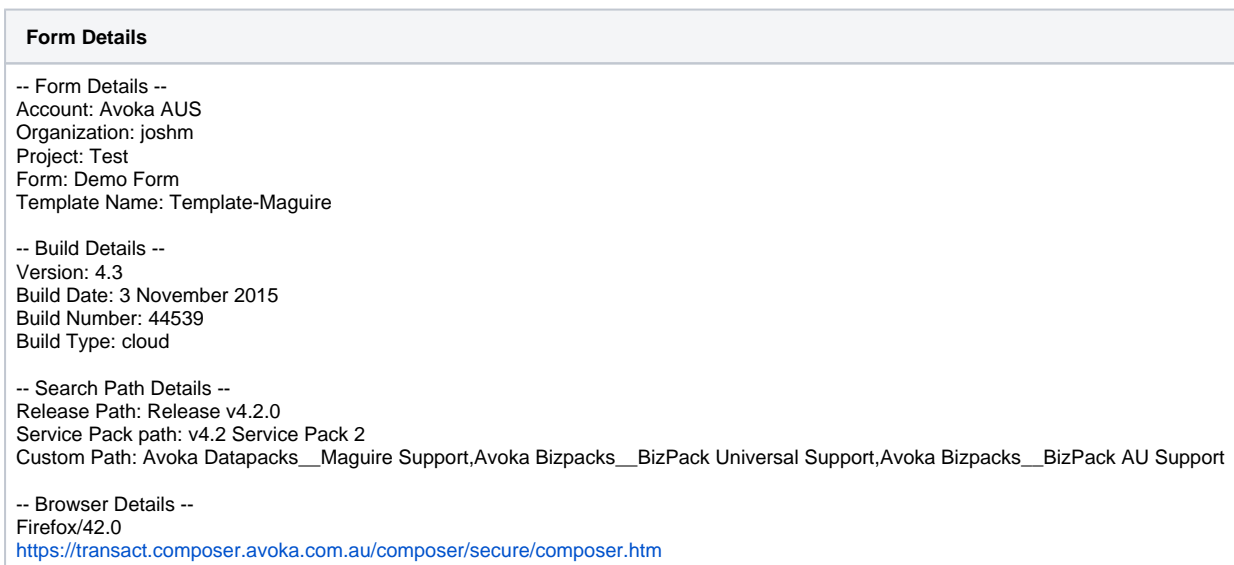
1. Open your form in composer.
2. Next to the form title, above the wireframe is the form details button.



3. Clicking this brings up a form details view from which you can copy the details to be sent or saved as desired.



4. The details copied from this dialog appear as follows.



5. This information is used to help support staff diagnose problems with forms by providing useful information related to the form in composer.



Related articles

- [Transaction Delivery](#)
- [Transaction Reporting Feed](#)
- [TField and Locations](#)
- [Collaboration Jobs - Rolling Back To a Previous Step \(State\)](#)
- [test](#)

Form Schema Design and Composer

Summary

I will answer why good form schema / seed file design is important.

I will discuss the issues with some of the ways composer map fields does by default and why this is bad, the data model tab and absolute bindings.

If only this Alternative Method in Composer Worked! We will do the mapping quickly and create a seed file with the minimum of fuss.

Note

This needs reviewing by Client Services form developers

The Alternate Method is very useful for quickly prototyping forms

Background

A stable structure for the Form XML is important for integration into other systems. The form schema is a contract with the other system. The old form best practice for PDF was to design the form schema separate from the form structure. It means that structural changes to the form do not affect the integration. If a text field is moved from **section1.blah** to **section2.blah** it doesn't require a change to the designed schema / seed file.

Form Schema / Seed File Design

We use any existing form data schema / seed file exists. This may be the case when we are replacing existing forms that have been integrated into back end systems (Systems of Record). In this case we may choose to use the absolute bindings.

When working on a new form or collaboration job I have a preference for the seed files to be flattened. This makes composer mapping and integration easier. The example below is a seed file I created I have dummied up to show how to bind fields..

Sample flattened form

```
<AvokaSmartForm>
  <Application>
    <FirstName />
    <LastName />
  </Application>
</AvokaSmartForm>
```

Composer Mapping Fields to Schema

Composer Generated Seed File

Composer default behaviour is to generate its own seed file its created based on structure below.

Composer generates its own seed file that looks like this.

Lets say we move the Last Name field.

The generated seed now looks like this

The time saved not designing a separate schema comes back to bite you when changes are required. Because this is Composer default behaviour I see this happen repeatedly with lot on client projects and even client services projects.

The workaround for an existing form is to absolute bindings on the fields that are moved.

Provide your own seed file

Composer does allow you to upload a separate schema seed file. When designing a form in composer go into the Data Model Tab. Click on icon to add a data connection.

The Data Model tab allows you to map fields into seed file by dragging it. This creates an absolute binding as per the next section below.

Disadvantages

- You have to create a seed file before
- People have found this to be tricky and time consuming
 - Navigating the Data Model hierarchy on the left isn't easy.
 - The user has to do a lot of clicks.
 - Its also easy to make mistakes and miss binding.

Set Absolute Bindings Directly (Preferred Method)

Mapping each field with an Absolute binding. Below the First Name to the seed file can be done as follows.

Advantages

- Can be done from the main screen.
- Can produce a seed file.
- Fields are easy to see in the composer tree list (in basic view)
- It is easy to create a spreadsheet that has all the form field definitions. Creating an column that calculates the value of the absolute binding so it can be a cut and paste..

Disadvantages

- Not Automatic. Every field requires an absolute binding.
- Still requires a number of clicks.
- Section Level 1 and 2 need to be unbound.

If only this Alternative Method in Composer Worked

The following was a failed alternative based upon a trick I stole from Ben Warner. I use it to create test forms for Collaboration Jobs. This does the mapping in a lot less steps and produce a seed file.

It takes advantage of Composers Field Name mappings. A field name First Name creates a xml node <FirstName> seed file..

It gets around the structure issue in the generated file above by mapping multiple sections / containers to the same xml node. Note this is the bit that doesn't work.

We take a copy of the original sample form.

Edit Properties of the Section Level 1 S1A. In the data model tab, Change the binding to "Skipped - Process Children"

Repeat for S1B

Edit S1A2A and set and absolute binding as follows.

Here is where it differs and FAILED.

If I repeat above for S1B2A. Note this means both S1A2A and S1B2A will have a absolute binding to \$record.Application

Note we don't have to do anything to the Fields. Below is the Data Modal for the First Name field.

Produce a seed file from Composer. In the Data Model Tab click Preview Seed xML

However This Does Not Work. As the Application node is only binding fields from the Section Level 2 S1B2A.

Back to using Absolute Bindings!

Related articles

- [Transaction Delivery](#)
- [Transaction Reporting Feed](#)
- [TField and Locations](#)
- [Collaboration Jobs - Rolling Back To a Previous Step \(State\)](#)
- [test](#)

Form Test Mode in Transaction Manager

When a new form is deployed to a Transaction Manager(TM) instance it often needs to be tested before it is ready for production. This is an important stage in form development particularly for forms with more complex functionality such as custom prefill or delivery services. In order to avoid disruption to production forms and services a form in TM can be set to Test Mode. Test Mode allows for the forms functionality to be tested whilst controlling what it affects through the use of a separate delivery channels.

Important

Note: While a form is in test mode submissions made will not count towards your organisation's transactions count. It is important that organisations that are not running Dev/Uat environments set their forms to test mode while they are testing before the form is released for general use.

Setting A Form To Test Mode

In TM navigate to your form. On the Form Details tab is a check box option labled Test Mode.

Dashboard	Details	Flow Config	Email Verification	Form Versions	Abandonment	Page Tracking	Spaces	Group Access	Form Promotio
Form Display Name * <input type="text" value="Demo Form"/> ?									
Form Code * <input type="text" value="demo-form"/> ?									
Ref Form ID <input type="text"/> ?									
Form Manager <input type="text" value=""/> ?									
Transaction Value <input type="text"/> ?									
Active <input checked="" type="checkbox"/> ?									
<u>Test Mode</u> <input checked="" type="checkbox"/> ?									
Form Version Selector <input type="text" value=""/> ?									
Submission Expiry Date <input type="text"/> ?									
Submission Expiry Days <input type="text"/> ?									
Tracking Code									
Use Receipt No. for Tracking Code <input type="checkbox"/> ?									
Tracking Number Service <input type="text" value=""/> ?									

Setting this will put your form in Test Mode. Once in Test Mode you can set the Test Delivery Channel as well.

Delivery Channels

Production Delivery	<input type="text" value=""/> ?
<u>Test Mode Delivery</u>	<input type="text" value=""/> ?
Validation Failure Delivery	<input type="text" value=""/> ?

A form in Test Mode can be clearly identified from the form dashboard by a red Yes in the form details pane, and in the forms page with a red dot in the Test Mode column.

Form Details

Form Display Name : Demo Form

Form Code : demo-form





Organization : Maguire

Test Mode : Yes

Delivery Channel: Trash Can Delivery

Created : 04 Nov 2015 - 13:46 by administrator

Last Modified : 10 Dec 2015 - 15:34 by administrator

Form Name	Org.	Form Code	Current Version	Active	<u>Test</u>	Delivery Channel	Last Modified	Action
Demo Form	maguire	demo-form	3.0	✓	●	Trash Can Delivery	10 Dec 2015 by administrator	 
Dual Demo Form	Macquarie	demo-form2	1.0	✓		Dual	07 Dec 2015 by administrator	 

Forms in Test Mode

Submissions made while a form is in Test Mode have most of the functionality of an active form, however the transactions are clearly marked as test submissions.

ID	Tracking Code	Form	Org.	Time	Space	User / Contact Email	Payment	Test	Transaction Status
450	NQT5SZ	Demo Form	maguire	10 Dec 15:38	Maguire			●	Delivery Ready
449	SSY52W	Demo Form	maguire	10 Dec 15:38	Maguire			●	Receipt Render Pending
446	ZHJTSS	Demo Form	maguire	10 Dec 14:59	Maguire				Delivery Completed
444	3XCQGN	Demo Form	maguire	10 Dec 14:51	Maguire				Delivery Completed

Note that a forms in test mode will not appear in the Space forms page, so must be rendered in TM. The form submission will also not appear in any of the reporting or analytics included in TM.

Groovy Scripting in Transact Manager

This guide provides an introduction into utilising groovy scripting to maximise the functionality of Transact Manager.

Handling Basic Form Objects

A form in TM is represented by an object in the JVM that can be accessed by groovy script. Using groovy script it is possible to access form data, and basic form properties.

How-to articles

[Add how-to article](#)

Title	Creator	Modified
Transaction Delivery	Stuart Sharpless	Jul 08, 2016
Transaction Reporting Feed	Stuart Sharpless	Mar 14, 2016
TField and Locations	Stuart Sharpless	Feb 26, 2016
Collaboration Jobs - Rolling Back To a Previous Step (State)	Larry Bunton	Feb 19, 2016
test	Unknown User (bwarner)	Jan 05, 2016
Form Details in Composer	Joshua Miller	Nov 12, 2015
Form Schema Design and Composer	Larry Bunton	Oct 29, 2015

Keeping track of Composer Forms in TM

Transact Manager allows for each form to have multiple versions for forms published from composer. This allows user to track changes and provides the facility to roll back to earlier versions if something goes wrong. A common issue users experience when publishing updated forms to TM is forms failing to update existing versions or create new versions against an existing form entry. The usual reason for this is that a different form is being updated and published in composer than the original one that was first published to TM.

How to find the location of the form associated with a particular version in TM.

- Login to TM and navigate to the form dashboard
- Click **Properties** on the form version which you want to update or create a new version of

The screenshot shows the 'Form Dashboard' for a form named 'Demo Form'. The dashboard includes several sections:

- Form Details:**
 - Form Display Name: Demo Form
 - Form Code: demo-form
 - Organization: Maguire
 - Delivery Channel: Trash Can Delivery
 - Created: 04 Nov 2015 - 13:46 by administrator
 - Last Modified: 22 Dec 2015 - 16:08 by administrator
- Form Versions:**

Version	Current Version	Last Modified	Properties	Attachments	Services	Data Config
4.0	✓	22 Dec 2015	Properties	Attachments	Services	Data Config
3.0	Make Current	22 Dec 2015	Properties	Attachments	Services	Data Config
2.0	Make Current	07 Dec 2015	Properties	Attachments	Services	Data Config
1.0	Make Current	04 Nov 2015	Properties	Attachments	Services	Data Config
- Form URLs:**

Spaces	Friendly	Landing	Form	Direct	QR Code
Acme					
Maguire					
TransactField App					
Web Plug-in					
PDF Receipt Test					
- Latest Transactions:**

ID	Receipt Number	Time	Transaction Status	Receipt
532	demo-form-35	04 Jan 16 15:27	Delivery Completed	
446	demo-form-25	10 Dec 15 14:59	Delivery Completed	
444	demo-form-24	10 Dec 15 14:51	Delivery Completed	
443	demo-form-23	10 Dec 15 14:50	Delivery Completed	
349	demo-form-22	01 Dec 15 11:50	Delivery Completed	

Submissions: 12 Requests: 32 Submission Rate: 37% Avg. Submit Time: 1 min

- In the properties page click the **Form Archive Info** tab

The screenshot shows the 'Form Archive Info' tab in the properties page. It displays the following information:

- Author: jmiller@avoka.com
- Organization: joshm
- Form Name: Demo Form
- Form Key: Avoka AUS/joshm/Test/Demo Form/html-receipt
- Form Description: Form for recreating images in articles
- Form Generation Date: 22 Dec 2015 - 4:07 PM
- Form Archive MD5: c097be1cc97d8ec903b0b21faf0adc56
- Build Server URL: https://transact.composer.avoka.com.au/composer
- Data Version: 4.2.0sp2
- Build Version: 4.3
- Build Number: 44742
- Build Environment: cloud
- HTML Desktop Template:
- HTML Receipt Template:
- HTML Tablet Template:
- PDF Form Template:
- PDF Receipt Template:
- Supports Save Challenge:

This tab gives an overview of the form information regarding composer, including who created the form, the organisation and where in composer the form was published from. This information can be used to diagnose issues with the form when trying to publish from composer.

Managing Notifications

The following links will take you to the notifications blogs for the various streams of notifications you can select to receive notifications from. To select to receive email notifications, follow the relevant link and the opt to watch the blog in the top right of the page.

[Transact Product Notifications](#)

[Transact Field Notifications](#)

[Transact Service Maintenance](#)

[General Notices](#)

Error rendering macro 'blog-posts'

'TFN' is not an existing space's key

Meeting notes

[Create meeting note](#)

Incomplete tasks from meetings

Task report

Looking good, no incomplete tasks.

All meeting notes

Title	Creator	Modified
2015-02-11 AvokaDoc Structure Review Meeting	Unknown User (bwarner)	Feb 11, 2015

Multi-Step Groovy Delivery Process

This guide provides an introduction and how-to for creating multi-step groovy delivery processes.

Transact Manager already provides the capacity for multi-step delivery processes through the use of delivery check points in the Groovy Delivery Process. However for larger or more complicated delivery methods it may be easier to separate delivery steps into individual delivery processes. This has the added advantage of allowing forms to call only the delivery method they need. For forms that require these multi stage delivery methods it is possible to configure a groovy delivery process to call other delivery processes. This allows form developers to create a type of landing point for form submissions that can branch off to any number of different delivery methods, providing a wide range of functionality for accepting form submissions.

This guide will provide a guide for creating a dual delivery process that utilise two separate existing delivery processes in TM.

Basic Groovy Delivery Process

First lets make a basic Groovy Delivery Process. In TM go to the service definitions page and click new and fill out the new service information.

Create a new Service based on a Service Template.

Service Type

Service Template *

Service Name *

Organization

The basic groovy delivery process will have the following code provided for you.

```
1  /* Provides a Groovy script based submission delivery process.
2
3  Script parameters include:
4  submission : com.avoka.fc.core.entity.Submission
5  submissionXml : String
6  receiptPDF : byte[]
7  attachmentsMap : Map<String, com.avoka.fc.form.service.DataDocument>
8  deliveryCheckpoint : com.avoka.fc.core.service.submission.DeliveryCheckpointService
9  serviceDefinition : com.avoka.fc.core.entity.ServiceDefinition
10 serviceParameters : Map<String, String>
11 deliveryDetails : com.avoka.fc.core.entity.DeliveryDetails
12 formDataMap : Map<String, String>
13
14 Script return:
15 null or
16 deliveryResult : com.avoka.fc.core.service.DeliveryResult
17 */
18
19 if (deliveryCheckpoint.doCheckpoint("Checkpoint A", "Description...")) {
20     try {
21         // TODO: provide your custom delivery code
22         // ..
23
24         deliveryCheckpoint.completedCheckpoint("Checkpoint A")
25     } catch (Throwable e) {
26         deliveryCheckpoint.errorCheckpoint("Checkpoint A", e)
27     }
28 }
29
30
31 if (deliveryCheckpoint.doCheckpoint("Checkpoint B", "Description...")) {
32     try {
33         // TODO: provide your custom delivery code
34         // ..
35
36         deliveryCheckpoint.completedCheckpoint("Checkpoint B")
37     } catch (Throwable e) {
38         deliveryCheckpoint.errorCheckpoint("Checkpoint B", e)
39     }
40 }
41 }
```

When this process is called, after a form submission is made, it will first run the code in the try/catch block labeled Checkpoint A and then the code in the next try/catch block labeled Checkpoint B. If the code in the try catch block should fail it will retry this code again in 15 mins.

 The delivery retry delay and the number of retry attempts can be adjusted in the service parameters of the delivery process, the fields are retryDelayMins and maxDeliveryAttempts.

In this code even if Checkpoint A should fail Checkpoint B will still make an attempt to be delivered, this is known as a parallel delivery method. You can also setup the delivery process to be serial, this is useful if you have delivery processes that rely on each other. The groovy code for a serial delivery looks like this.

```

1  |/* Provides a Groovy script based submission delivery process.
2
3  |   Script parameters include:
4  |       submission : com.avoka.fc.core.entity.Submission
5  |       submissionXml : String
6  |       receiptPDF : byte[]
7  |       attachmentsMap : Map<String, com.avoka.fc.form.service.DataDocument>
8  |       deliveryCheckpoint : com.avoka.fc.core.service.submission.DeliveryCheckpointService
9  |       serviceDefinition : com.avoka.fc.core.entity.ServiceDefinition
10 |       serviceParameters : Map<String, String>
11 |       deliveryDetails : com.avoka.fc.core.entity.DeliveryDetails
12 |       formDataMap : Map<String, String>
13
14 |   Script return:
15 |       null or
16 |       deliveryResult : com.avoka.fc.core.service.DeliveryResult
17 | */
18
19 | if (deliveryCheckpoint.doCheckpoint("Checkpoint A", "Description...")) {
20 |     try {
21 |         // TODO: provide your custom delivery code
22 |         // ..
23
24 |         deliveryCheckpoint.completedCheckpoint("Checkpoint A")
25 |     } catch (Throwable e) {
26 |         deliveryCheckpoint.errorCheckpoint("Checkpoint A", e)
27 |     }
28 | }
29 |
30 |
31 | if (deliveryCheckpoint.isCompleted("Checkpoint A") && deliveryCheckpoint.doCheckpoint("Checkpoint B", "Descrip
32 |     try {
33 |         // TODO: provide your custom delivery code
34 |         // ..
35
36 |         deliveryCheckpoint.completedCheckpoint("Checkpoint B")
37 |     } catch (Throwable e) {
38 |         deliveryCheckpoint.errorCheckpoint("Checkpoint B", e)
39 |     }
40 | }
41 | }

```

This code won't call the the second checkpoint until the first has been successfully completed.

You can find additional documentation for these standard delivery processes in the Avoka Transact Services Guide.

Dual Delivery

It is possible, using groovy script, to call another process from within your delivery process.

Delivery Service Call

```

IDeliveryProcessService deliveryService = (IDeliveryProcessService) ServiceLocator.getServiceForName("[SERVICE_NAME])
deliveryService.deliverSubmission(submission, new DeliveryDetails())

```

These two lines of code first locate the service by its name, then calls the service by passing the submission object and delivery details to the service. Using this code in a groovy delivery service it is possible to create a lanning point for submissions that can be delivered in a variety of methods based on any number of factors, such as submission data or other successful delivery methods.

An example case:

A form is submitted that calls a basic delivery process 'logger', which logs the form submission data for review by a backend system. The form is updated to include an email delivery function only if a certain email is input into the form. To get this functionality to work a delivery process is created to make the initial 'logger' delivery and also check for the email and complete that delivery process if it is found.

Dual Delivery

```

import com.avoka.fc.core.entity.DeliveryDetails
import com.avoka.fc.core.entity.Submission
import com.avoka.fc.core.dao.SubmissionDataDao
import com.avoka.fc.core.service.IDeliveryProcessService
import com.avoka.fc.core.service.ServiceLocator
import com.avoka.fc.core.service.submission.DeliveryCheckpointService
import com.avoka.fc.forms.api.DeliveryException

import java.util.List

if (deliveryCheckpoint.doCheckpoint("Checkpoint A", "Description...")) {

    try {
        // Call first delivery process
        IDeliveryProcessService deliveryService = (IDeliveryProcessService) ServiceLocator.getServiceForName("Logger Delivery")
        deliveryService.deliverSubmission(submission, new DeliveryDetails())
        deliveryCheckpoint.completedCheckpoint("Checkpoint A")
    }
}

```

```

    } catch (Throwable e) {
        deliveryCheckpoint.errorCheckpoint("Checkpoint A", e)
    }
}

if (deliveryCheckpoint.doCheckpoint("Checkpoint B", "Description...")) {
    try {
        def list = submission.getSubmissionExtractData();
        //Iterate through Submission Data
        for(def temp in list){
            //Check for Email Field with specific Email value
            if(temp.getName() == "email" && temp.getValue() == "BackEnd@domain.com"){
                //Call Second Delivery process
                IDeliveryProcessService deliveryService = (IDeliveryProcessService) ServiceLocator.getServiceForName('Email Delivery')
                deliveryService.deliverSubmission(submission, new DeliveryDetails())
            }
        }
        deliveryCheckpoint.completedCheckpoint("Checkpoint B")
    } catch (Throwable e) {
        deliveryCheckpoint.errorCheckpoint("Checkpoint B", e)
    }
}
}

```

The code for this is relatively simple and can be used to call any number of delivery processes.

Overview of Search Functionality in Space Pages

This article contains an overview of how the search functions for each of the Space Forms, Tasks, and History pages.

Forms

The forms page lists all the available forms for the logged in space that the users has access to. The search functionality on this page has two fields, keyword search and category filter.

Keyword Search takes a user entered keyword and runs a query against the database that returns a list of forms. The key word is matched against a number of form related attributes.

List of attributes

The category filter filters out forms based on category. These categories are set against form version in TM. A form can have multiple categories or none. The form list is only filter after the search button is clicked.

Form Version Properties Attachment Rules Services **Form Categories** Form Tags Form Archive Info

Select the set of categories applicable to this form version. The set of categories is defined on the organization level.

Available Categories

- FOREIGN EXCHANGE
- SUPERANNUATION
- TEST FORMS

Assigned Categories

- CREDIT CARDS
- SALES

Categories

> < >> <<

Save Close

Tasks

The task page lists all the available task assigned to the logged in user. This list of tasks can be filter by three fields, keyword search, status filter, and group items filter.

The keyword search takes a user entered search term and matches it against the task receipt number.

This task list can be filtered by the status of the task, assigned, saved, or submitted. The list can also be set to include task assigned to groups as well as the individual user by checking the group items check box.

History

Search

Last Filter(days)

Group Items

Product requirements

[Add Product requirements](#)

Title	Designer	Developers	Document owner	Document status	Epic	QA	Target release
Product Requirement - Standard			Unknown User (bwarner)	DRAFT			

Reset forgotten password for Avoka support portal

For users who have misplaced or forgotten their credentials used to access the Avoka Transact support portal, there is a reset/recover function available.

When on the login page for the support portal, click the [Can't access your account?](#) link.

Welcome to Avoka Support

Username

Password

Remember my login on this computer

Need an account? Send a request to [Avoka Support](#).

[Can't access your account?](#)

Powered by Atlassian JIRA Project Management Software (v6.4.9#64024-sha1:1f1084e)

User can either request a password reset, or recover an account username. To reset the password you will need to enter you username. To recover the account user name you will need to enter the email address associated with that account.

Can't access your account?

i If you can't access JIRA, fill in this form and an email will be sent to you with the details to access your account again.

Which did you forget Password
 Username

Enter your *

username

Powered by Atlassian JIRA Project Management Software (v6.4.9#64024-sha1:1f1084e)

Password reset

Reset password link sent successfully

✓ A reset password link has been sent to you via email.
You can follow that link and select a new password.
If the email does not arrive, please contact your JIRA administrators.

Powered by Atlassian JIRA Project Management Software (v6.4.9#64024-sha1:1f1084e)

Once you have submitted the password reset request you will receive an email (sent to the email address registered for the given account).

Password reset request

A request has been received to reset your password.


If you follow the link below you will be able to personally reset your password.
https://support.avoka.com/tickets/secure/ResetPassword/default.jspx?os_username=Test&token=08c8be606302d2ca62364bd8036c300344325026

This password reset request is valid for the **next 24 hours**.
Don't worry you can always ask for a new password using the following link:
<https://support.avoka.com/tickets/secure/ForgotLoginDetails.jspx?username=Test>

Here are the details of your account:

Username: Test
Email: [REDACTED]
Full Name: Test

This message was sent by Atlassian JIRA (v6.4.9#64024-sha1:1f1084e)



The email contains two links, the first leads to a password reset page where you can enter in a new password. This will overwrite the old password with the new one. The other leads back to the forgotten login details page.

Note: The password reset link is only valid for 24 hours, after this period a new reset email will need to be requested.

Reset Password

Username

New Password

Confirm

Powered by Atlassian JIRA Project Management Software (v6.4.9#64024-sha1:1f1084e)

Once you have reset your password you will be able to login to with the new credentials.

Reset Password

Your password has been reset. You can now [log in](#).

Powered by Atlassian JIRA Project Management Software (v6.4.9#64024-sha1:1f1084e)

Username Recovery

Users who have forgotten their login username can request it be sent to them. This requires that you know the email registered for the account you are trying to recover.

Username sent successfully



Your username has been sent to you via email.

You can then request a new password with that username [here](#).

If the email does not arrive, please contact your JIRA administrators.

Powered by Atlassian JIRA Project Management Software (v6.4.9#64024-sha1:1f1084e)

The email will contain the login Usernames for any accounts that have that email address. You can then use the username to login in to the portal or request a password reset, as above.

Username(s) reminder request

You (or someone else) have requested your username(s).

Your username(s) are:

• • •

• • • **Test**

You can request a new password for the username(s) here:

<https://support.avoka.com/tickets/secure/ForgotLoginDetails.jspa>

This message was sent by Atlassian JIRA (v6.4.9#64024-sha1:1f1084e)



Resubmitting a Previously Submitted Transaction

This article details the steps for resubmitting a previously submitted transaction. This method can be used to retrigger submission processor or backend delivery services, or reproduce receipts for cases where the previous submission had an issue when rendering a receipt.

Resubmitting XML from previous transaction

1. Open the transaction record in TM that you wish to resubmit.
2. Open the **Form XML Data** tab and copy the XML. Save this copied XML to a file as you will need it later.

Transaction Details

Home Dashboard > Form Transactions > Transaction Details

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <AvokaSmartForm>
3   <SFMDData>
4     <SystemProfile>
5       <DisplayMode>Receipt</DisplayMode>
6       <HostContext>Page</HostContext>
7       <ReceiptNumber>simple-form-25</ReceiptNumber>
8       <SubmitDateString>08/03/2016 09:57:05</SubmitDateString>
9       <SubmissionMessage/>
10      <FormDataServiceURL>https://localhost/maguire/servlet/FormDynamicDataService</FormDataServiceURL>
11      <RequestLogKey>2a2600894e6df7da163ed649e072e50c</RequestLogKey>
12      <TrackingCode>QBW57R</TrackingCode>
13    </SFMDData>
14    <Job>
15      <ReferenceNumber/>
16      <StepName/>
17      <AvailableRoutes/>
18      <RouteName/>
19      <Assignes/>
20      <AssignRepeatIndex/>
21      <AssignRepeatItem/>
22    </Job>
23    <AutoSaveDisabled/>
24    <FormCode>simple-form</FormCode>
25    <ComposerDataVersion>4.2.0sp2</ComposerDataVersion>
26    <UserName/>
27    <SubmissionType>Submitted</SubmissionType>
28    <OnlineSaveEnabledFlag>true</OnlineSaveEnabledFlag>
29    <LocalSaveEnabledFlag>true</LocalSaveEnabledFlag>
30    <TransactionScore/>
31    <TemplateVersionNumber>2.0</TemplateVersionNumber>
32    <Services>
33      <AddressLookup>0</AddressLookup>
34    </Services>
35    <ShareForm>
36      <EmailAddresses/>
37      <SenderName/>
38      <EmailBody/>
39    </ShareForm>
40    <ModalPage/>
41    <SubmissionExpiryDate/>
42    <RevisionNumber>1</RevisionNumber>
43    <ExternalAnalyticsID/>
44    <ServerBuildNumber>4.2.5</ServerBuildNumber>
45    <Referer>https%3A%2F%2Flocalhost%2Fmanager%2Fadmin%2Fform%2Fform-edit.htm%3FtabPanelIndex%3D0%26entityId%3D23%26wid%3Dfa58</Referer>
46    <UpdateUserProfile>false</UpdateUserProfile>
47    <TestMode>false</TestMode>
48    <SubmissionNumber>756</SubmissionNumber>
49    <OfflineSubmissionId>50ac51244a058bb48c7a924cc30e0f71</OfflineSubmissionId>
50  </SystemProfile>
51  <PaymentDetails>
52    <Payment>
53      <SubTotal/>
54      <GST/>
55      <ServiceFee/>
56      <Total/>
57    </Payment>
58    <PaymentType/>
59    <AccountNumber/>
60    <ItemList>

```

3. Navigate to the **Assigned Tasks** page.
4. Create a new assigned task by clicking the **New** button.

5. Fill out the task details, type, subject, user login name, form name, and upload the xml that you obtained in step 1 then click **save**.

i Ensure that the correct **User Login Name** is filled in for the user who will be resubmitting the form as this is who the task will be assigned to. Also make sure the Form name matches the form the XML is from.

Edit Assigned Task

Home Dashboard > Assigned Tasks > Assigned Task

Task Type* Form

Task Subject*

Task Message

Previous Submission ID

Scheduled Date

Expiry Date

Task Address

Location Latitude

Location Longitude

Allow Claiming

User Login Name

Group

Email Notice
Job Applicants
Job Managers
Job Reviewers

Task Deletable by User

Send Email to Assignees

Form Name Simple Form

Form Data Form XML Data

Upload Form XML* No file selected.

6. The newly created task should now be visible from the **Assigned Tasks** page in TM.

Assigned Tasks

Home Dashboard > Assigned Tasks

i The Submission has been successfully saved.

search Type Status Assigned Start Date 01 Mar 2016 00:00 End Date

ID	Tracking Code	Form Code	Task Type	Subject	Status	User	Group	Created	Scheduled	Completed	Expiry	Expiry Status	A
757	VNQCTP	simple-form	Form	Form For Resubmission	Task Form Assigned			08 Mar 2016					

< >

7. The user can either access the task from within TM or they can login to the space associated with the form, from here the user should see the task listed on their tasks page.



Home Forms **Tasks (2)** History Reviews

Task List

Complete your outstanding forms and tasks.

Filter:
 Group Items:

▶ **VNQCTP - Form For Resubmission**

Assigned Task

Tracking Code: **VNQCTP**

Assigned To: ..

Created: **8 Mar 2016 10:08 AM**

8. Open the task. The form will be rendered with all the data from the previous submission prefilled into the form. Simply navigate through to the final page and click the **Submit** button to finish the submission.



Need Help ?

Reference Code: **VNQCTP**

Getting Started Section 2 Layout Signature

Save For Later Cancel / Exit

Getting Started

Simple Form

*Fields marked with * are required*

About You

Section help goes here.
 Utilising the inbuilt help on level 2 sections to give some context around the section is an effective form design principle.

Title First Name * Family Name

Address Line 1

Address Line 2

Suburb State Postcode

Extract Me Cost Age













9. The submission will now appear in TM. Navigate to the **Form Transactions** page and find the transaction. Here you can render the receipt for the new submission and check the transaction details as well.

Form Transactions

Home Dashboard > Form Transactions

search Form Status Start Date

Attachments Payments Email Verification Delivery End Date

ID	Tracking Code	Form	Org.	Time	Space	User / Contact Email	Payment	Test	Transaction Status	Action
757	VNQCTP	Simple Form	maguire	08 Mar 10:15	Maguire				Delivery Completed	   
756	QBW57R	Simple Form	maguire	08 Mar 09:57	Maguire				Delivery Completed	   
754	PDBB7X	Demo Form	maguire	07 Mar 13:53	Acme				Form Saved	   

Save Challenge Behaviour

Article for outlining the observed behaviour for save challenge

Save/Resume Single User

Scenario - User renders new form, with save/resume feature enable and save challenge configured. User then saves form.

Path 1 - User re-opens form from save confirmation page link. User is prompted to answer save challenge.

Url : <https://tm.staging.avoka.com/maguire/servlet/SmartForm.html?formCode=save-challenge-test&saveChallenge=true&referenceNumber=GZZJ42>

Path 2 - User returns to form with bookmark, without closing browser so session is maintained. User is allowed back into form without save challenge.

Url : <https://tm.staging.avoka.com/maguire/servlet/SmartForm.html?submitKey=542dfa64b9e153f57be416eda566abbd>

Path 3 - User Emails reminder link to email. Opens link in email without closing browser so session is maintained. User is prompted to answer save challenge.

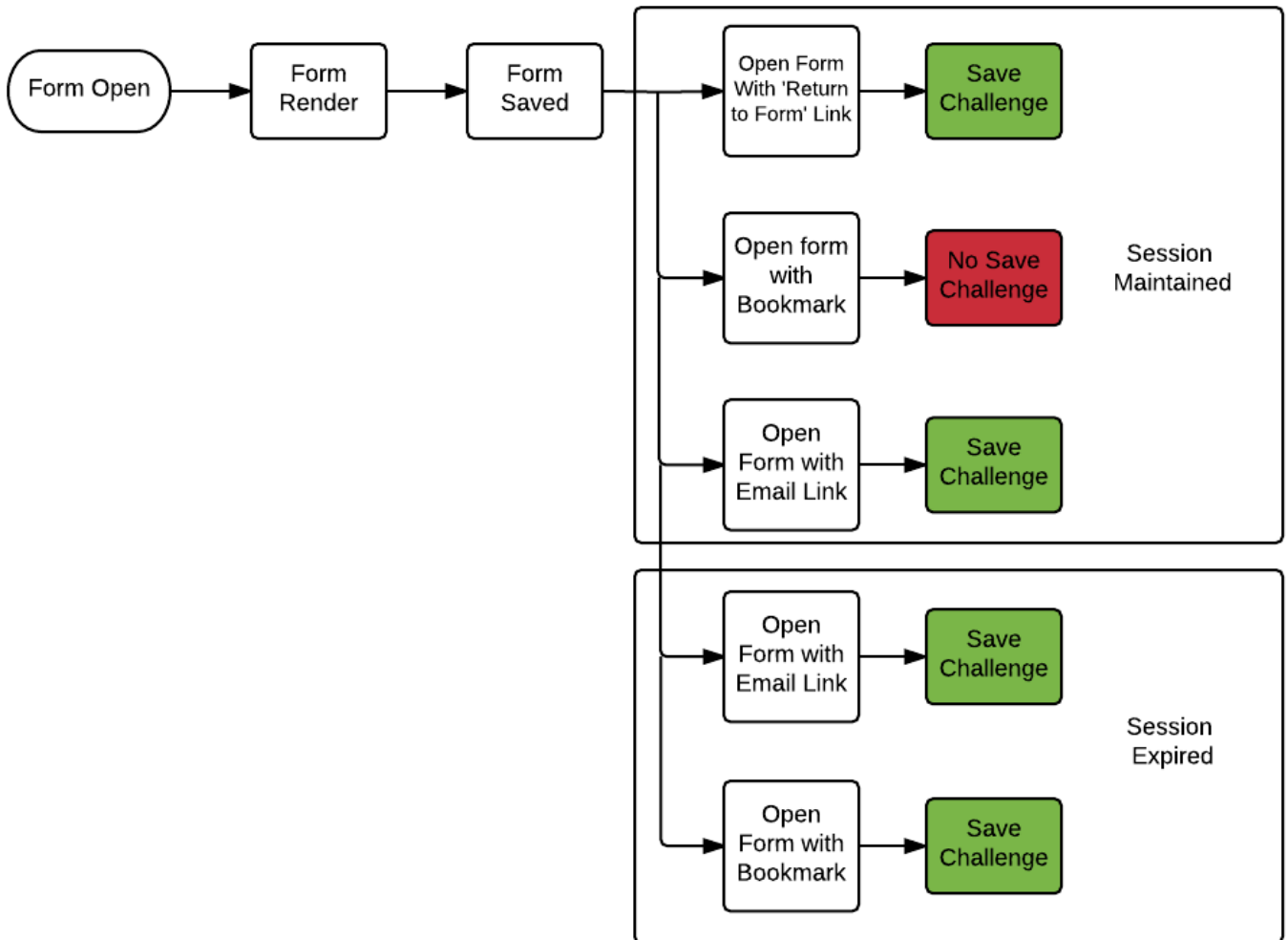
Url : <https://tm.staging.avoka.com/maguire/servlet/SmartForm.html?formCode=save-challenge-test&saveChallenge=true&referenceNumber=GZZJ42>

Path 4 - User Closes browsers and allows session to expire. User returns to form using emailed save url. User is prompted to answer save challenge.

Url : <https://tm.staging.avoka.com/maguire/servlet/SmartForm.html?formCode=save-challenge-test&saveChallenge=true&referenceNumber=GZZJ42>

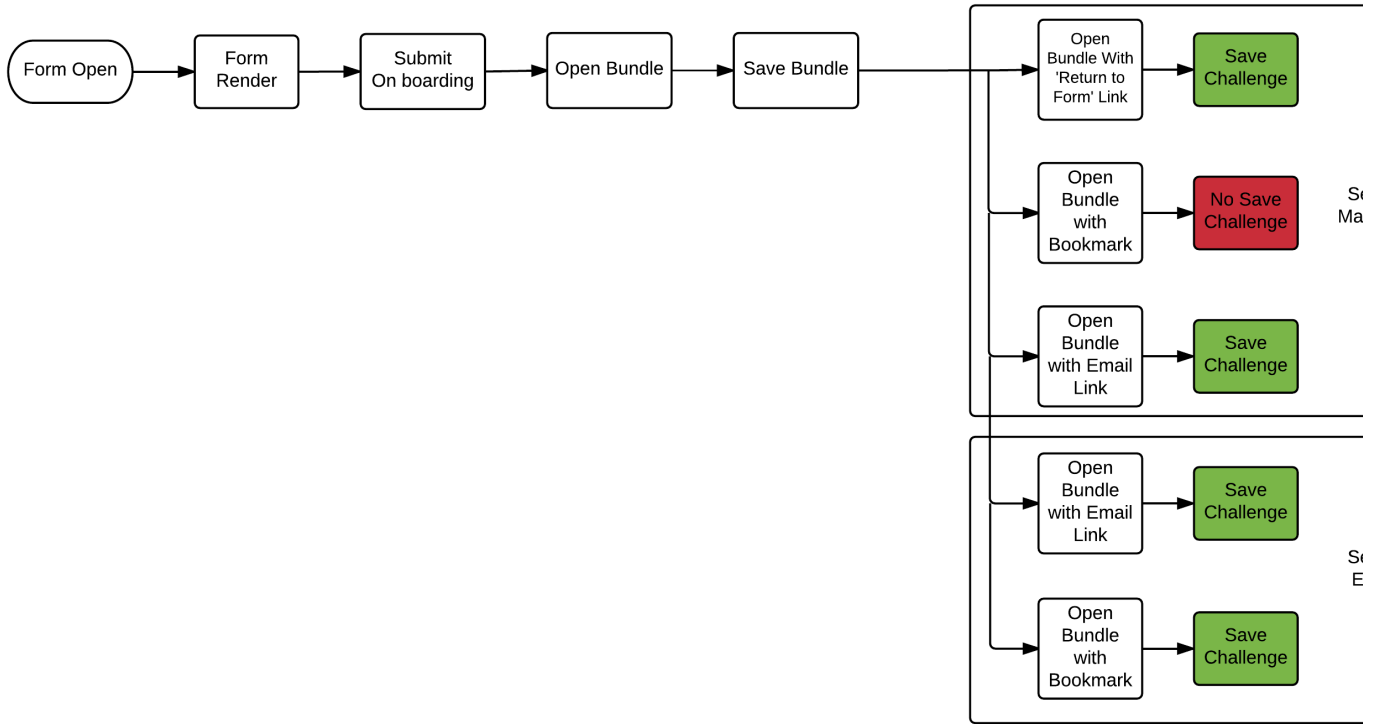
Path 5 - User Closes browsers and allows session to expire. User returns to form using bookmarked link. User is prompted to answer save challenge.

Url : <https://tm.staging.avoka.com/maguire/servlet/SmartForm.html?submitKey=542dfa64b9e153f57be416eda566abbd>



Form Bundle - Single User

Scenario - User renders a basic form bundle, with save/resume feature enable. User submits initial bundle form, then opens main bundle forms. User bookmarks form and then saves form.



When a user navigates between forms in the bundle using the bundle navigator it uses the formCode and SubmitKey. While the session is active this will not trigger a savechallenge. If the user shares the form or saves the form it uses the formCode/saveChallenge/trackingNum this will trigger a save challenge regardless of session active.

Shared form - Two users

Scenario: User 1 renders form with share feature enable, then shares form with User 2

test

Step-by-step guide

[Link Test](#)



Related articles

- [Transaction Delivery](#)
- [Transaction Reporting Feed](#)
- [TField and Locations](#)
- [Collaboration Jobs - Rolling Back To a Previous Step \(State\)](#)
- [test](#)

TField and Locations

TField has the ability to capture the devices location and embed this information into the users transaction. This article described how this location is obtained.

Technology

TField uses the Cordova geolocation plugin to get the position which in itself uses native API's to get the location.

The following is the link to the details about the Cordova geolocation plugin

<https://github.com/apache/cordova-plugin-geolocation>

Known Behaviour

1 - Currently our integration does not enforce a high accuracy location. This means that which ever services is available to the device at the time will be available.

For Example, If the GPS does not have a location at the time then Wifi or Cell may be used for less accurate locations.

2 - Currently the system will return the last know location not necessarily the current location.

3 - Some devices may prioritise Wifi over GPS.

Device Specific Information

Windows

iOS

Android

TM - Security Audit Logs (user-auth.log)

Transaction Manager maintains a file based security audit log for archive purposes.

It is usually a requirement for security audits to keep these files for audit purposes for 7 years.

What does the log try to achieve?

The aim of this log is to provide an audit trail of user activity that can provide access to users PII data. Each entry in the log included a time stamp of when the event happened.

What is logged?

User Authentication to the system console

- Users failed logins attempts

- User logouts

- User session expiry

- Successful user logging

- User password reset requests

Access to Customer Data

Through the Manager Console:

- Submission history XML view

- View submission XML

- Update submission XML

- Viewing of attachments

- Viewing of receipts

Through the Spaces:

- Viewing of attachments

- Viewing of receipts

Samples Entries...

Transaction Delivery

Generally, Once a user has completed all the necessary parts to their transaction (Data, Attachments & Payments) the transaction is scheduled for delivery and this is the stage when the transaction is handed to an area of the business for processing. Once a transaction has been delivered is it viewed as completed and will then become a candidate for purging at some time in the future

There are a number of different ways to delivery transactions. This article only attempts to give a top level overview of different type of delivery that are possible and also descres what the different delivery statuses mean and how that are used by Transaction Manager.

This article will be relevant to operations staff who are monitoring transaction flow and also developers who are developing custom delivery processes and who may need to implement some of these transitions.

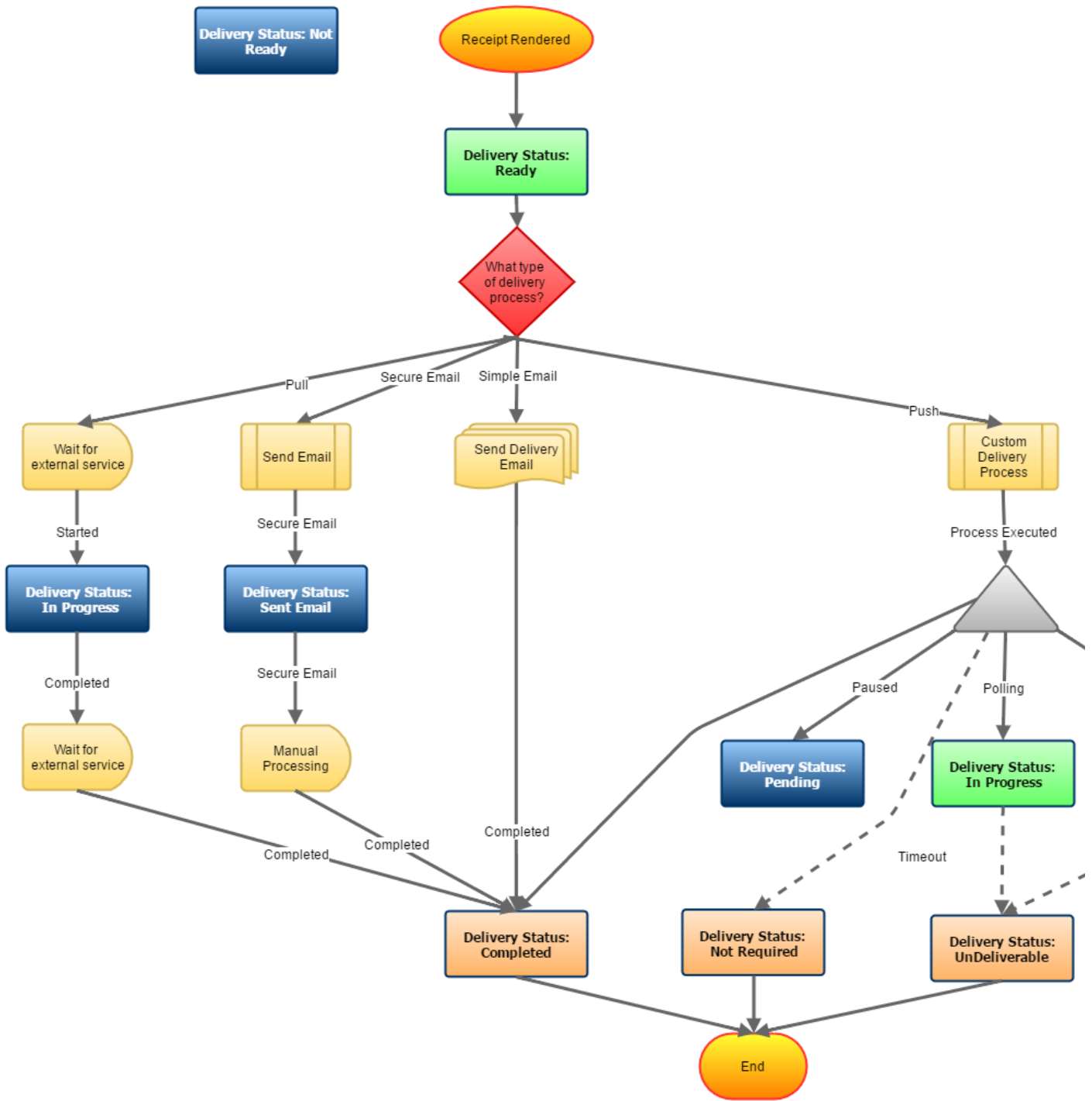
Transaction Manager delivery options and delivery statuses.

Overview of the supported Transaction Manger delivery types

Name	Type	Description
Email	Push	An email is sent to an email address which contains the contents of the transaction. Least secure option as the transaction data is sent over email which is plain text.
Secure Email	Push	An email is sent to an email address to notify of a transaction waiting for delivery. User needs to follow a link to login and download the transaction details for processing. Secure as download of the transction details is done over HTTPS.
Custom Delivery Process	Push	A custom delivery process can be written using Groovy Script to customise the delivery process.
Web Service	Pull	External systems can poll using Webservice API to download transactions that are waiting for delivery
Rest	Pull	External systems can poll using Rest API to download transactions that are waiting for delivery

You may be asking what is the significance of Push or Pull type of delivery. Push delivery is initiated by Transaction Manager and is handled by the Transaction Processor, which is scheduled job that runs at regular intervals and starts the push delivery processes. Pull delivery involves an external service polling TM to check if there are transaciton that need to be delivered. In the case of a pull delivery transcation manager just waits to receive a call from the external system and hence does not actively attempt to delivery the transactions.

Transaction Manager Delivery Statuses



State	Type	Description
Not Ready	Not Ready	The transaction is not ready for delivery yet. This is the start state before a transaction has been completed by the user. Generally the receipt render service will kick the transaction from this state to the Ready state when the receipt is rendered
Ready	Waiting	The transaction is ready for delivery by either the Transaction processor, Rest or Web Service delivery methods.
Sent Email	Paused	Secure Email has been sent. System is waiting for the user to login and process the transaction
In Progress	Waiting*2	Delivery has been started but is waiting to be completed
Pending	Paused	The transaction delivery is paused. The transaction will be pushed out of this status by some other means (Groovy Scheduled Task)
Error*1	Waiting*2	Transaction Delivery has failed and is waiting to be retried.
Undeliverable	End	Delivery as not possible, Check the error logs
Not Required	End	Delivery is not required for this transaction

(*1) - When does TM put a transaction into Delivery Error status?

- 1 - The Delivery Process returns a DeliveryResult object with the status of Error.
- 2 - After the Delivery Process has executed and there are new checkpoint error entries added during the last execution
- 3 - The Delivery Process throws an exception

In this case the underlying database transaction is rolled back
Groovy Logger output down to debug level is stored against the Groovy Service Log entry
A TM ErrorLog entry is added with the message from the exception

(*2) - Controlling the retry of In Progress and Error transactions

Each delivery services has a parameter to control the maximum number of retry attempts. This value is controlled through parameters on the delivery service. After this an administrator will need to fix the transaction and attempt delivery again.

Next retry times can be controlled by custom delivery processes by returning a DeliveryResult object. See the JavaDoc for more information.

Waiting States (Green)

Transactions with Delivery Statues of In Progress, Error and Ready fall into the scope of transaction and will be processed as part of the Transaction Processor scheduled job.

End States ()

Transaction which are in one of the end states will become elegable for purging.

Related articles

- [Multi-Step Groovy Delivery Process](#)

Transaction Reporting Feed

Communication

Transaction data is stored in a Amazon S3 Bucket. The data is encrypted on S3 and the account used to post the data is read-only and can not be used to recover data from the bucket.

Connection to S3 is initiated from a Transact Manager environment and is over HTTPS. The following is the URL that needs to be enabled (port 443) for outgoing access from the TM environment to enable this feed.

<https://s3-ap-southeast-2.amazonaws.com/>

<https://avoka-transactions.s3-ap-southeast-2.amazonaws.com/> (For transparent proxy, for example Sophos UTM, both URL will need to be added to allow list)

What does it contain

Primarily the data feed contains the data that is contained in the 'Transaction Log' and 'Org Transaction Log' view under the 'Reports' Menu in Transact Manager.

Some information about the TM installation is also included, E.g. Database Type, OS Type, Manager URL, Customer Name etc.

This feed does not contain information about individual transactions and hence does not contain personal data. The feed is based on daily transaction auditing statistics.

What is it used for

The data feed has two purposes:

1 - Transaction Licensing Auditing/Monitoring.

Part of the customer agreement with Avoka is to provide information to allow Avoka to monitor usage. Historically this was performed by BIRT reports that were automatically executed monthly and these are based on the same Transaction Log records. This data feed is a replacement for these report.

Examples of items that are monitored here are:

Transaction usage against transaction pack purchased

Address Lookup usage against address lookup packs purchased.

2 - Operation Monitoring and Support.

This feed also provides information to enhance our ability to support our clients. The information about transaction usage and features used by customers help us understand the usage of the system when we are responding to customer support tickets and responding to questions. Tracking the TM versions, Database and Operating System types used helps Avoka tailor support to customers systems. This is particularly relevant to on-premise installations where we do not manage the environment and are not always aware of the timings when upgrades are performed.

Sample File

The following is a sample daily data file as an example.

[SampleDataFile.json](#)

Transact Submission Flow and Timings

This article will outline the flow and associated timing of a transaction in TM

- Form Request - When a new form is request a record is created in TM associated with the transaction. The record will be generate a Submission ID, Tracking Code and Submission Key.
- Form Opened - Once the form is rendered the transaction will be in the Opened Status. The Form opened and last user activity timestamps are added to the transaction record.
- Formed Saved - If a user choose to save and close a form the transaction will move to the saved status. The transaction record will be updated with saved timestamp and the last user activity will be updated.
- Formed Submitted - Once a user submits the form the transaction record will be updated with the form submission timestamp.
- Form Completed - Only once a form has all it's required attachments and payments will the transaction move to a completed state.

Timestamp	Description
Form Opened	Time the form was first rendered
Form Saved	Time a form is saved and closed without being submitted
Form Submitted	Time form is submitted, when user clicks submit button
Form Completed	Time form is completed, includes attachment and payments

Differences with tfield

Troubleshooting articles

[Add troubleshooting article](#)

Title	Creator	Modified
Adobe Reader PDF form help	Joshua Miller	Oct 29, 2015
Viewing PDF documents in IE 10	Joshua Miller	Oct 29, 2015
Viewing PDF documents in Firefox	Joshua Miller	Oct 29, 2015
Viewing PDF documents in Windows 8	Joshua Miller	Oct 29, 2015
Viewing PDF documents in Safari	Joshua Miller	Oct 29, 2015
Viewing PDF documents in Chrome	Joshua Miller	Oct 29, 2015
Viewing PDF documents in IE 8 and IE 9	Joshua Miller	Oct 29, 2015
Troubleshooting Article - Standard	Unknown User (bwarner)	Feb 11, 2015

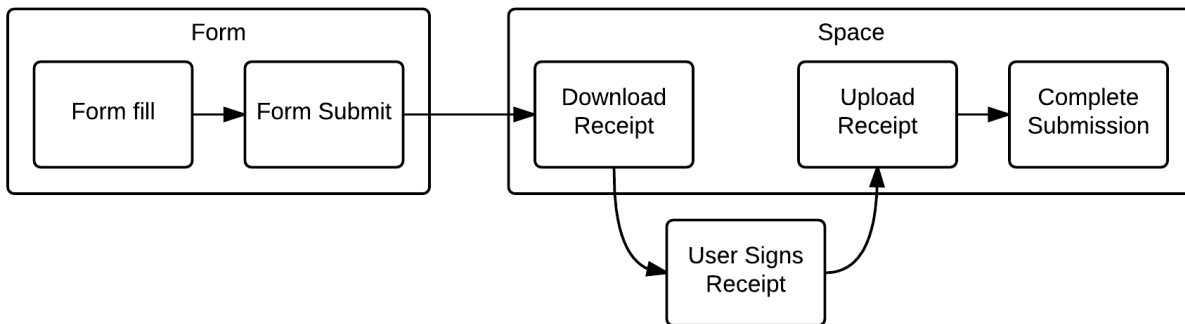
Wet Signatures in FTX Forms - DO NOT USE

DO NOT USE

This article will outline how to implement Wet Signatures in FTX forms. Wet signatures in Transact allow for the capture of real "wet" signature on a piece of paper.

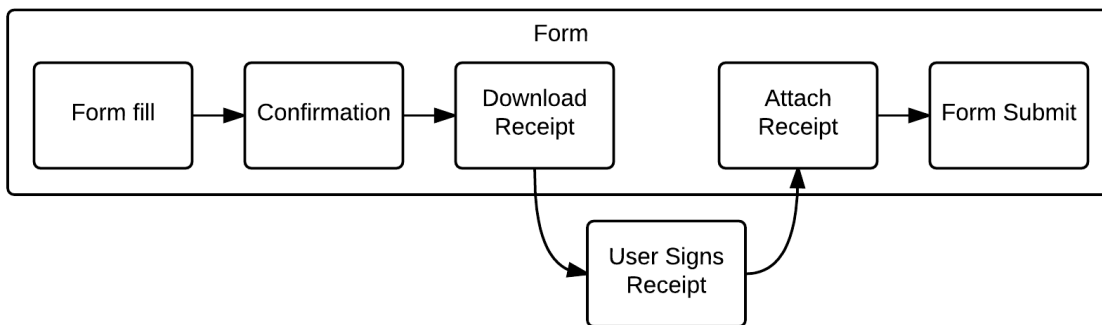
Normally when implementing wet signatures the method is to have users submit the form and then redirect to the wet signature space page. Here the user downloads a copy of their receipt, signs it then uploads the signed receipt. This takes the user experience outside the form and can often be jarring if styling between the form and the work space are not consistent.

Normal Wet Signature Flow



It is possible to implement this same behavior in-form.

In-Form Wet Signature Flow



Below are the steps required to get a working wet signature in-form. These steps will be outlined using an example application form.

Our example form contains two sections that provide the wet signature functionality. One is a confirmation page where users confirm they are happy with the form and can then download the receipt, sign it, and then uploaded it as an attachment. The other contains the wet signature widget that users sign on.

i Note that it is generally recommend that controls be put in place to prevent users from editing the form once they have downloaded and signed the receipt. This can be done by adjusting either the ability to edit or the visibility of the previous sections of the form once the user has confirmed they are ready to sign.

Confirmation

Confirmation

Application Form - Wet Signature

Fields marked with * are required

Please confirm you application

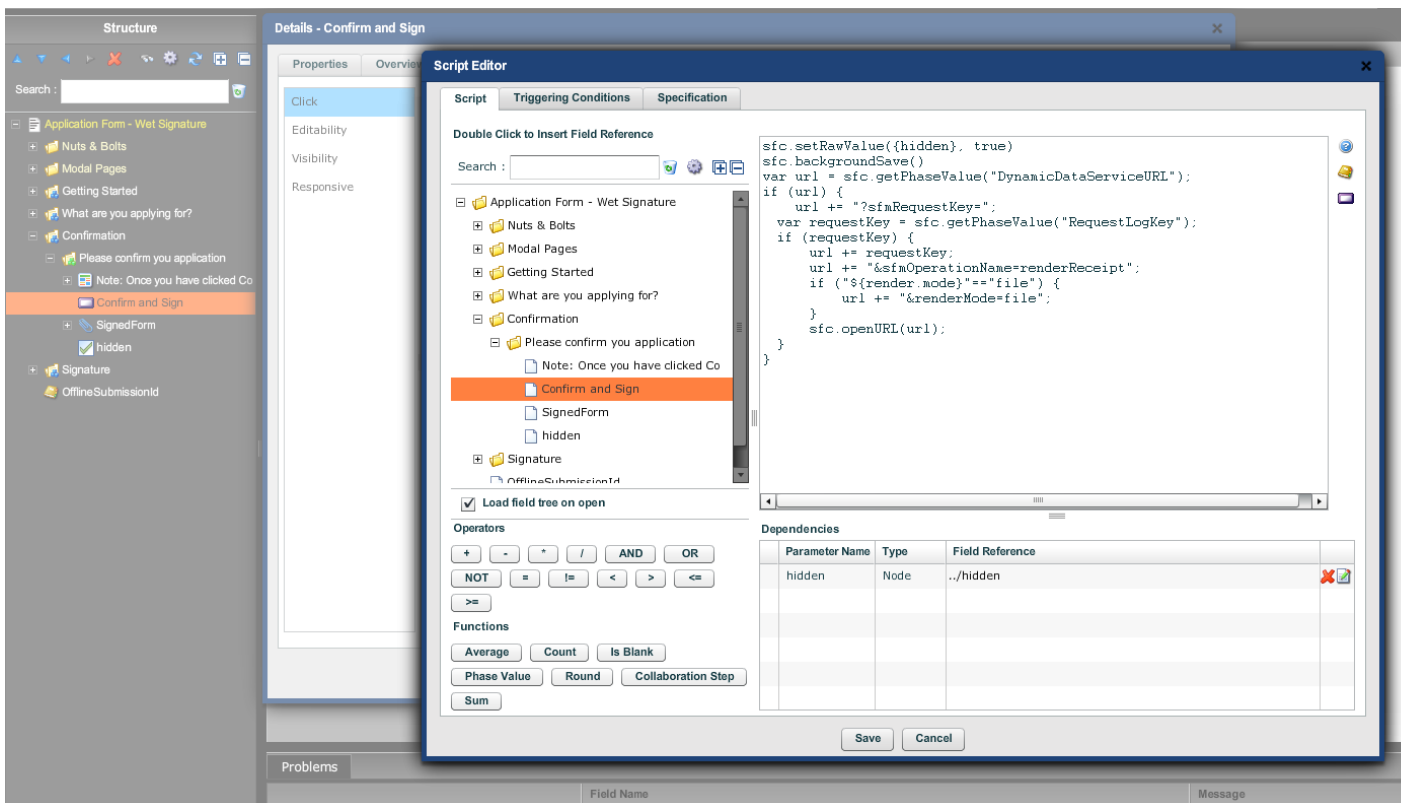
If you are happy with you application please click the confirm and sign button. This will allow you to download a copy of the form to sign. You must then upload it to finish the application.

Note: Once you have clicked Confirm and Sign link you will no longer be able to edit the form. Please make you are happy with the information provided.

[Confirm and Sign](#)

Your confirmation page is where the form user will download, sign and upload the receipt. It is important that you let the user know that they will not be able to go back and edit the form after finalizing the signed receipt. Whether this is done on download or upload of the receipt is up to the business use case.

First we need to generate the receipt. To do this we need to make a call to TM to generate a receipt from the current form data. This is done using the Dynamic Data Service URL and the Request Log Key. Start by adding a basic push button, and open up the on click script editor.



The following script will save the form and then retrieve the form receipt from TM.



It is important that you make sure the form is saved before you call the receipt render operation so that all form data in TM is up to date.

```
sfc.setBackgroundSave()
var url = sfc.getPhaseValue("DynamicDataServiceURL");
if (url) {
    url += "?sfmRequestKey=";
    var requestKey = sfc.getPhaseValue("RequestLogKey");
    if (requestKey) {
        url += requestKey;
        url += "&sfmOperationName=renderReceipt";
        if ("${render.mode}"=="file") {
            url += "&renderMode=file";
        }
        sfc.openURL(url);
    }
}
```

The receipt will have a signature section (outlined later in the article) that is set to be visible only in the receipt.

Signature

Please Sign Here

Date / /

Print Name

Once the receipt has been downloaded and signed the user will need to upload it to the form as an attachment. On the confirmation page add an attachment field for this purpose.

Confirmation

Application Form - Wet Signature

*Fields marked with * are required*

Please confirm you application

If you are happy with you application please click the confirm and sign button. This will allow you to download a copy of the form to sign. You must then upload it to finish the application.

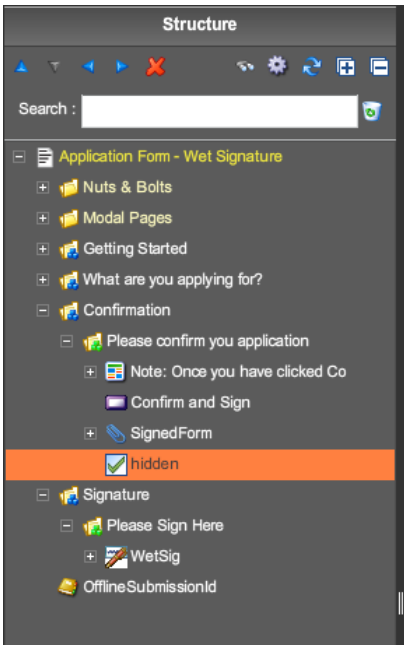
Note: Once you have clicked Confirm and Sign link you will no longer be able to edit the form. Please make you are happy with the information provided.

[Confirm and Sign](#)

Please uploade signed form receipt *

[Click to Upload](#)

In the example form the attachment field is hidden until the Confirm and Sign link button is clicked. The hide functionality is implemented by adding a hidden check box to the section. When the confirm link is clicked the value of the check box is changed to checked. The attachment field then has a visibility script that returns the value of the check box to determine if it should be visible or not.



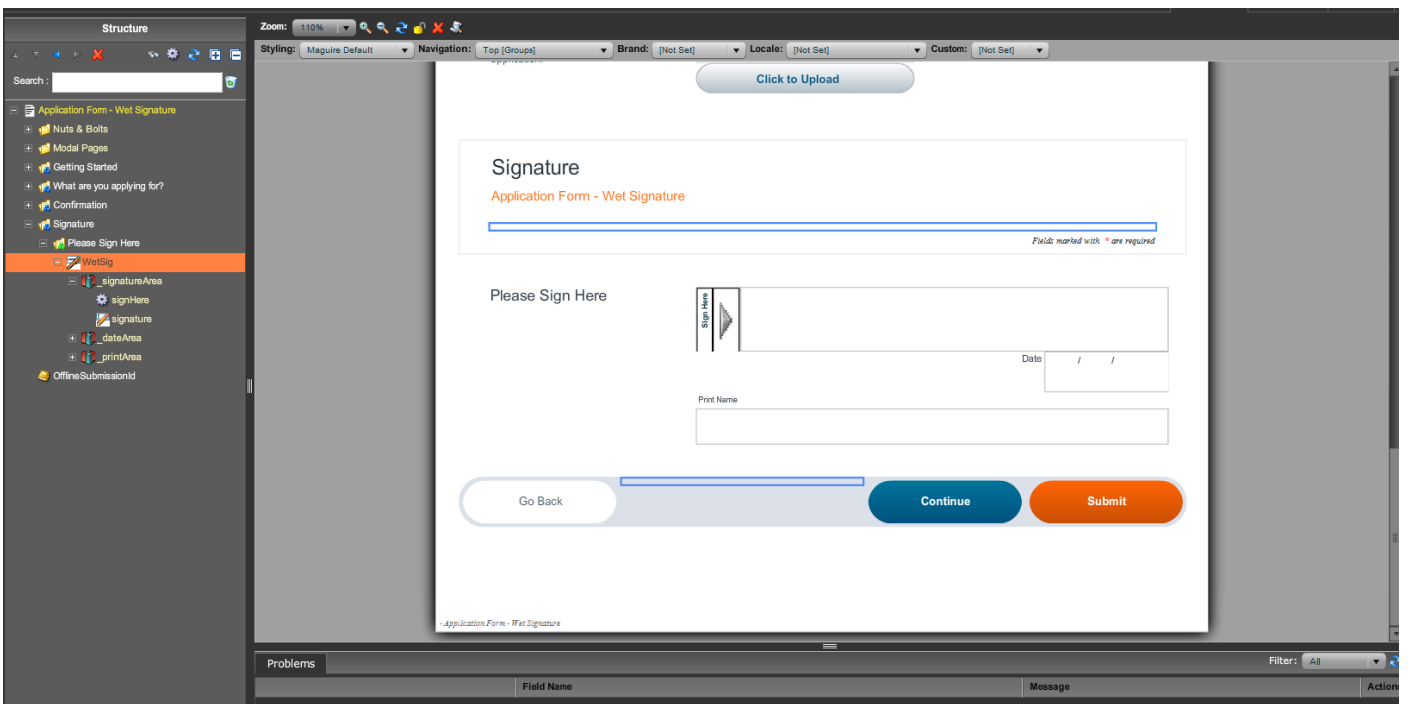
The code added to the confirm link is:

```
sfc.setRawValue({hidden}, true)
```

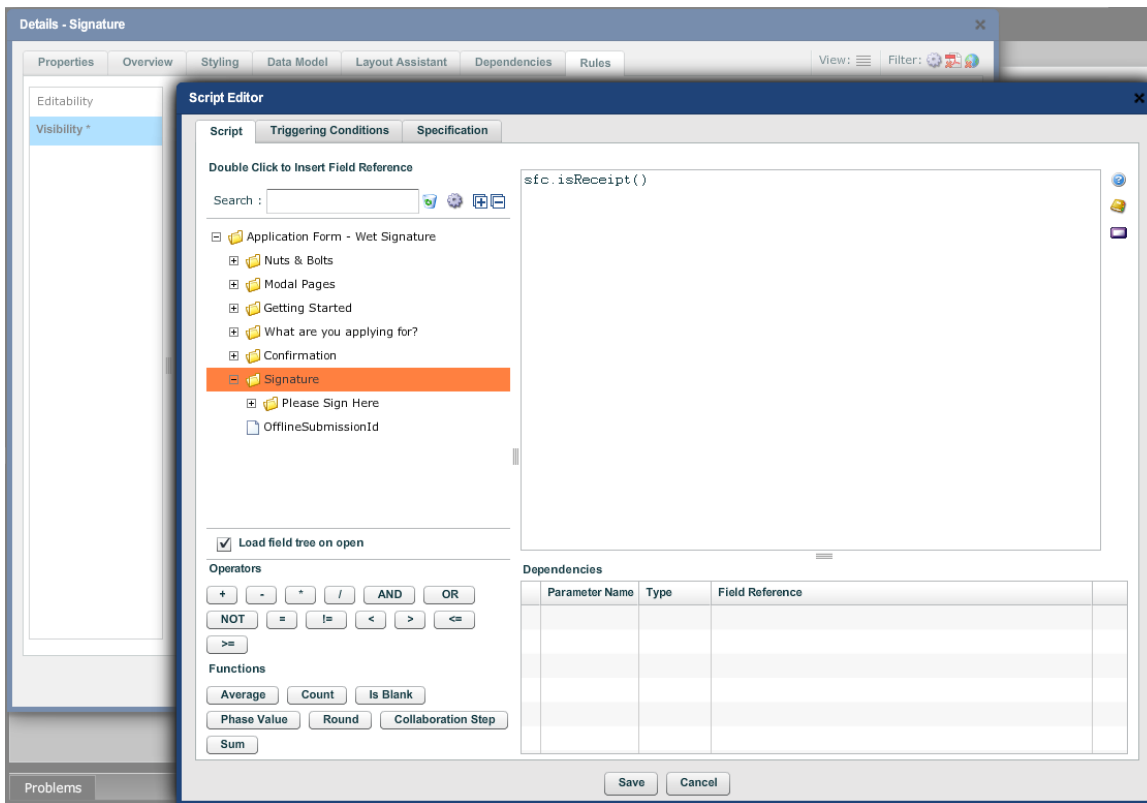
This check box can be used to determine where the user is in the form flow and control a number of things in form that determine how you want the user to work through the wet sign function. More on the form flow is outlined in the sections below.

Signature

The signature section simply contains the Wet signature widget that users will sign on when they download the receipt.



In the example application form there is added logic to hide this section in form, so that it is only displayed in the receipt. To do this on the section visibility rules add a script to check if it is in receipt mode (see screenshot below)



Depending on the use case you may also want to add some visibility logic to the Confirmation section as well, so that the confirmation doesn't appear in the receipt.

Form Flow

When the user reaches the confirmation section of the form we want to control when they are allowed to edit the form. There are two main points of activity in the confirmation stage of the form submission, when they download the receipt, and when they upload the signed copy of that receipt. It will be up to the form designer, based on the form use case, as to what to allow the user to do once they have downloaded and/or signed the receipt. In the sample form provided users are unable to edit the rest of the form once they have click the 'Confirm and Sign' button link.

This functionality can be added two ways, either by hiding the other sections or setting them to not editable. For our example form we chose to set the previous form sections to not editable, this allows the users to still go back and check what they have entered into the form though they can't change it.

To add this functionality in the example form the hidden check box that handled the visibility of the attachment field was used. When the confirm and sign link button is pressed this check box is checked. On each of the previous sections there is an edit-ability rule script that checks the value of the checkbox. If it is checked the sections are disabled so they can't be edited. If the desire was to allow the user to still edit the form after downloading the receipt, the editable script on each of the previous form sections could instead point to the attachment field that will flag when a file is attached, disabling the sections.

We also want to make sure the user doesn't submit the form before they have uploaded the signed copy of the receipt. To control this simple add a bit of code to the 'is editable' script for the submit button that checks that a receipt has been attached to the form in similar way as the section editability.

Composer form descriptor

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<formdescriptor description="Test form for wet signature internal" stylesheet="Styling-Maguire-Default,Navigation-TopGroups" template="Template-Maguire">
  <content>
    <field name="_wizardNavigationBar" type="Block-WizardNavigationBar">
      <field name="_submitButtonBlock" type="Block-Standard" styles="style.wizard.navbar.button.block">
        <field name="Submit" type="Button-Submit" styles="style.wizard.navbar.button,style.wizard.navbar.button.submit">
          <setproperty name="rule.editable" value="javascript"/>
          <setproperty name="rule.editable.script" value="#cdata#">
            <![CDATA[
{hidden}
          ]]>
        </setproperty>
        <setproperty name="rule.editable.script.dependencies" value="../../../Confirmation/_outerArea/_contentArea/PleaseConfirmYouAppl/_outerArea/_contentArea/hidden"/>
        <setproperty name="rule.editable.script.parameters" value="hidden|isselected|!"/>
      </field>
    </field>
  </field>
  <field name="GettingStarted" type="Section-Standard-Level1">
    <setproperty name="rule.editable" value="javascript"/>
  </field>
</content>
</formdescriptor>
```

```

        <setproperty name="rule.editable.script" value="#cdata#">
        <![CDATA[
!{hidden}
        ]]>
        </setproperty>
        <setproperty name="rule.editable.script.dependencies" value="../Confirmation/_outerArea/_contentArea
/PleaseConfirmYouAppl/_outerArea/_contentArea/hidden"/>
        <setproperty name="rule.editable.script.parameters" value="hidden|isselected|!"/>
        <setproperty name="section.heading" value="Getting Started"/>
        <field name="_outerArea" type="Block-SectionOuter" styles="{section.outer.stylename}">
            <field name="_contentArea" type="Block-SectionContent" styles="{section.content.stylename}">
                <field name="AboutYou" type="Section-Standard-Level2">
                    <setproperty name="section.heading" value="About You"/>
                    <setproperty name="section.help.text" value="Section help goes here. Utilising the
inbuilt help on level 2 sections to give some context around the section is an effective form design principle."
/>
                    <field name="_outerArea" type="Block-SectionOuter" styles="{section.outer.stylename}">
                        <field name="_contentArea" type="Block-SectionContent" styles="{section.content.
stylename}">
                            <field name="Title" type="Field-DropDownList-Title" styles="styleset.size.small">
                                <setproperty name="label" value="Title"/>
                                <setproperty name="binding.name" value="Title"/>
                            </field>
                            <field name="FirstName" type="Field-TextField" styles="styleset.size.medium">
                                <setproperty name="layout.data" value="sameline"/>
                                <setproperty name="label" value="First Name"/>
                            </field>
                            <field name="LastName" type="Field-TextField" styles="styleset.size.medium">
                                <setproperty name="layout.data" value="sameline"/>
                                <setproperty name="label" value="Last Name"/>
                            </field>
                            <field name="Address" type="Block-Address">
                            </field>
                        </field>
                    </field>
                </field>
            </field>
        </field>
        <field name="WhatAreYouApplyingFo" type="Section-Standard-Level1">
            <setproperty name="rule.editable" value="javascript"/>
            <setproperty name="rule.editable.script" value="#cdata#">
            <![CDATA[
!{hidden}
            ]]>
            </setproperty>
            <setproperty name="rule.editable.script.dependencies" value="../Confirmation/_outerArea/_contentArea
/PleaseConfirmYouAppl/_outerArea/_contentArea/hidden"/>
            <setproperty name="rule.editable.script.parameters" value="hidden|isselected|!"/>
            <setproperty name="section.heading" value="What are you applying for? "/>
            <field name="_outerArea" type="Block-SectionOuter" styles="{section.outer.stylename}">
                <field name="_contentArea" type="Block-SectionContent" styles="{section.content.stylename}">
                    <field name="TellUsWhatYouWant" type="Section-Standard-Level2">
                        <setproperty name="section.heading" value="Tell us what you want."/>
                        <field name="_outerArea" type="Block-SectionOuter" styles="{section.outer.stylename}">
                            <field name="_contentArea" type="Block-SectionContent" styles="{section.content.
stylename}">
                                <field name="CreditCard" type="Field-Checkbox">
                                    <setproperty name="label" value="Credit Card"/>
                                </field>
                                <field name="Limit" type="Field-Currency" styles="styleset.size.medium">
                                    <setproperty name="label" value="Limit"/>
                                    <setproperty name="data.initial.value" value="1000"/>
                                    <setproperty name="rule.visibility" value="javascript"/>
                                    <setproperty name="rule.visibility.script.dependencies" value="../CreditCard"
/>
                                    <setproperty name="rule.visibility.script" value="#cdata#">
                                    <![CDATA[
{CreditCard}
                                    ]]>
                                    </setproperty>
                                    <setproperty name="rule.visibility.script.parameters" value="
CreditCard|isselected|!"/>
                                </field>
                                <field name="HomeLoan" type="Field-Checkbox">
                                    <setproperty name="label" value="Home Loan"/>
                                </field>
                                <field name="LoanAmount" type="Field-Currency" styles="styleset.size.medium">

```

```

        <setproperty name="label" value="Loan Amount"/>
        <setproperty name="data.initial.value" value="1000000"/>
        <setproperty name="rule.visibility" value="javascript"/>
        <setproperty name="rule.visibility.script.dependencies" value="../../HomeLoan"/>
        <setproperty name="rule.visibility.script" value="#cdata#">
            <![CDATA[
{HomeLoan}
                ]]>
        </setproperty>
        <setproperty name="rule.visibility.script.parameters" value="
HomeLoan|isselected|!"/>
            </field>
        </field>
    </field>
</field>
</field>
</field>
</field>
</field>
</field>
<field name="Confirmation" type="Section-Standard-Level1">
    <setproperty name="rule.visibility" value="javascript"/>
    <setproperty name="rule.visibility.script" value="#cdata#">
        <![CDATA[
!sfc.isReceipt()
            ]]>
        </setproperty>
        <setproperty name="rule.visibility.initial" value="%null%"/>
        <setproperty name="section.heading" value="Confirmation"/>
        <field name="_outerArea" type="Block-SectionOuter" styles="{section.outer.stylenam}">
            <field name="_contentArea" type="Block-SectionContent" styles="{section.content.stylenam}">
                <field name="PleaseConfirmYouAppl" type="Section-Standard-Level2">
                    <setproperty name="section.heading" value="Please confirm you application"/>
                    <setproperty name="section.help.text" value="If you are happy with you application
please click the confirm and sign button. This will allow you to download a copy of the form to sign. You must
then upload it to finish the application."/>
                    <field name="_outerArea" type="Block-SectionOuter" styles="{section.outer.stylenam}">
                        <field name="_contentArea" type="Block-SectionContent" styles="{section.content.
stylenam}">
                            <field name="NoteOnceYouHaveClick" type="Text-Rich">
                                <setproperty name="label" value="Note: Once you have clicked Confirm and
Sign link you will no longer be able to edit the form. Please make you are happy with the information provided."
/>
                                    </field>
                                <field name="ConfirmAndSign" type="Button-Standard" styles="styleset.button.link"
>
                                    <setproperty name="label" value="Confirm and Sign"/>
                                    <setproperty name="rule.visibility.script.dependencies" value=""/>
                                    <setproperty name="rule.visibility.script" value="#cdata#">
                                        <![CDATA[
                                            ]]>
                                        </setproperty>
                                        <setproperty name="rule.visibility.script.parameters" value=""/>
                                        <setproperty name="rule.click.script" value="#cdata#">
                                            <![CDATA[
sfc.setRawValue({hidden}, true)
sfc.backgroundSave()
var url = sfc.getPhaseValue("DynamicDataServiceURL");
if (url) {
    url += "?sfmRequestKey=";
    var requestKey = sfc.getPhaseValue("RequestLogKey");
    if (requestKey) {
        url += requestKey;
        url += "&sfmOperationName=renderReceipt";
        if ("${render.mode}"=="file") {
            url += "&renderMode=file";
        }
        sfc.openURL(url);
    }
}
                                            ]]>
                                        </setproperty>
                                        <setproperty name="rule.click.script.dependencies" value="../../hidden/../../..
../../../../WhatAreYouApplyingFo/_outerArea/_contentArea/TellUsWhatYouWant/_outerArea/_contentArea/CreditCard,..
../../../../OfflineSubmissionId,../../../../_avoka_toolkit/_sfmSupport/DynamicDataServiceURL"/>
                                        <setproperty name="rule.click.script.parameters" value="hidden|node|-,
CreditCard|node|-,OfflineSubmissionId|string|-,DynamicDataServiceURL|string|-"/>
                                            </field>
                                        <field name="SignedForm" type="Block-Attachment-Field">

```

```

        <setproperty name="rule.visibility" value="javascript"/>
        <setproperty name="rule.visibility.script.dependencies" value="../hidden"/>
        <setproperty name="rule.visibility.script" value="#cdata#"
            <![CDATA[
{hidden}
                ]]>
            </setproperty>
            <setproperty name="rule.visibility.script.parameters" value="
hidden|isselected|!"/>
                <setproperty name="rule.mandatory" value="always"/>
                <setproperty name="rule.mandatory.message" value="Please upload a signed
copy of this form."/>
                    <setproperty name="attachmentblock.name" value="WetSignedForm"/>
                    <setproperty name="attachmentblock.description" value="Form receipt with wet
signature"/>
                        <setproperty name="attachment.caption.text" value="Please uploade signed
form receipt"/>
                            </field>
                            <field name="hidden" type="Field-Checkbox">
                                <setproperty name="label" value="hidden"/>
                                <setproperty name="rule.visibility" value="never"/>
                            </field>
                        </field>
                    </field>
                </field>
            </field>
            <field name="Signature" type="Section-Standard-Level1">
                <setproperty name="layout.data" value="region=center"/>
                <setproperty name="rule.visibility" value="javascript"/>
                <setproperty name="rule.visibility.script" value="#cdata#"
                    <![CDATA[
sfc.isReceipt()
                ]]>
                    </setproperty>
                    <setproperty name="rule.visibility.initial" value="%null%"/>
                    <setproperty name="section.heading" value="Signature"/>
                    <field name="_outerArea" type="Block-SectionOuter" styles="{section.outer.stylenam}">
                        <field name="_contentArea" type="Block-SectionContent" styles="{section.content.stylenam}">
                            <field name="PleaseSignHere" type="Section-Standard-Level2">
                                <setproperty name="section.heading" value="Please Sign Here"/>
                                <field name="_outerArea" type="Block-SectionOuter" styles="{section.outer.stylenam}">
                                    <field name="_contentArea" type="Block-SectionContent" styles="{section.content.
stylenam}">
                                        <field name="WetSig" type="Block-Signature">
                                            <field name="_signatureArea" type="Block-Standard">
                                                <field name="signature" type="Field-Multiline-TextField">
                                                    <setproperty name="binding.name" value="signature"/>
                                                </field>
                                            </field>
                                        </field>
                                    </field>
                                </field>
                            </field>
                        </field>
                    </field>
                </field>
            </field>
            <field name="OfflineSubmissionId" type="Data-Phase" styles="styleset.size.medium">
                <setproperty name="label" value="OfflineSubmissionId"/>
                <setproperty name="binding.name" value="OfflineSubmissionId"/>
                <setproperty name="binding.ref" value="$record.SFMDData.SystemProfile.OfflineSubmissionId"/>
            </field>
        </content>
    </formdescriptor>

```